

# Quality Everywhere

How quality fits in every step of the software process

UVA Software Testing  
Fall 2017

**Corey Vaudo**  
Senior Vice President,  
Quality Assurance

**Allie Ivener**  
Software Engineer

**Conrad Rybka**  
Senior Quality Assurance Engineer



# Agenda



**Who we are**



**Quality  
Definition**



**Quality  
Everywhere**



**Q & A**



**Who we are**

# Applied Predictive Technologies

## Mission

Help drive evidence-based business decisions



## Clients

Deployed globally by over 300 organizations



## Expertise

Combine business strategy, math, and large databases to inform decisions



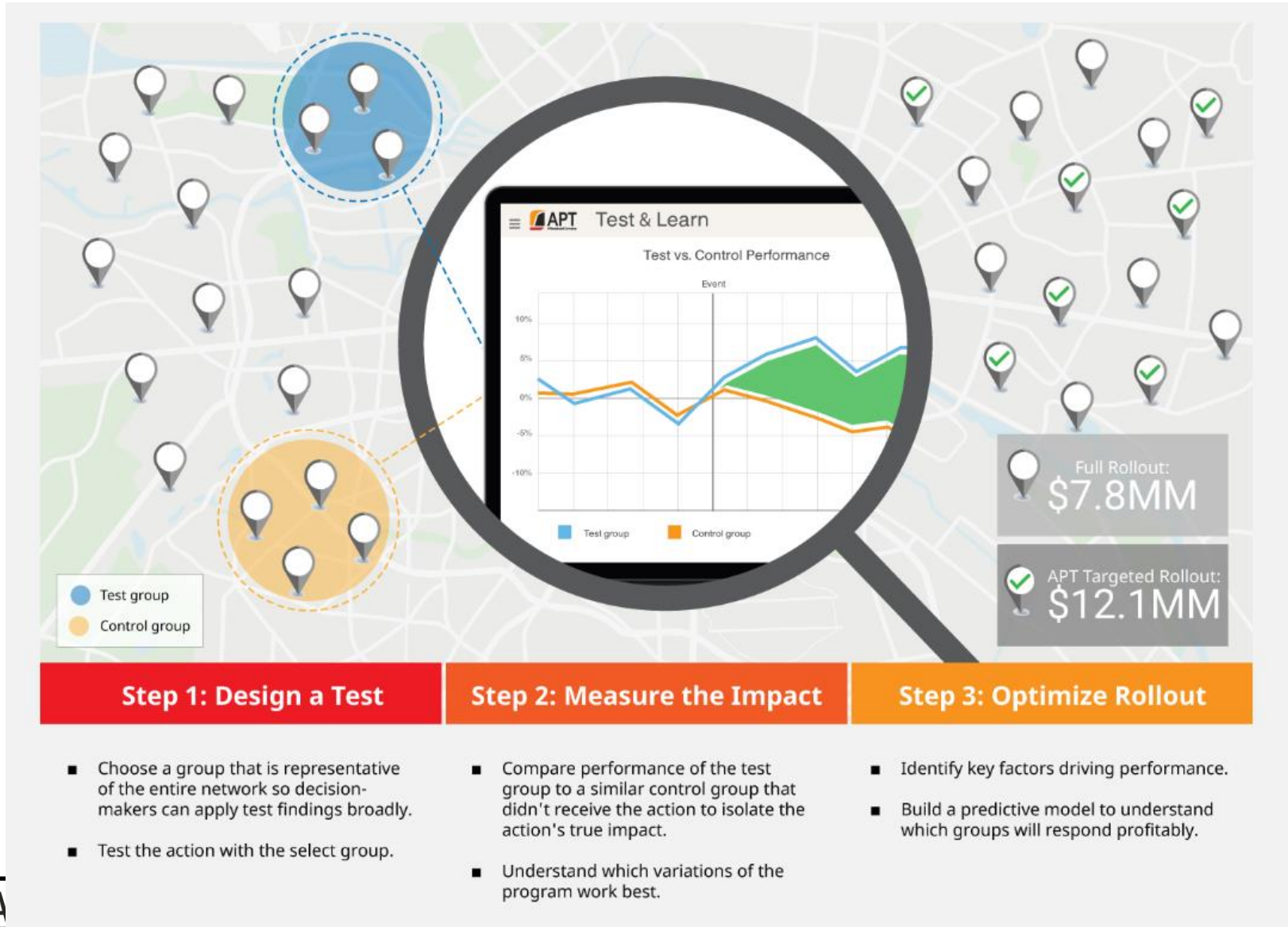
## Product

Cloud based software platform utilizing same underlying technology across industries



# APT – The Concept

We enable business experimentation



# APT – Business experiment examples

We tackle a variety of high stakes “experiments” across industries

Promotion Planning – is my promotion driving traffic or just eroding margin?

Abercrombie & Fitch

Menu Development – what entrees should I remove from my menu?  
What should I add?



Market Basket Profiling – How do different customers react to a promotion? How can we target promotions by customer segment?



Customer Incentive Programs – what rate should I offer on a CD to a new customer?



Network Planning – where should we build the next Holiday Inn?



Online-to-store Advertising - how much does online advertising affect my in-store sales? Do online sales cannibalize in-store sales?



# APT - Technology

Our approach requires expertise in numerous areas

## Big Data

- Host over 500 individual SQLServer databases
- Host over 1 PB of data
- Conduct “sku” level analysis for the world’s largest retailers

## Analytics

- Hold numerous patents for advanced analytic techniques
- Utilize both leading statistical packages and develop in-house algorithms

## Web Dev

- Host software entirely over the web utilizing latest front-end technologies (e.g. Redux, React, etc)

## CI / CD

- Deploy code multiple times per day
- Run tens of thousands of tests daily
- Utilize git, Jenkins, NUnit, Chef, Selenium & numerous other world-class “testing & deployment” technologies



**Quality**



# Stereotypical approach to quality

---

Product Management team develops  
“requirements”



Engineers build software to “spec”



QA team confirms software is high quality



Software delivered to users

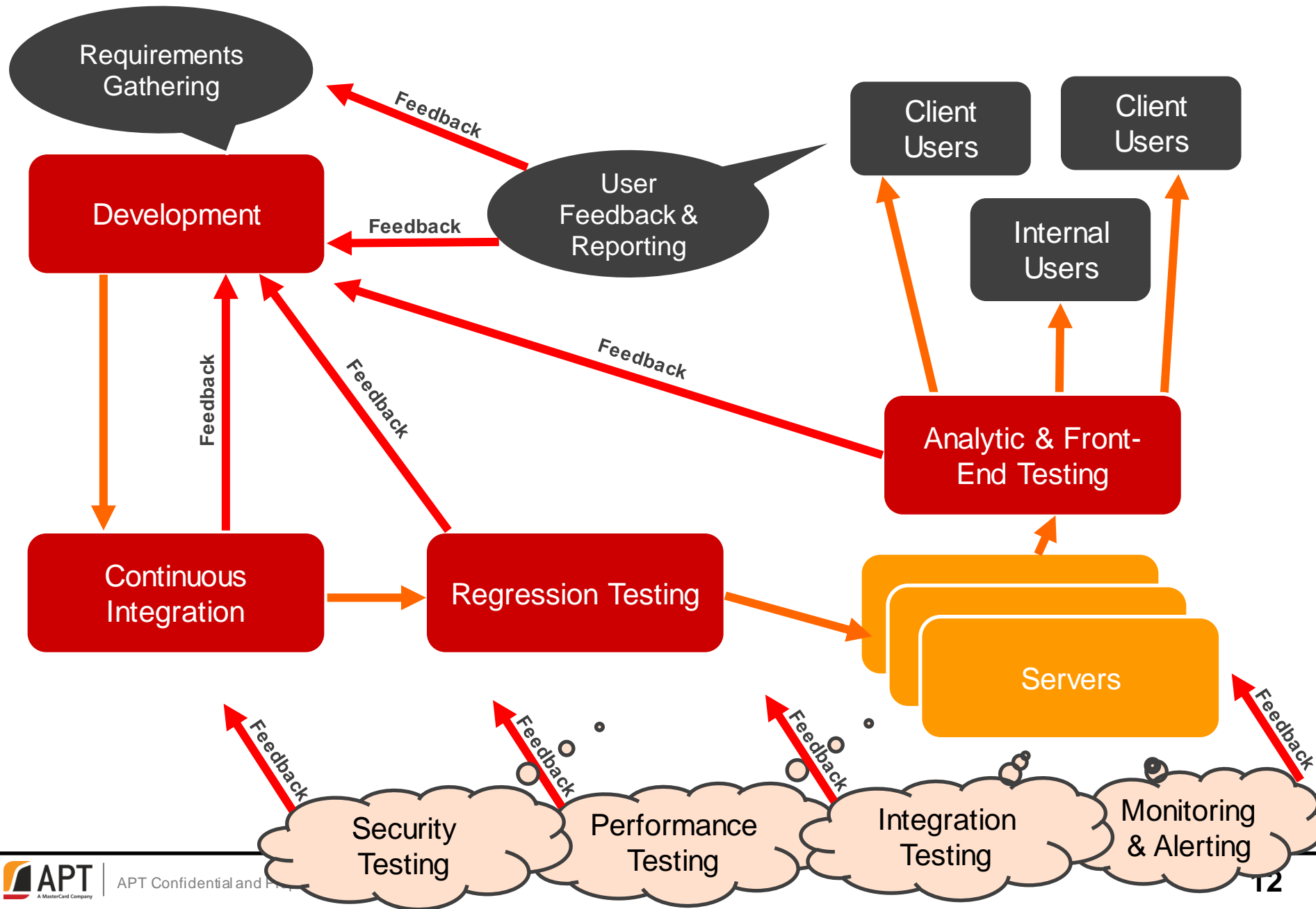
A word cloud featuring various terms related to software development, testing, and deployment. The words are arranged in a dense, overlapping manner, tilted at an angle. The colors of the words include shades of blue, red, brown, and orange. The terms include:

- Dogfooding
- Design Reviews
- Regression Testing
- Redundancy
- Reliability
- Code Reviews
- Manual Testing
- Vulnerabilities
- Software
- Analytic Accuracy
- Security
- Continuous Delivery
- Exploratory Testing
- Continuous Integration
- Integration Testing
- Front End-Load Times
- SQL Injection
- Scalability
- Audit Usage Reporting
- Feedback Loops
- Alerting
- Monitoring Performance
- Load Testing
- Automation

