**APT – Applied Predictive Technologies**

Product Management team develops requirements -> Engineers build software to spec -> QA team confirms software is high quality -> Software delivered to users

Prototyping

Usability Sessions

Requirements testing

Quality is built-in everywhere throughout development process.

Quality is shared amongst all developers. Numerous developer practices drive software quality.

Pair programming, Test driven development (more difficult, but forces that each code has specific purpose)

Continuous integration tools ensure all changes are merged and testing together frequently (Jenkins, Bamboo).

Automated Regression Testing.

Manual test only once, but test automatically forever (n unit, Jenkins, sonarqube (can get coverage %).

Pairwise coverage (can’t test everything, so test every possible pairs).

Configuration Management.

Server consistency is a key driver of software quality (Chef (installs software across all servers)).

Vulnerability Scanning.

Testing for security vulnerabilities is essential when dealing with sensitive data.

Performance Testing.

A functional application is only valuable if it is performant (Web Page Test, splunk, n unit).

Front-end Integration Tests.

Test against live versions of the application (Selenium).

Load Testing.

Test against live versions of the application (Loadster).

Monitoring and Alerting.

Extensive monitoring of production system minimizes downtime (Pagerduty).

Front-end & analytic testing.

Extensive test planning and review lead to more effective testing.

Testing Blitzes allow from testing from multiple perspectives.

Analytic validation is essential to providing a trustworthy tool.

User feedback and reportings.

Multiple deployment levels allows for internal user feedback.

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