The Business of Continuous Delivery

Kurt Bittner
Principal Analyst

@ksbittner
The Business of Continuous Delivery

Kurt Bittner
Principal Analyst

@ksbittner
What Is Driving The Need for Continuous Delivery?

“How concerned is your business with the following issues?”
(Percentage answering “very concerned” or “concerned”)

- Rising pressure to reduce costs: 70%
- The need to improve the capabilities of your products/services: 70%
- Increasing expectations from customers: 68%
- Competition against your products/services: 62%
- Reducing IT operational costs to free up money for new technology development and product/service innovation: 61%
- The changing demographics of your workforce and finding good employees: 59%
- The accelerating rate of technology change and its effect on your product/service life cycles: 58%
- Rising regulations in your industry and/or government involvement in the economy: 54%
- The quality of technology infrastructure in your local market (e.g., voice, data): 44%

Base: 3,659 IT budget decision-makers (multiple responses accepted)

Source: Forrsights Budgets And Priorities Tracker Survey, Q2 2012

© 2014 Forrester Research, Inc. Reproduction Prohibited
Disruption Is Everywhere

Your only security is in being the disruptor

Disruption demands delivering fast & frequently, with high quality

<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009</td>
<td>USAA Bank introduces mobile check deposit</td>
<td>Competitors view it as a novelty</td>
</tr>
<tr>
<td>2011</td>
<td>1 in 4 consumers want mobile check deposit</td>
<td>1 in 2 mobile banking users want mobile check deposit</td>
</tr>
<tr>
<td>2012</td>
<td>65% of consumers at least somewhat likely to switch banks to get mobile check deposit</td>
<td>Mobile check deposit is “table stakes”; those banks that don’t have it have lost customers</td>
</tr>
</tbody>
</table>

Demand for Mobile and Cloud Applications are Behind the Trend

Increased customer expectations

- Mobile, cloud becoming the dominant means of customer interaction
- Customers expect the same capabilities across all channels
- Delivering exceptional experiences means integrating all channels
Organizations are struggling to keep up

How often does your development team release mobile apps?

- Other (please describe): 2% IT Dev, 4% Pro Dev
- We release patches for defects as needed and then release major updates when we are ready: 10% IT Dev, 13% Pro Dev
- Releases to coincide with new versions of device OS: 8% IT Dev, 12% Pro Dev
- Every month or more (as needed): 2% IT Dev, 11% Pro Dev
- 5+ regular releases/yr/device: 13% IT Dev, 11% Pro Dev
- 2-4 regular releases/yr/device: 21% IT Dev, 28% Pro Dev
- <=1 release/yr/device: 25% IT Dev

Base: 172 Professional software developers and 87 IT developers building mobile applications
Source: Forrsights Developer Survey, Q1 2013
Shorter Time-To-Feedback = Faster Time-To-Value

Feedback Drives Improved Customer Experience and Business Results

› The faster the feedback, the less waste
› The less waste, the lower the cost
› Faster feedback means better results to customers, faster
› Happier customers = more customers, increased revenue
› Increased revenue and lower cost = better business results

“Measuring Mobile Apps”, December 13, 2013
Leading Organizations Use A Different Approach

- Rapid feedback loops drive innovative solutions
  - Customers are part of the feedback loop
- "Small batches" = less disruption, lower risk
  - Faster to market with smaller feature sets
- Silos dissolve; extended product teams are cross-functional
  - Hand-offs are eliminated; “one team” product development
- Less planning, more doing
  - “Commit by delivering”, not by planning
- The ability to pivot is a strategic advantage

Continuous Delivery is reshaping our approach to delivering software
The Change Is Already Happening

“The future is already here — it’s just not evenly distributed”

— William Gibson

› **Amazon** releases every 11.6 seconds, is the leader in public cloud infrastructure.

› **Netflix** is testing in production ~30% of NA Internet traffic.

› **Instagram** went from 0 to 14 million customers in 1 year, massive wealth creation.

› **Etsy** deploys API changes in 18 seconds, new website in <150 seconds.
Delays are often due to resource over-utilization; fix these first

Reducing task time is secondary, once wait time is eliminated
The Eight Tenets Of Faster Application Delivery

1. Create a Delivery Pipeline
3. Automate the Delivery Pipeline
8. Use Faster Delivery To Drive Business Strategy & Execution

2. Break down big efforts into small batches of work

4. Use Continuous Testing

5. Adopt Loosely coupled architectures

6. Treat Infrastructure As Code

7. Eliminate “Release Drama”
Agile: A Small-Batch WIP Management Approach

- Sprints provide a convenient way to manage small batches of work
- Demonstrations provide feedback that reshapes the Backlog and reduces waste
- In the absence of an Agile approach, “maintenance” releases are often inherently small-batch
Infrastructure-as-Code In Practice

- Configuration information (settings, users, permissions, accounts, installed software, properties, …) saved in SCM tool
- Scripts necessary to build new environments from configurations saved in SCM tool
- Environments automatically created, configured and populated with test data using automation
The Continuous Delivery Cycle

Small batches of work

- Release management, release automation, and change control software.
- What remains after testing has been automated is largely exploratory and usability testing, performed manually. With high levels of automation, this testing requires less coverage and focuses on spot-checking specific capabilities.
- Automated API-based testing frameworks, static code analysis tools, typically called from CI tools.
- Automated API-based testing frameworks, static code analysis tools, typically called from CI tools.