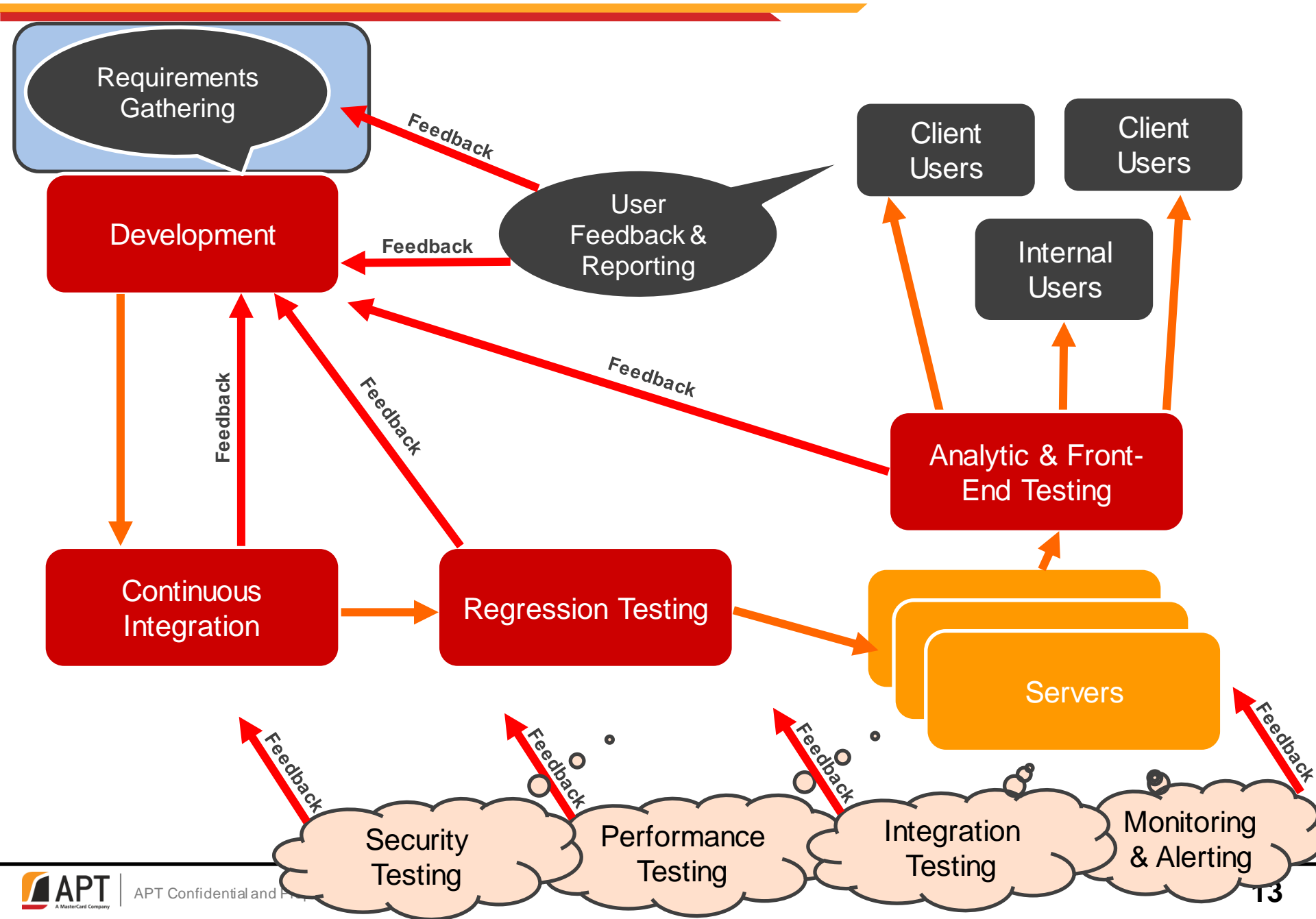


**Quality  
Everywhere**

# Quality is built-in everywhere throughout development process



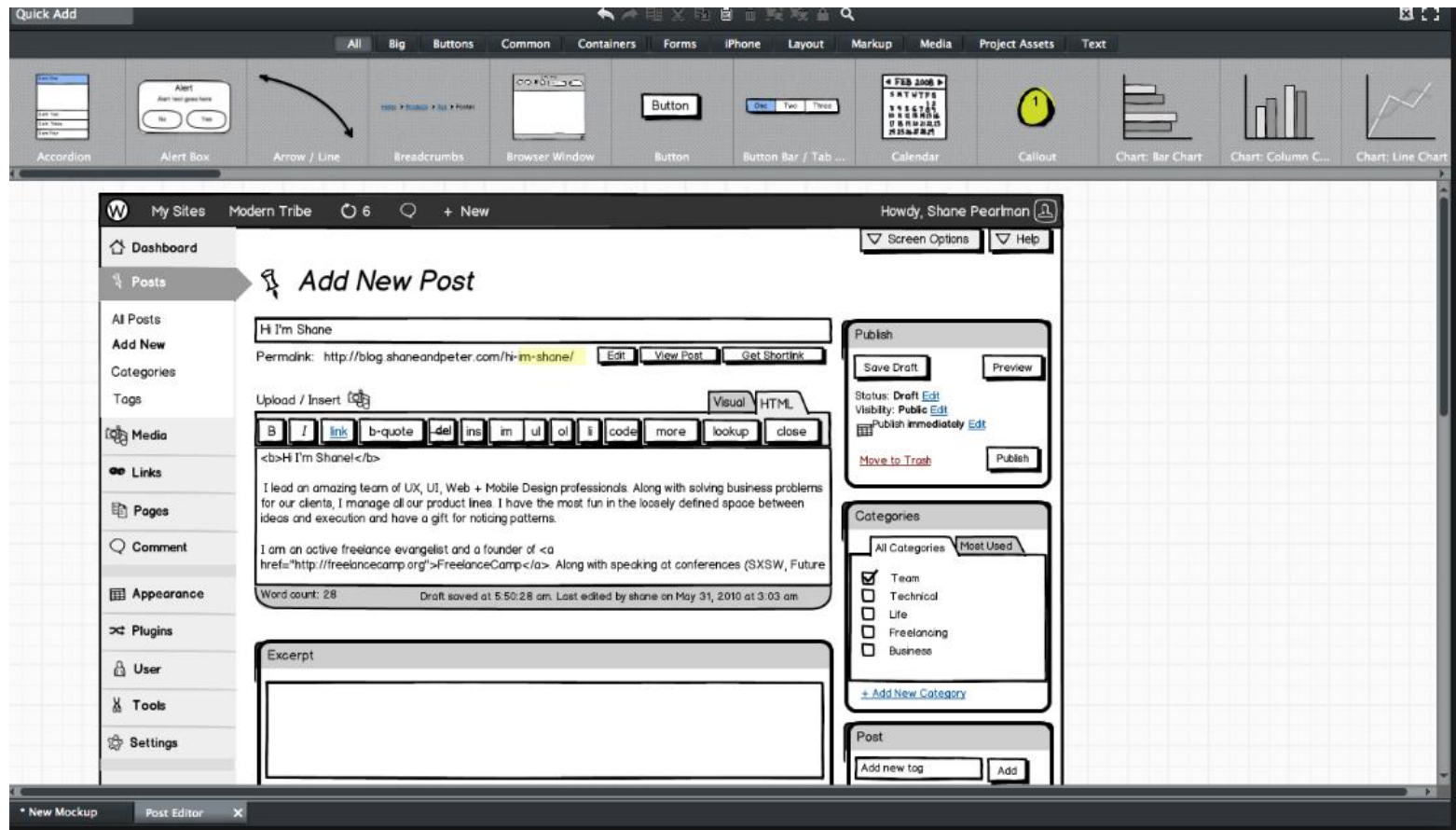
# Quality is built-in everywhere throughout development process



# Quality in the Requirements Process

Quality is built in the requirements process in multiple dimensions

- Prototyping



# Quality in the Requirements Process

Quality is built in the requirements process in multiple dimensions

- Usability Sessions



Developer watching videotape of usability test.

# Quality in the Requirements Process

## Quality is built in the requirements process in multiple dimensions

- Requirements testing

### Use Cases

Listing out the drilldown use cases, prioritized and grouped by similar/related actions. The priority is informed by the following aspects:

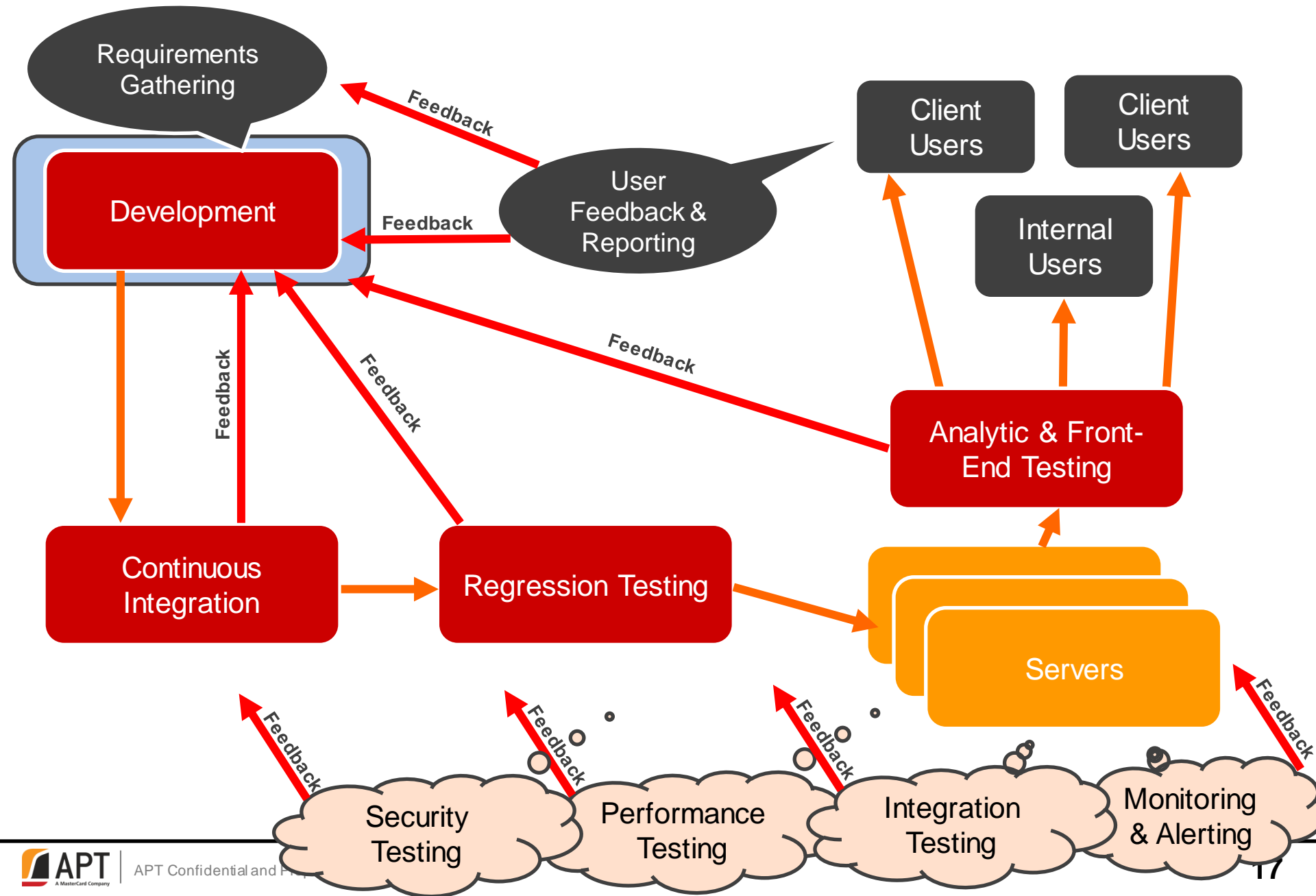
- frequency of use
- importance to the output generation
- dependence on other use cases - i.e. if one use case depends on the selection of another use case, it should generally be lower in priority - see Settings Hierarchy section below

The priority can then be used to help inform both the placement of the features to achieve these use cases as well as the relative emphasis of each feature on the drilldown page.

Priority	Item	Details / Notes
High	I want to select a metric most relevant to my business question so that I can best answer it. (metric selection)	
High	I want to select a specific timeframe so that I can focus on the relevant period for my business question. (timeframe selection)	
High	I want to select a visualization that best displays the answer for my relevant business question. (visualization selection)	
High	I want to summarize the metric and metric format used to construct an output so that I can quickly understand the underlying data for a given output. (metric summary)	
High	I want to summarize the timeframe used to construct an output.	
Medium	I want to choose a metric format that best answers my business question. (metric format selection)	
Medium	I want to summarize the metric and metric format used to construct an output so that I can quickly understand the underlying data for a given output.	
Medium	I want to choose whether to include a benchmark so that I can better contextualize my portfolio's performance. (benchmark toggle)	
Medium	I want to choose one or more attributes to split my output so that I can analyze metric performance segmented by those attributes. (split selection)	



# Quality is built-in everywhere throughout development process



# Quality is shared amongst all developers

## Numerous developer practices drive software quality

- Pull Requests and Code reviews

The screenshot displays a GitHub pull request interface. At the top, a code diff is visible, showing changes to a C# file. The diff includes line numbers 16 through 21. A comment by Pushkar Rege asks, "Can we check that the AptGeospatialDataset view exists and throw if not?" with a checkbox labeled "address". Below the comment, the code diff continues with lines 19 and 20, showing assignments for `this._aptQuery` and `this._tableProvider`.

The pull request is titled "APT-122720/tlog data refresh" and is created by Allie Ivener. The status bar shows "OPEN" and "master" as the target branch. The interface includes tabs for "Overview", "Diff", and "Commits". The "Details" section lists the changes: "c# changes to bring in new spend columns", "sql script for new currency attributes", "fixing tests", "most tests pass", "updates to views", "update tests for transactionCount", "update transactionCount in prior spend", and "fix type for new columns, fix GCO". On the right side, the status bar shows "1 build" with a green checkmark, "6 open tasks" with a checkmark, and a "Learn more" link.

```
16 17
17 18     public SqlGeospatialAttributeCalculator(AptQuery aptQuery, ITableProvider tableProvider)
18 19     {
19 20         this._aptQuery = aptQuery;
20 21         this._tableProvider = tableProvider;
21 22     }
```

Pushkar Rege  
Can we check that the AptGeospatialDataset view exists and throw if not?  
Reply · Create task · Like · 01 Sep 2017

☒ address

19 20 this.\_aptQuery = aptQuery;  
20 21 this.\_tableProvider = tableProvider;  
21 22 }

Allie Ivener APT-122720/tlog-data-refresh → master OPEN +5 Merge ...

### APT-122720/tlog data refresh

Overview Diff Commits

#### Details

Allie Ivener created a pull request 4 days ago

- c# changes to bring in new spend columns
- sql script for new currency attributes
- fixing tests
- most tests pass
- updates to views
- update tests for transactionCount
- update transactionCount in prior spend
- fix type for new columns, fix GCO

1 build ✓  
6 open tasks  
Learn more

# Quality is shared amongst all developers

## Numerous developer practices drive software quality

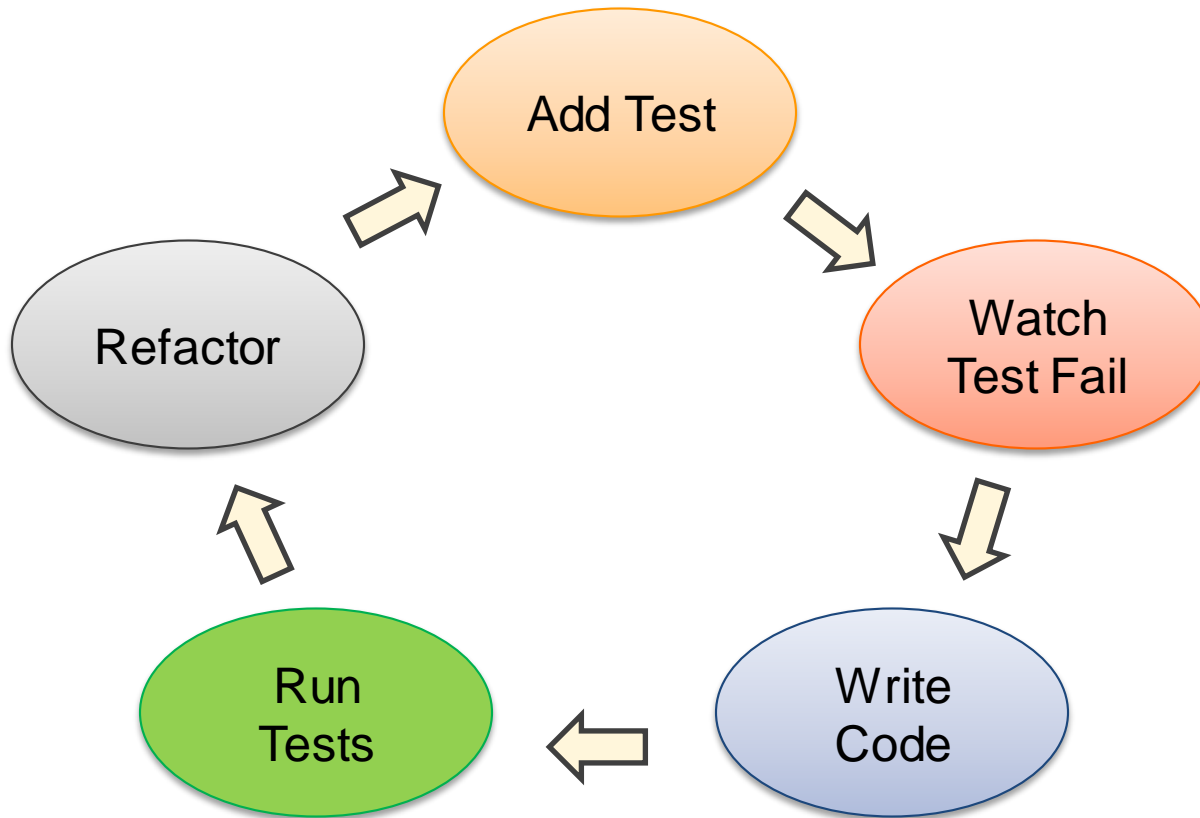
- Pair programming



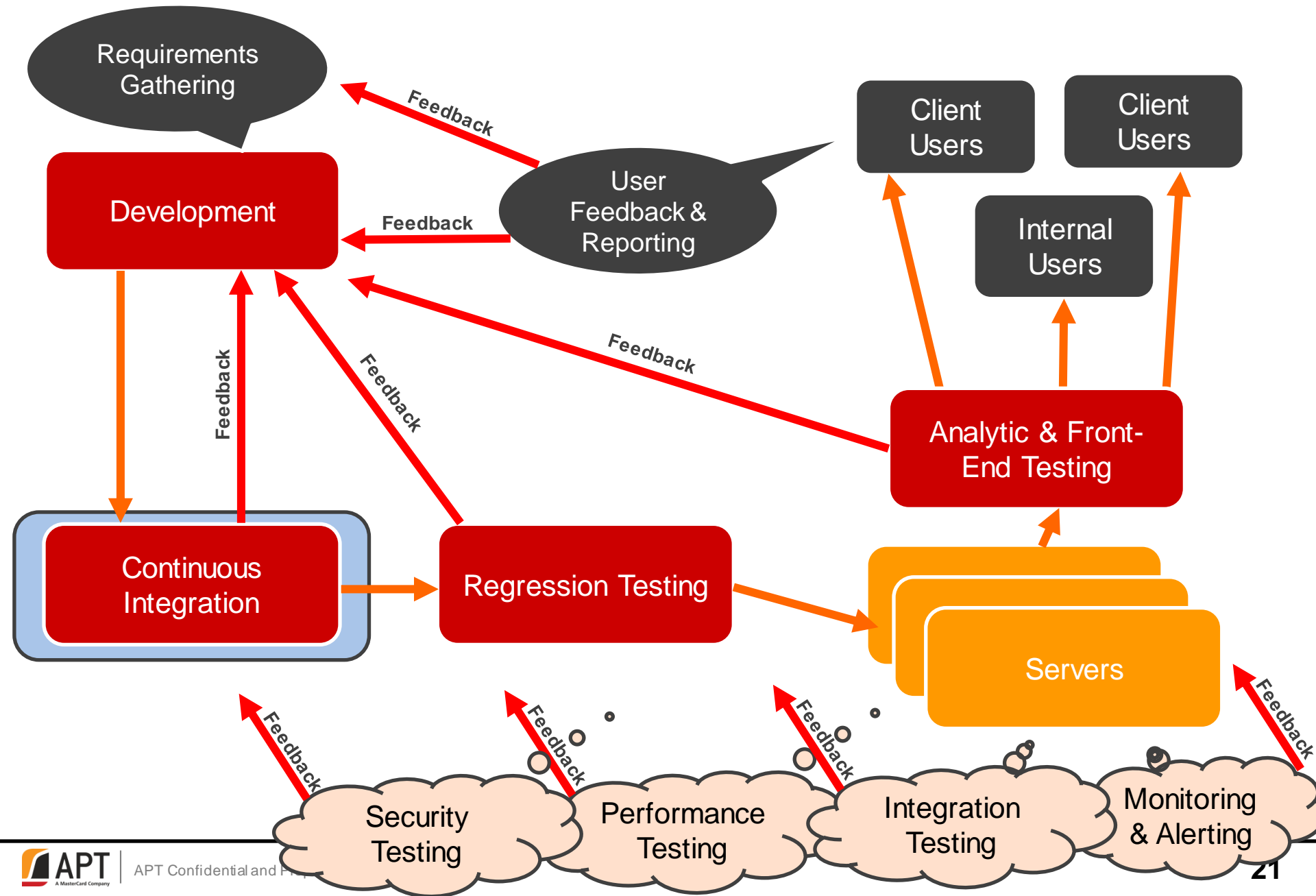
# Quality is shared amongst all developers

## Numerous developer practices drive software quality

- Test Driven Development



# Quality is built-in everywhere throughout development process



# Code is “integrated” every hour

“Continuous Integration” tools ensure all changes are “merged” and testing together frequently

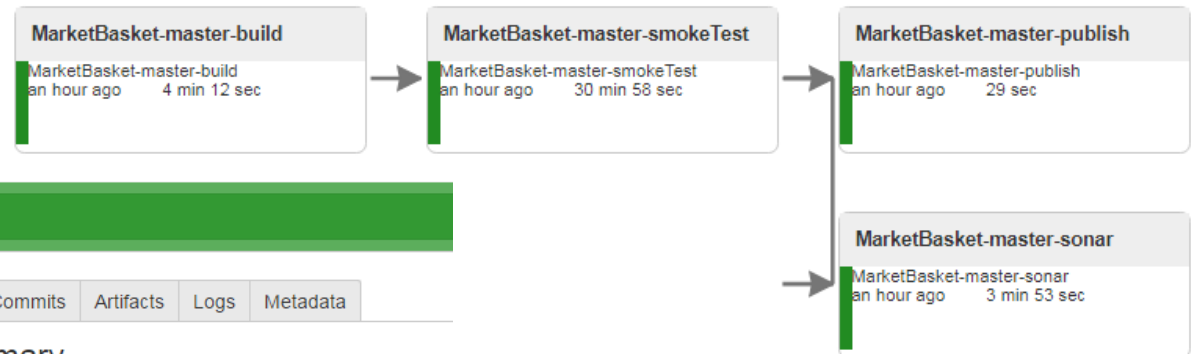


## Jenkins



Master

Total build time: 39 min 4 sec



✓ #987 was successful – Changes by [Rishab](#)