Environment Management

- Release (to Prod)
- Develop
- Deliver
- Build & Integrate
- Unit Tests
- Automated Acceptance Tests (including Regression Tests)
- Capacity, Security Tests
- User Acceptance Tests

Representative Tools and Technologies

- Integrated Development Environments (IDEs), editors/compilers/debuggers, application design tools, content management systems, packaged application customization tools, code review and collaboration tools and sourced components and services.
- Source code control systems and related asset management tools. Can involve code analysis and unit testing as part of a pre-commit process.
- Continuous integration tools
- Automated API-based testing frameworks, typically called from CI tools
- Release management and release automation software

What remains after testing has been automated is largely exploratory and usability testing, performed manually. With high levels of automation, this testing requires less coverage and focuses on spot-checking specific capabilities.

Small batches of work
API Testing In Action

UI – Presentation Layer

API Layer

Resource Layer
(including other Applications)

Test Harnesses
Loose Coupling Enables Independent Change

Each Layer and Service can change independently from all the others

- Services are versioned to allow interfaces to evolve
- Resources can be replaced without affecting applications
- Risk is reduced by isolating and eliminating dependencies
- The unit of release becomes an API change, not an application
- Applications and services can choose when they “upgrade” to a new interface
Maximize Utilization ➔ Maximize Throughput

Maximizing Utilization maximizes WIP and Waste

› With everyone as busy as possible, most work is partially completed most of the time
  • Waiting for someone else? Start something new.
  • Unfortunately there is someone downstream of you, too.
› When things change, a lot of that partially completed work gets scrapped
› The more delays and hand-offs, the more documents and ceremony, creating more delay

Maximizing Throughput minimizes Wait Time

› Work moves through the delivery pipeline faster
› Even though people are less busy, more useful work gets done because there is less scrap
› Organizational silos tend to emphasize utilization over throughput; focus everyone on the same measure
Typical Benefits Realized

› *Infrastructure as Art ➔ Infrastructure as Code*
  • Reduction in Production Incidents due to environment configuration errors
  • Elimination of “it works fine in my environment” excuses
  • Reduction in Mean Time to Repair (MTTR) by eliminating wait time for environment provisioning

› *Big Batches ➔ Small Batches, Maximize Utilization ➔ Maximize Throughput*
  • Decreased scrap and rework due to reduced WIP
  • Faster Cycle Time and reduced MTTR
  • Faster Time-to-Feedback, resulting in better solutions

› *Manual Builds ➔ Continuous Integration, Manual Testing ➔ Automated API-driven Testing*
  • Reduced labor cost
  • Improved consistency
  • Faster Cycle Time
  • Improved quality
  • Reduced governance and compliance costs
  • Improved effectiveness of governance

› *Integrated Architectures ➔ Loosely Coupled Services*
  • Faster Cycle Time and reduced MTTR
  • Lower Total Cost of Ownership through reduced maintenance costs and obsolescence avoidance
Typical Performance Measures

› **Cycle Time**
  • The time it takes to go from idea to business value realization

› **Mean Time To Repair (MTTR)**
  • The average time to takes to fix a defect or resolve an incident in production

› **Revenue Per (Application Delivery) Employee**
  • A measure of the economic productivity of the application delivery organization

› **Technical Debt**
  • A measure of the amount of corrective work that has been postponed
  • Lower technical debt means less risk and higher reliability

› **Revenue Acceleration or Lost Revenue Avoidance**
  • The value of being able to realize a revenue stream earlier, typically expressed as NPV
  • The value of correcting an error before it causes revenue to be lost

› **Customer Satisfaction Scores, Net Recommender Scores, App Ratings**
  • Measures of the happiness and loyalty of customers

› **Innovation Rate**
  • The percentage of the application development spending focused on new capabilities versus maintaining existing capabilities
Summary

» CD is driving the application delivery agenda for nearly every organization
  • Increased customer expectations
  • Increased competition

» Continuous delivery reduces cost
  • API test automation reduces manual testing costs
  • Static analysis reduces manual code review costs
  • Reducing WIP reduces the cost of scrap and rework
  • Small changes, released continuously results in fewer and less costly production incidents

» Continuous delivery improves quality
  • API test automation improves the accuracy of testing
  • Static analysis improves the accuracy and quality of code reviews
  • Automating release readiness evaluation ensures higher quality results

» Continuous delivery results in higher customer satisfaction
  • Faster innovation
  • Increased perceptions of responsiveness
  • Faster feedback leads to higher satisfaction
Thank you

Kurt Bittner
+1 617.613.8903
kbittner@forrester.com

@ksbittner

forrester.com