Stages & jobs

Build

✓ Build Solution

Smoke Tests

✓ Run Calc Tests

✓ Run Core Tests

✓ Run Data Api Tests

✓ Run DataUsage Tests

✓ Run Node Tests

✓ Run Results Client Tests

✓ Run Results Tests

✓ Run Web Tests

Summary Tests Commits Artifacts Logs Metadata

### Build result summary

#### Details

Completed 11 Sep 2017, 2:52:23 PM – 2 hours ago

Duration 10 minutes

Labels None

Revision [9a3c6cd0...](#)

Total tests 1203

Successful since #985 (1 hour before)

0

New failures

0

Existing failures

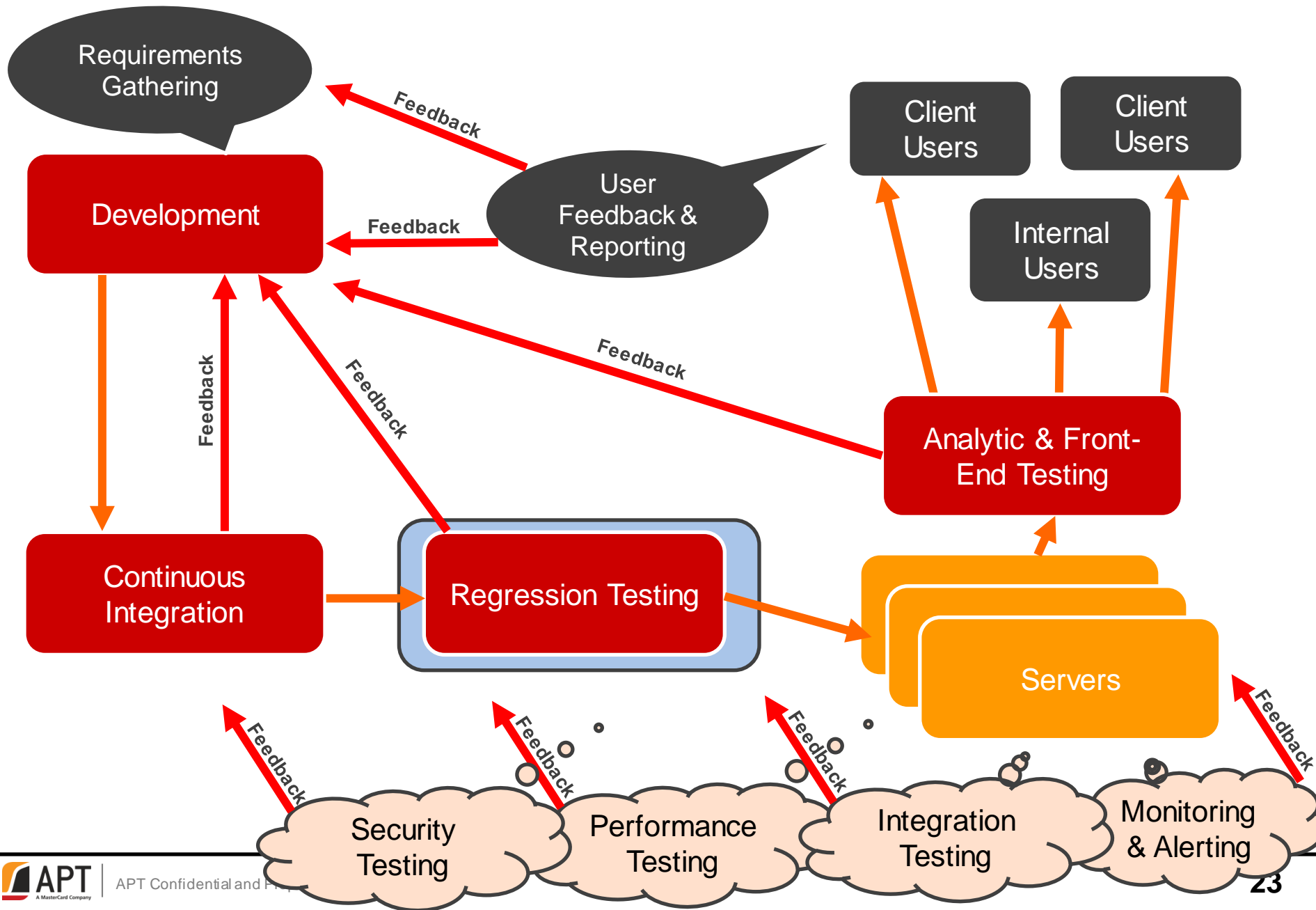
0

Fixed

21

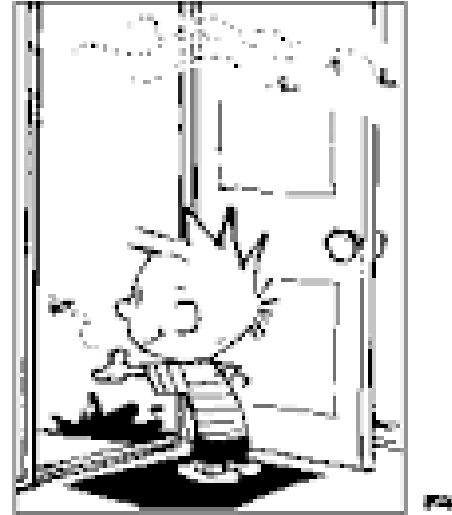
Quarantined /  
skipped

# Quality is built-in everywhere throughout development process



# Automated Regression Tesing

When you fix one bug, you introduce several newer bugs





# Automated Regression Testing

Manual test only once, but test automatically forever



Jenkins



## Test Result

0 failures (±0)

12 tests (±0)  
Took 6.7 sec.  
[add description](#)

## All Tests

Package	Duration	Fail	(diff)	Skip	(diff)	Pass	(diff)	Total	(diff)
<a href="#">Apt.Platform</a>	77 ms	0		0		2		2	
<a href="#">Apt.Platform.ABTesting.Tests.Client</a>	1.9 sec	0		0		3		3	
<a href="#">Apt.Platform.ABTesting.Tests.Client.Integration</a>	0.42 sec	0		0		1		1	
<a href="#">Apt.Platform.ABTesting.Tests.Core</a>	47 ms	0		0		1		1	
<a href="#">Apt.Platform.ABTesting.Tests.Web</a>	4.2 sec	0		0		5		5	

Lines Of Code

10,469

C#

Files

199

Directories

87

Lines

15,796

Functions

599

Classes

207

Statements

2,431

Accessors

713

Duplications

0.4%

Lines

66

Blocks

4

Files

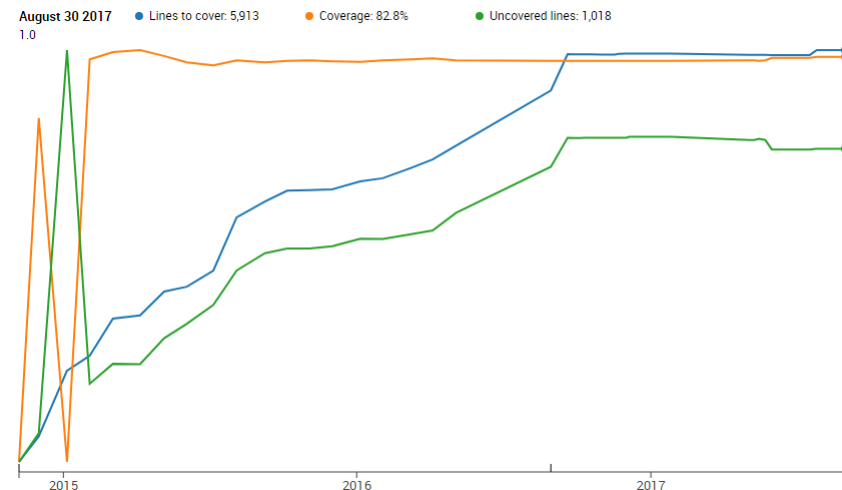
4

Unit Tests Coverage

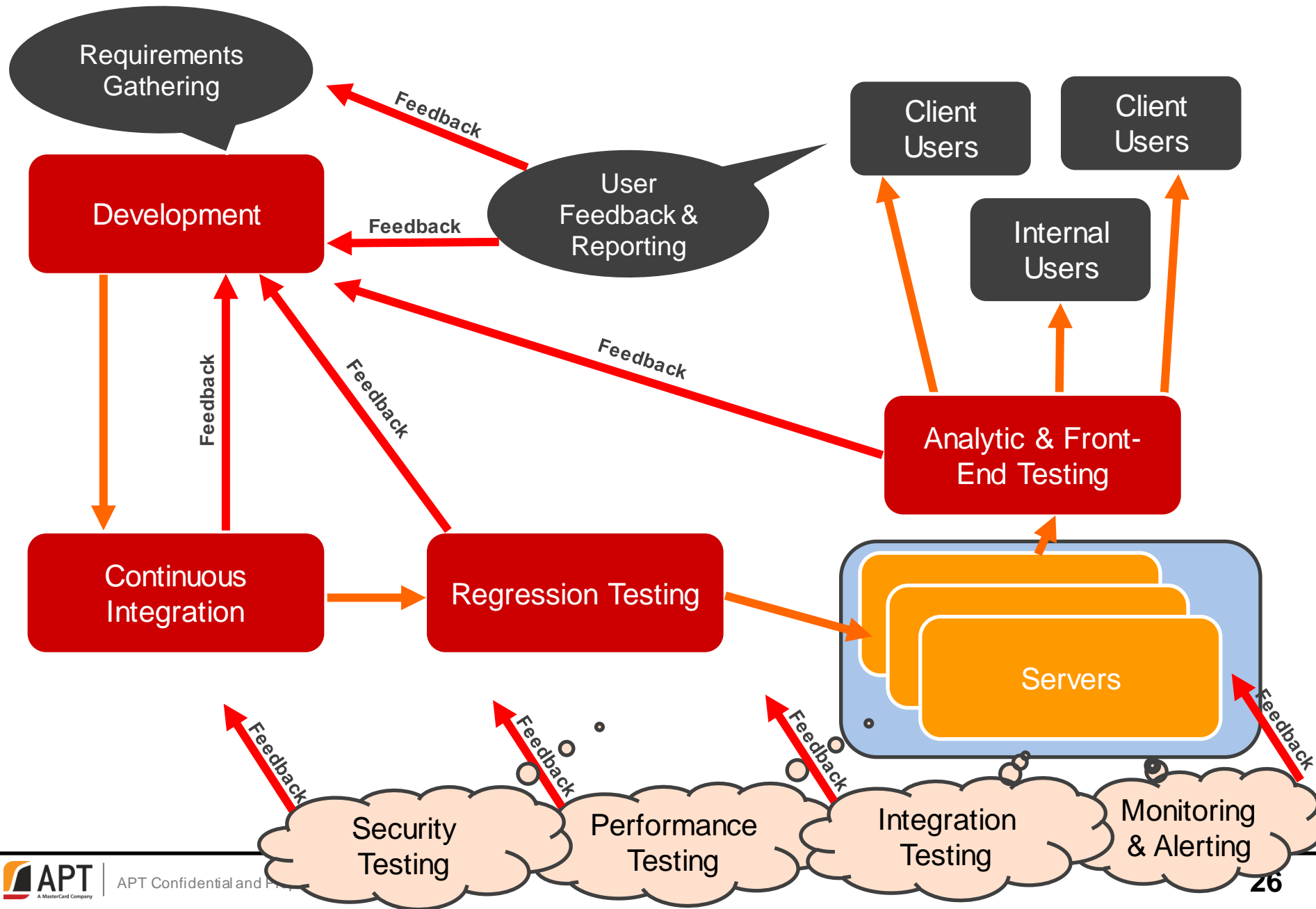
82.8%

Line Coverage

82.8%



# Quality is built-in everywhere throughout development process



# Configuration Management

Server consistency is a key driver of “software quality”



© c.20thC.Fox/Everett / Rex Featur



> Nodes

- Delete
- Manage Tags
- Reset Key
- Edit Run List
- Edit Attributes

Hirrfinger 0 0

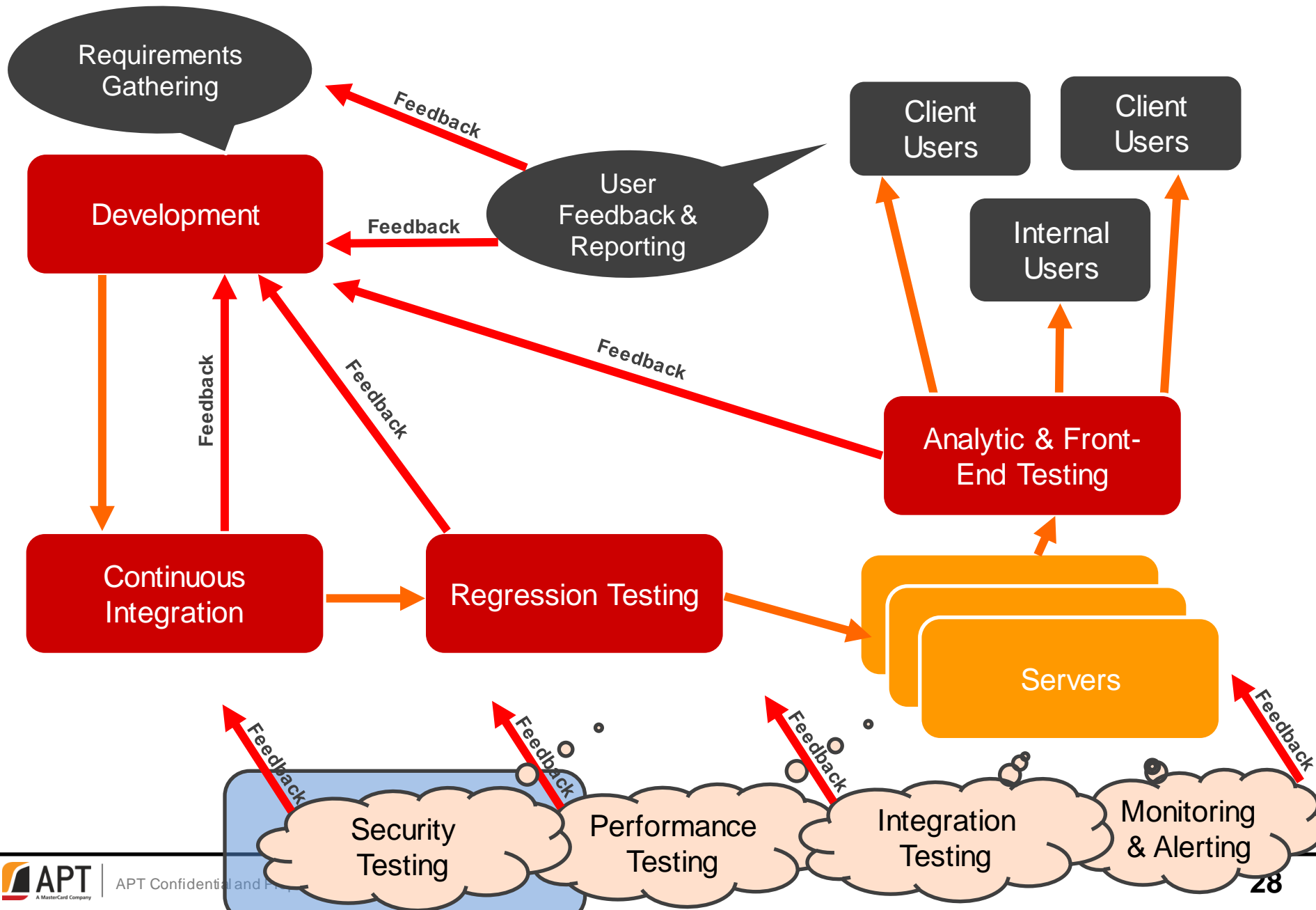
Actions

Settings

Edit

Version	Position
...	...
...	...
...	...

# Quality is built-in everywhere throughout development process



# Vulnerability Scanning

Testing for security vulnerabilities is essential when dealing with sensitive data



Scan Results (43 Risks)

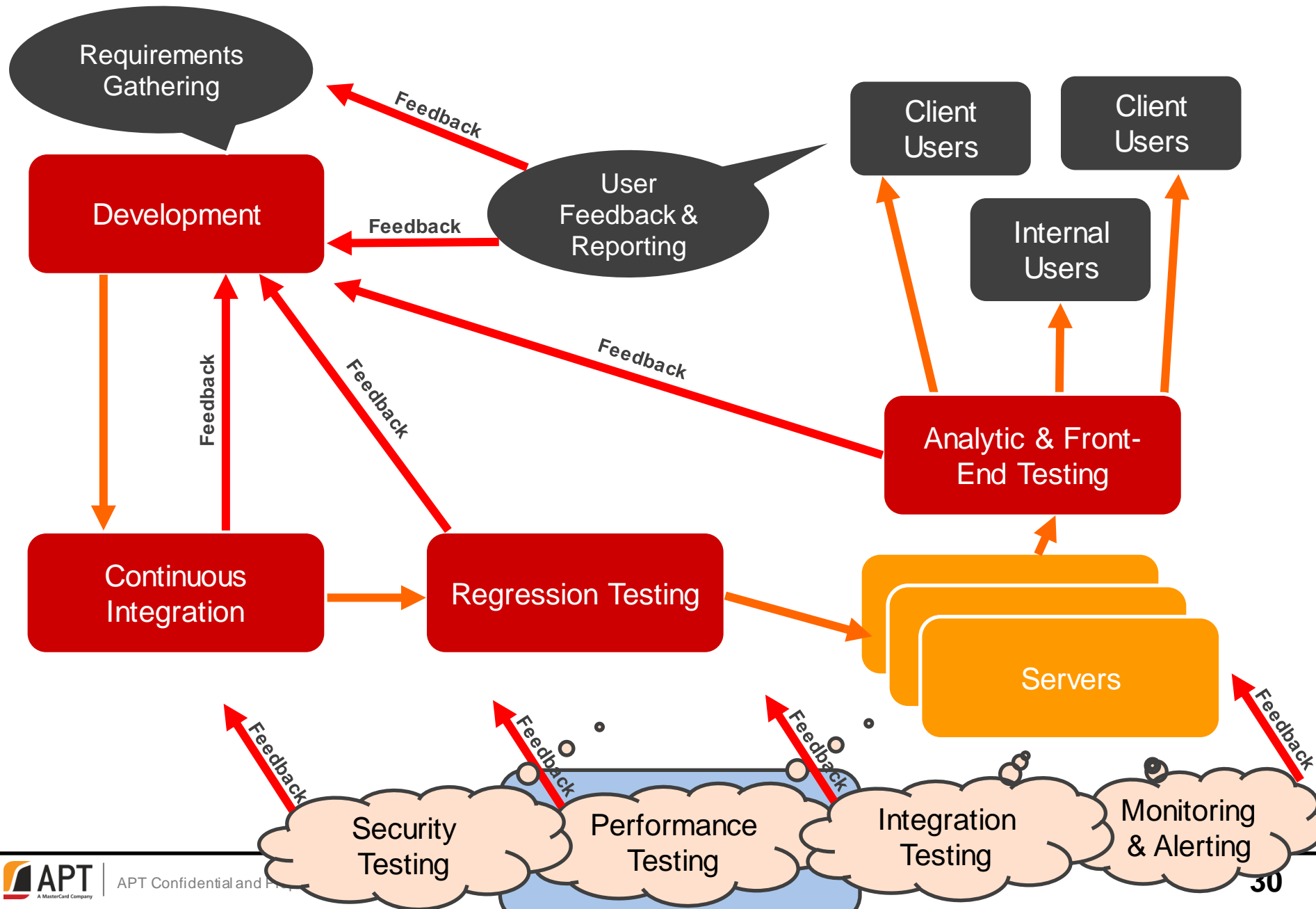
[Open Code Viewer](#)

<b>High Risk Vulnerabilities (0)</b>	
Reflected XSS All Clients	0
Client DOM XSS	0
<b>Medium Risk Vulnerabilities (13)</b>	
HttpOnlyCookies In Config	0
No Request Validation	0
Client DOM XSRF	11
Client Cross Frame Scripting Attack	1
Parameter Tampering	1
<b>Low Risk Vulnerabilities (30)</b>	
Missing X Frame Options	0
Client DOM Open Redirect	14
Client Insecure Randomness	6
Improper Exception Handling	4
Client Potential DOM Open Redirect	2
DebugEnabled	2
Off By One Error	1
Divide By Zero	1
CustomError	0
<b>Info Risk Vulnerabilities (0)</b>	

## Scan Details

Scan Start	Saturday, September 9, 2017 1:08:22 PM
Scan Time	00h:18m:34s
Scan Type	Full
Lines of Code	252089
File Count	1560

# Quality is built-in everywhere throughout development process



# Performance testing

A functional application is only valuable if it is performant



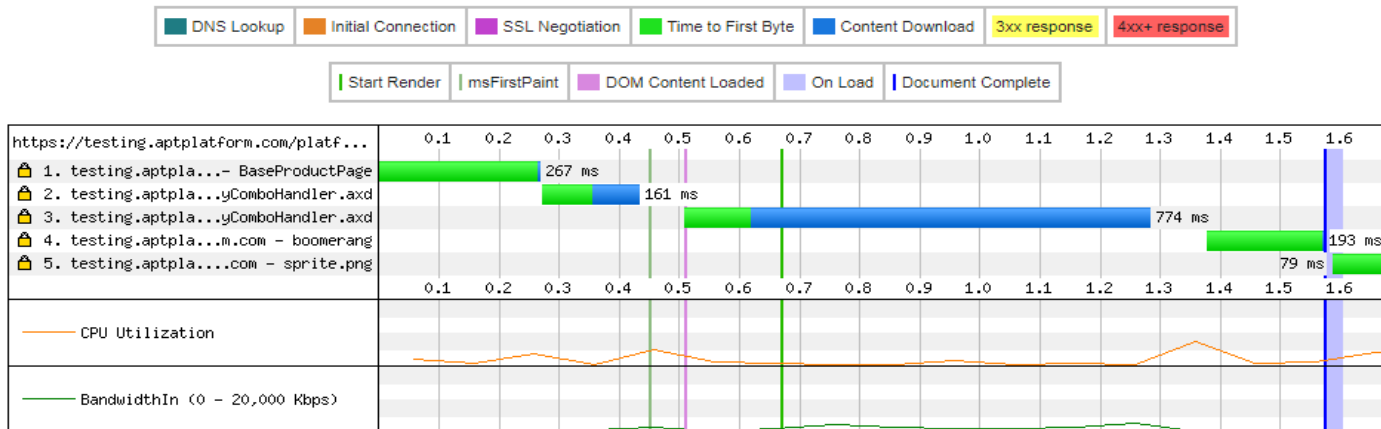
Tester: WEBPAGETEST-127.0.0.1  
Test runs: 3  
[Script](#) ▶

[Export HTTP Archive \(.har\)](#)

Load Time	First Byte	Start Render	Visually Complete	Speed Index	DOM Elements	Result (error code)	Document Complete			Fully Loaded		
							Time	Requests	Bytes In	Time	Requests	Bytes In
1.572s	0.263s	0.668s	1.700s	750	326	0	1.572s	4	171 KB	1.668s	5	178 KB

RUM First Paint	<a href="#">domContentLoaded</a>	<a href="#">loadEvent</a>
0.449s	0.509s - 0.509s (0.000s)	1.574s - 1.603s (0.029s)

## Waterfall View



[customize waterfall](#) • [View all Images](#)

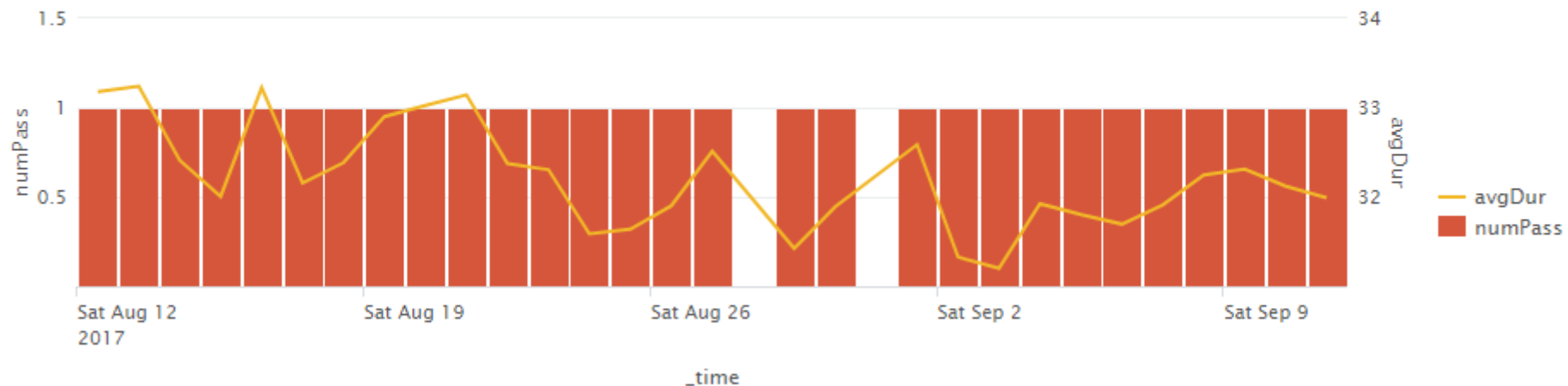
# Performance testing

A functional application is only valuable if it is performant



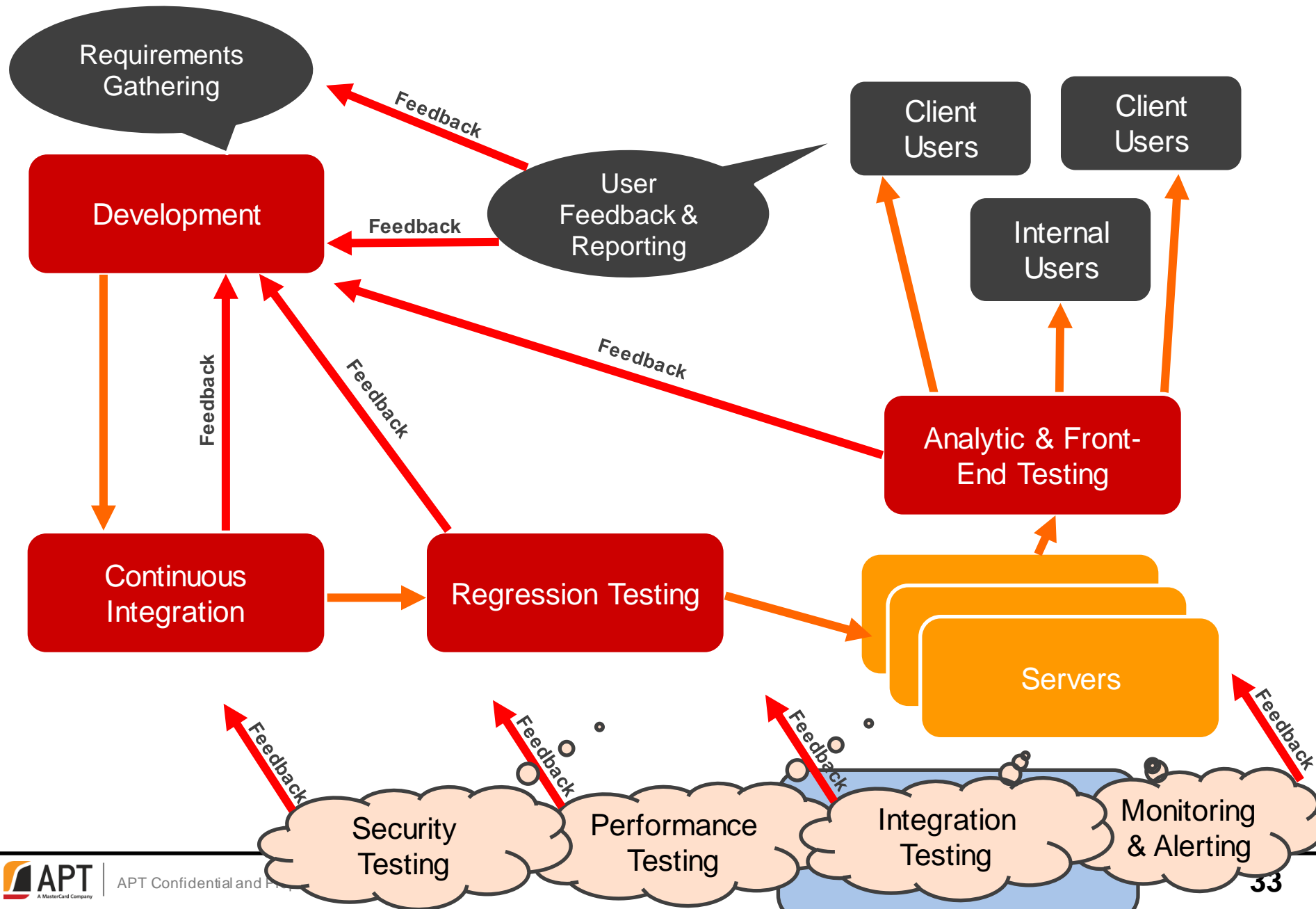
Performance test: 1k blackout dates

Look by





# Quality is built-in everywhere throughout development process



# Front-end Integration Tests

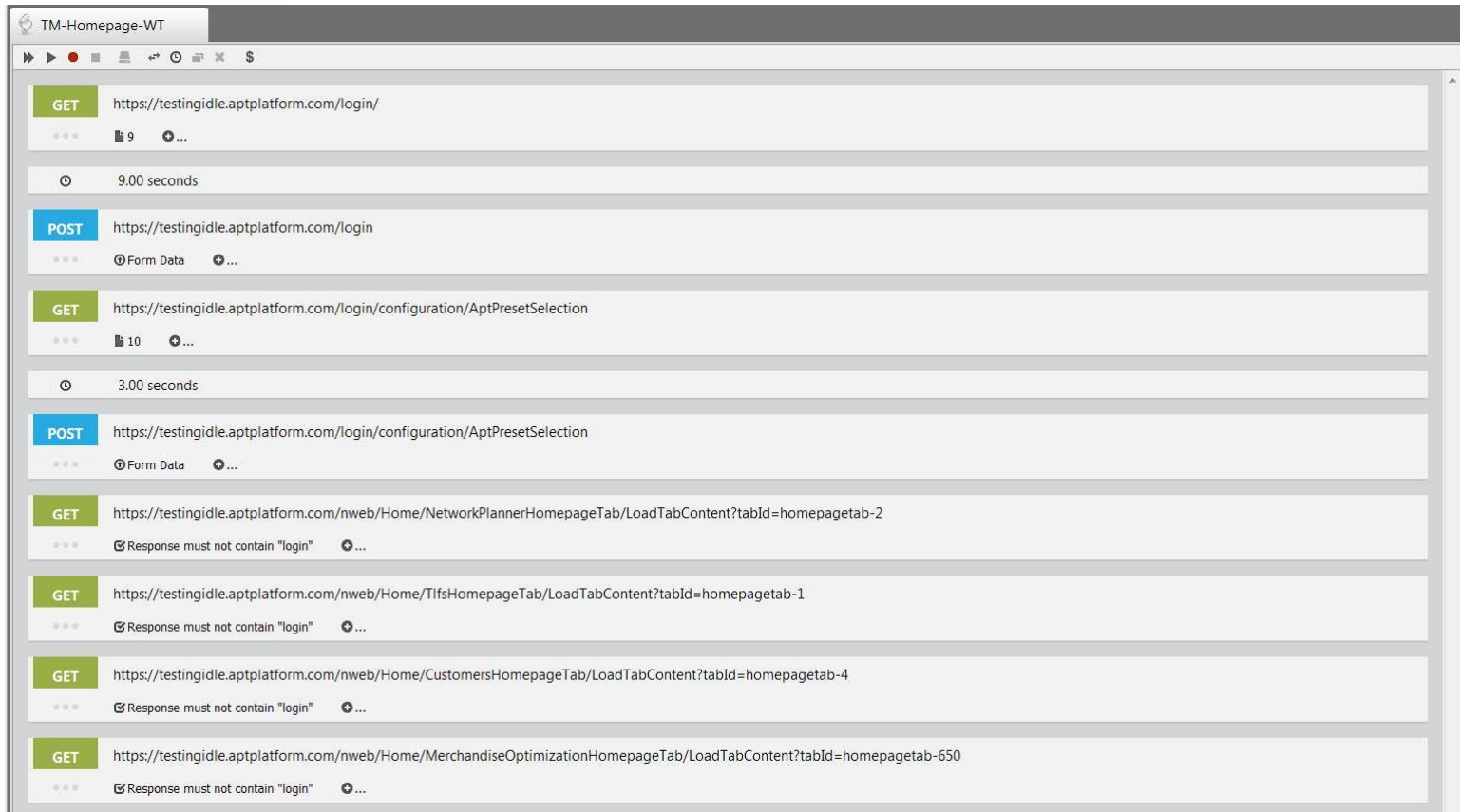
---

Test against live versions of the application

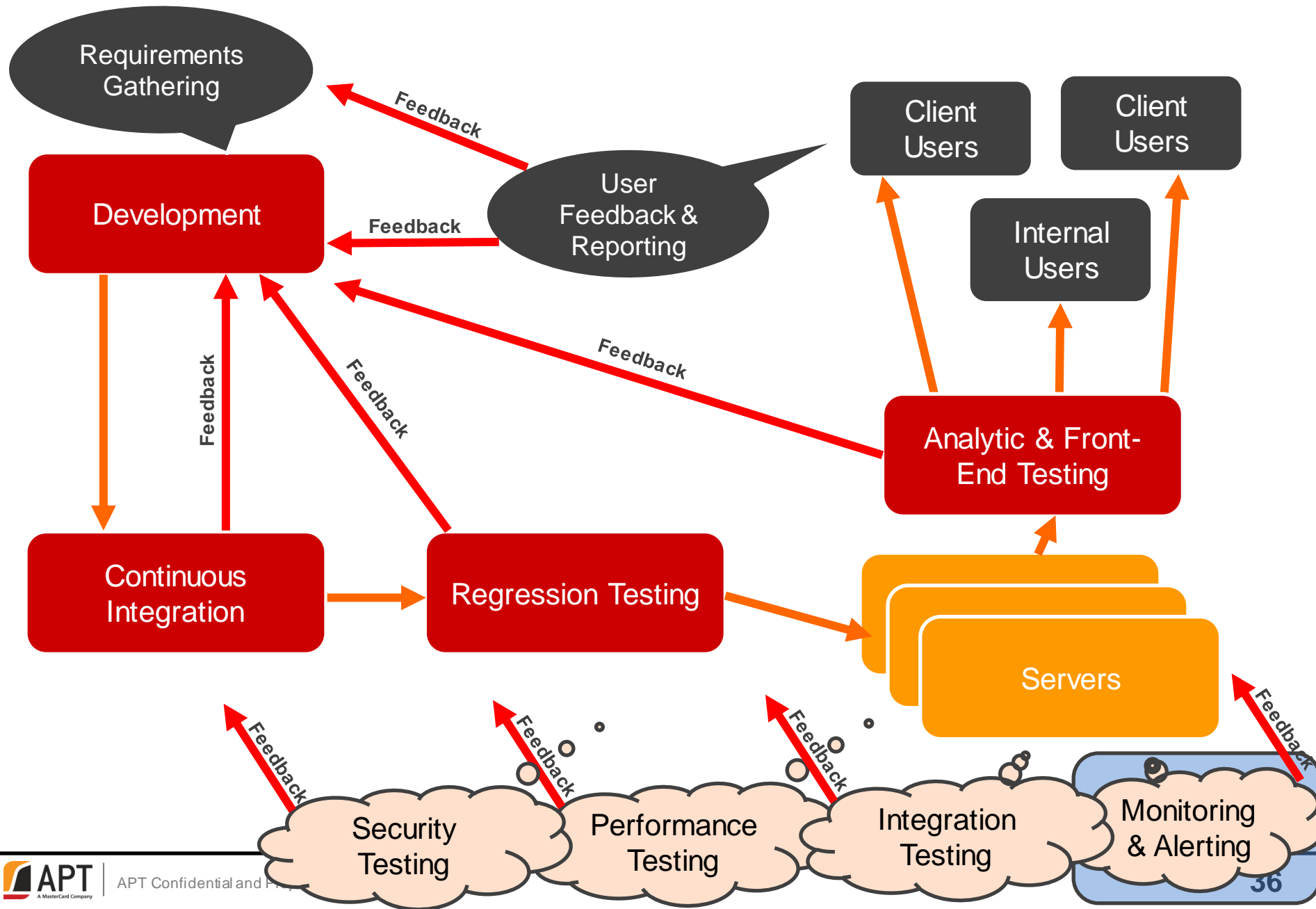


# Load Testing

## Test against live versions of the application



# Quality is built-in everywhere throughout development process



# Monitoring and Alerting

## Extensive monitoring of production system minimizes downtime

pagerduty

Incidents

Alerts

Configuration ▾

Analytics

INCIDENTS > INCIDENT #28476

CRITICAL: 'Health Checker Clustering Production-Boots Blue' on 'CLFUNC01QB' [Edit](#)

Status **Resolved**

Incident Times Open from Aug 23, 2017 at 7:23 PM to Aug 23, 2017 at 7:25 PM (for 2m)

Urgency **High**

Severity **Critical**

Incident Key event\_source=service;host\_name=CLFUNC01QB;service\_desc=Health Checker Clustering Production-Boots Blue

Impacted Service [Icinga Devops Q API](#)

Integration [Icinga](#)

Source CLFUNC01QB

Details

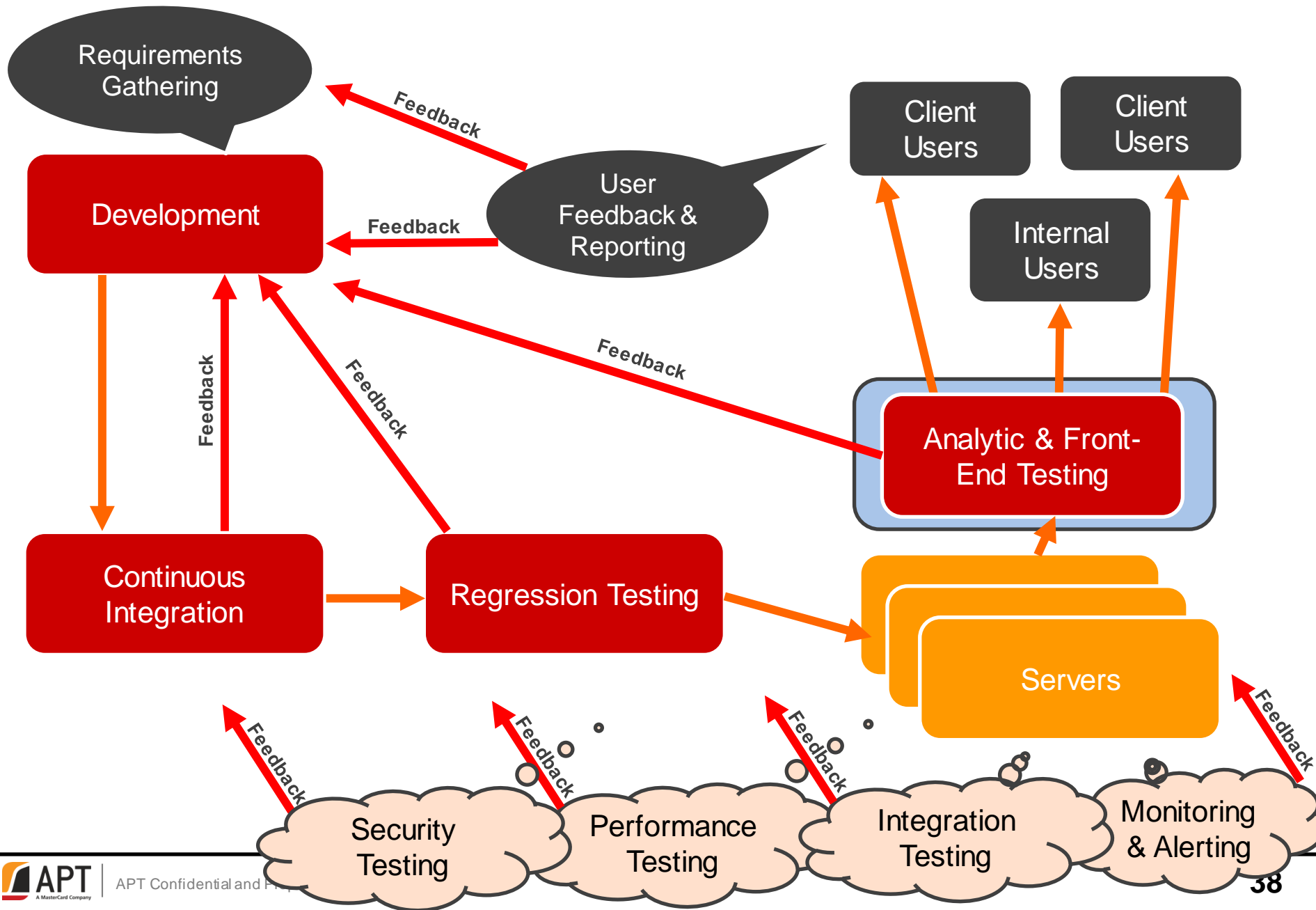
Response

Timeline

Time	Activity	ⓘ SHOW ALL DETAILS
on Aug 23, 2017 at 7:25 PM	<b>Resolved</b> through the API. Host: CLFUNC01QB ( <a href="#">View Message</a> )	
on Aug 23, 2017 at 7:24 PM	Notified Laurance George via email at <a href="mailto:lgeorge@aptnail.com">lgeorge@aptnail.com</a> .	
on Aug 23, 2017 at 7:24 PM	Notified Laurance George via push notification to Pixel.	
on Aug 23, 2017 at 7:23 PM	<b>Triggered</b> and assigned to 1 person.	ⓘ SHOW DETAILS

Per Page: 25 ▾ 1-4 >

# Quality is built-in everywhere throughout development process



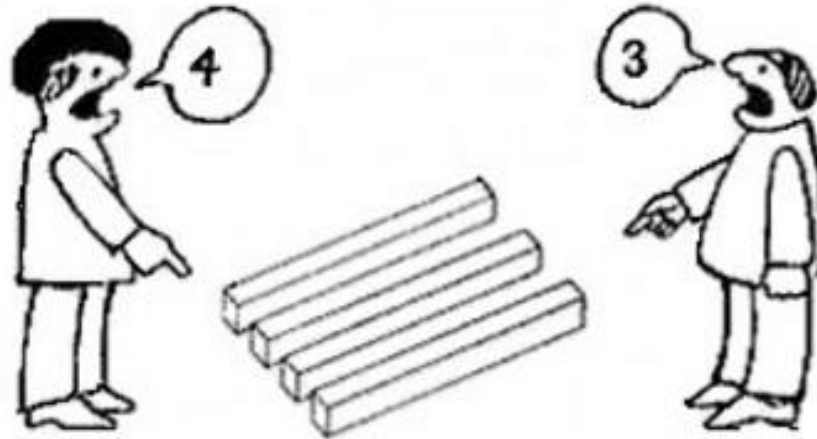
## Extensive test planning and review lead to more effective testing

86.73% after first 9 tests.



# Front-end & analytic testing

Testing Blitzes allow from testing from multiple perspectives





# Front-end & analytic testing

## Analytic validation is essential to providing a trustworthy tool



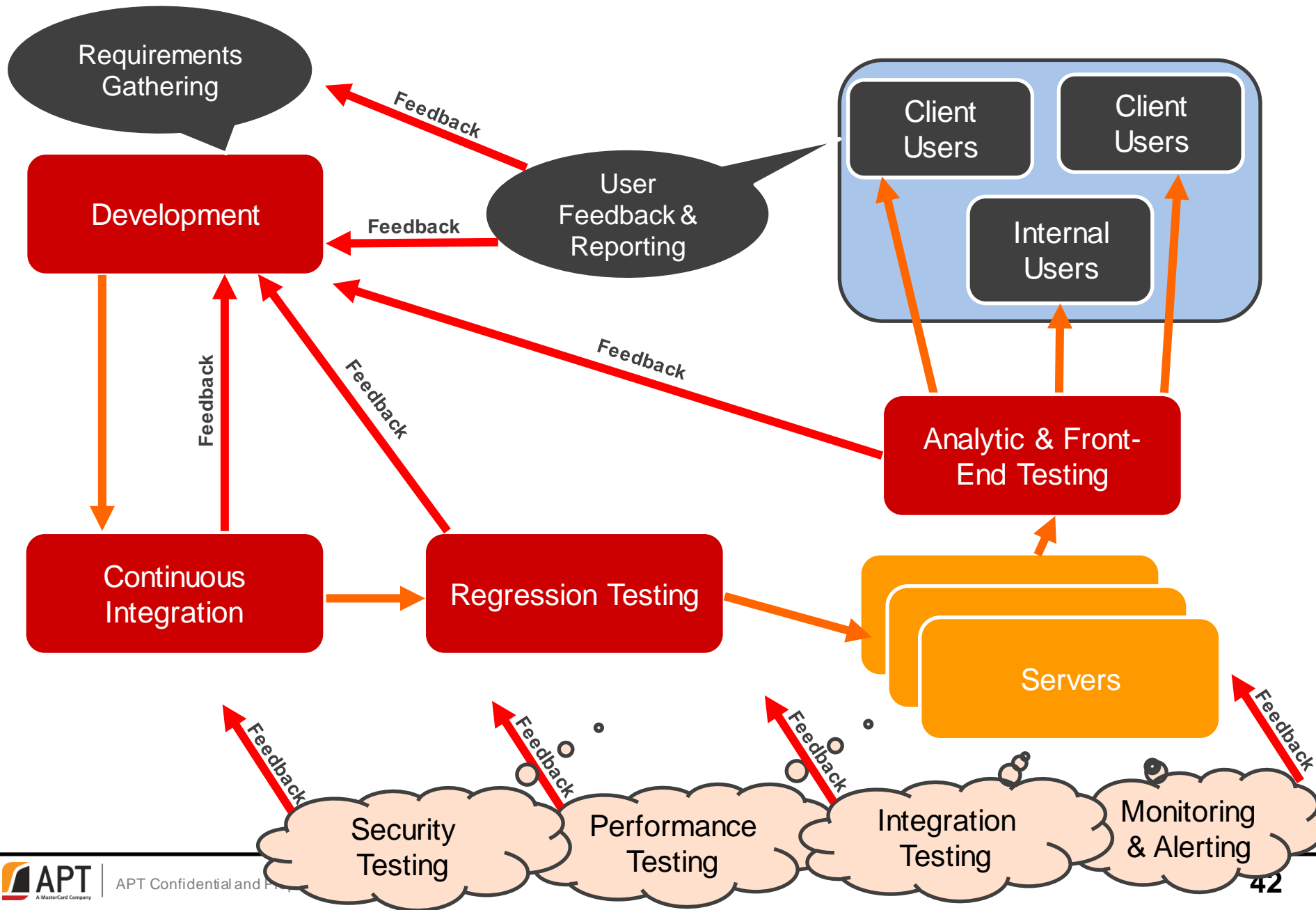
Microsoft®  
**SQL Server®**

### Analytic Test Plan

Math changes: Expectation, impact, and lift are calculated from the already aggregated top-level this and last year pre- and post-period values.

Analytic							Automated
Scenario	Tester	DB	Description	Expected	Actual	Status	Regression test
9	@Alex Berkow	Regression	Normal met/cat	New analysis numbers tie out	No aggregate: <a href="https://testing.aptpatform.com/nweb/ActivityOutputs/7227?OutputsDashboardViewModel.OpenOutputId=46999">https://testing.aptpatform.com/nweb/ActivityOutputs/7227?OutputsDashboardViewModel.OpenOutputId=46999</a>  Aggregate: <a href="https://testing.aptpatform.com/nweb/ActivityOutputs/7228?OutputsDashboardViewModel.OpenOutputId=47002">https://testing.aptpatform.com/nweb/ActivityOutputs/7228?OutputsDashboardViewModel.OpenOutputId=47002</a> > Click here to expand...  SSLY analytic cases.xlsx		2785109
10	@Alex Berkow		UDM Met/Met	New analysis numbers tie out	No aggregate: <a href="https://testing.aptpatform.com/nweb/ActivityOutputs/7500?OutputsDashboardViewModel.OpenOutputId=48491">https://testing.aptpatform.com/nweb/ActivityOutputs/7500?OutputsDashboardViewModel.OpenOutputId=48491</a>  Aggregate: <a href="https://testing.aptpatform.com/nweb/ActivityOutputs/7345?OutputsDashboardViewModel.OpenOutputId=47613">https://testing.aptpatform.com/nweb/ActivityOutputs/7345?OutputsDashboardViewModel.OpenOutputId=47613</a> > Click here to expand...  SSLY analytic cases.xlsx		2811254
11	@Alex Berkow		UDM Met/Atr	New analysis numbers tie out	No aggregate: <a href="https://testing.aptpatform.com/nweb/ActivityOutputs/7407?OutputsDashboardViewModel.OpenOutputId=47911">https://testing.aptpatform.com/nweb/ActivityOutputs/7407?OutputsDashboardViewModel.OpenOutputId=47911</a>  Aggregate: <a href="https://testing.aptpatform.com/nweb/ActivityOutputs/7406?OutputsDashboardViewModel.OpenOutputId=47910">https://testing.aptpatform.com/nweb/ActivityOutputs/7406?OutputsDashboardViewModel.OpenOutputId=47910</a> > Click here to expand... > Click here to expand...  SSLY analytic cases.xlsx		2817476

# Quality is built-in everywhere throughout development process

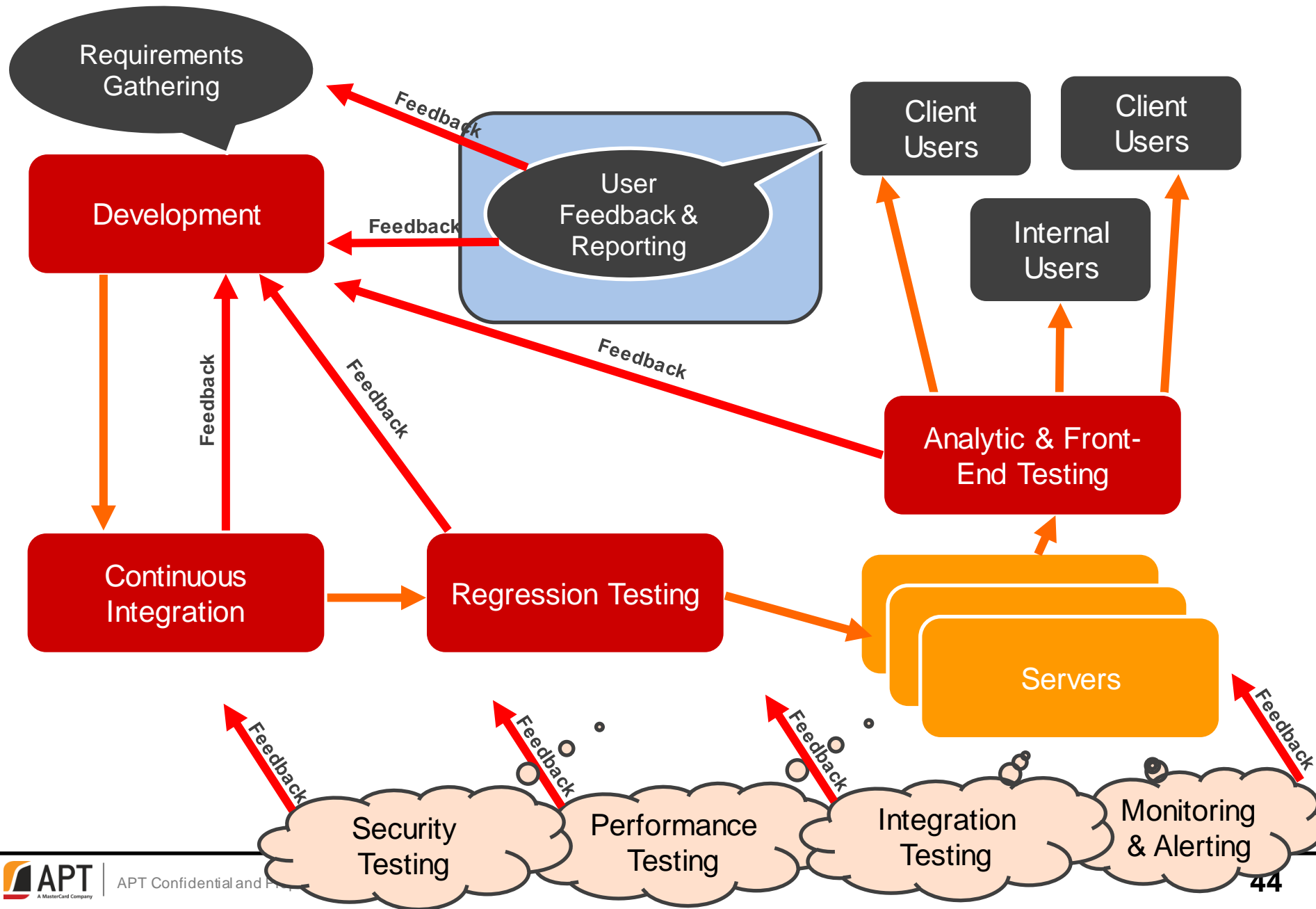


# Front-end & analytic testing

## Multiple deployment levels allows for internal user feedback



# Quality is built-in everywhere throughout development process



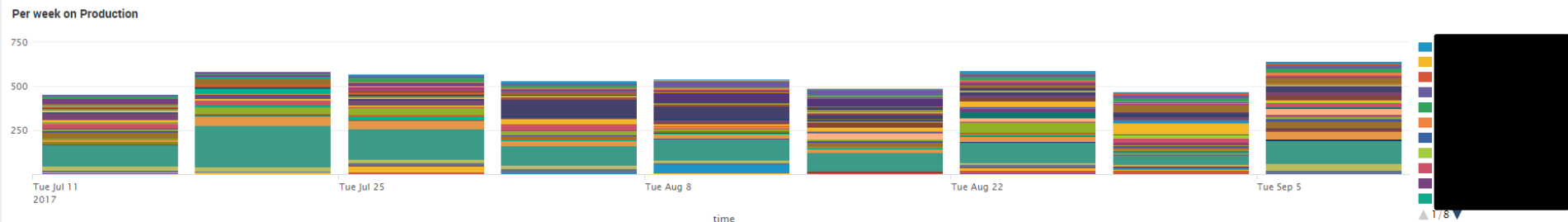
# User Feedback and Reporting

Monitoring usage and collecting user feedback helps close the loop between development team and client

## General Usage

Settings page hits in last 90 days					Number of Distinct Dashboards Visited, 90 Days				Market Basket Runs on Dashboards		
Staging and Production					Staging + Production				New vs Existing Dashboard (Past 90 Days, Production)		
userName	dbName	numHits	lastRun	isClient	userName	dbName	isClient	NumDashboards	Database	NewRuns	ExistingRuns
		55	09/11/17 18:42:45	Client User			APT user	1		6	
		85	09/11/17 17:52:55	Client User			APT user	1		6	
		11	09/11/17 17:29:59	Client User			Client User	1		7	
		113	09/11/17 17:08:51	Client User			Client User	1		4	
		12	09/11/17 17:04:38	Client User			APT user	1		26	
		122	09/11/17 16:57:36	Client User			APT user	1		5	
		1479	09/11/17 16:50:59	Client User			Client User	1		1	
		97	09/11/17 16:48:34	Client User			Client User	1		1	
		6	09/11/17 16:21:08	Client User			Client User	1		15	
		66	09/11/17 16:19:27	Client User			APT user	1		1	
« prev 1 2 3 4 5 6 7 8 9 10 next »					« prev 1 2 3 4 5 6 7 8 9 10 next »				« prev 1 2 3 4 5 6 next »		

Total BI runs since full release





**Q & A**