RRD LOGISTICS IT

Development 2016 Year in Review

> "There are two parts to learning craftsmanship: knowledge and work. You must gain the knowledge of principles, patterns, practices, and heuristics that a craftsman knows, and you must also grind that knowledge into your fingers, eyes, and gut by working hard and practicing."
>
> Robert C. Martin, Clean Code: A Handbook of Agile Software Craftsmanship

WHAT A YEAR…

As the demands for IT to find ways to provide critical solutions quicker, I cannot over state how important it continue to improve our processes, tooling, and skills.

As 2016 comes to a close, I wanted take a moment and reflect on what we have accomplished this year. It is amazing on how much has been done. The road has seemed long at times, but huge progress has been made in our pursuit to being the most advanced agile development team.

So, to start off, we have been fortunate to have really good partners, both in India and here in Bolingbrook. We have worked closely with Saggezza, IDC, Polaris and Microsoft. From training, delivering of projects, and to onboarding new technologies. We could not have been successful without these partnerships.

Here are some highlights of our success.

A safer place…

This year we deployed the open sourced Identity Server v3 from Thintecture. The server allows us to implement SSO with modern security protocols, using OpenID Connect and OAuth2.

A technology that supports a wide range of clients like mobile, web, and desktop applications and is extensible to allow integration in new and existing architectures. The server provides a stop-gap measure allowing us to bring modern application security to our apps today, while the infrastructure teams build out a corporate version.

One benefit of using a claim based approach to security, using claims has allowed us to successfully implement feature toggling. This ability allows us to deploy to production, maintaining our agile cadence, while allowing our business partners to determine how it is released to our customers.

The team has created a .net library includes all of resources needed to implement the new login and security features needed for our projects.

Cleaning up our act…

The UX team has been hard at work this year helping us look good. With projects leveraging MVC 5, responsive applications became a reality out of the box. Using the Bootstrap's support for theming, the UX team created a Logistics theme, based off of their work done on the CRM project. This allowed the designers to develop a common design philosophy that is now used in all of our green field projects. Our applications have never looked so good. The new designs have reduced the complexity, helping the teams to deliver value sooner than later

The UX team has also built a .net library to that implements the UX Bootstrap theme, along with pre-approved jQuery objects. Gone are the days when a designer would provide the HTML file for a page. Now, the development team can rapidly build out a clean UI that is fully integrated into the project during the sprint. Some teams have even starting implementing Lean UX concepts into their agile practices.

A user interface is like a joke. If you have to explain it, it's not that good.

Many resources and training sessions have helped to expedite onboarding the UX helper library and theme. There are code examples included with the library that shows how to use most of the classes. To help with understand the theme, a Style guide website has been set up at http://insidelogistics-dev.rrd.com/RRDBSTheme/

But does it work…

There have been several advancements made this year that increase both the speed and quality of what we delivered.

Our International team leveraged Release Manager in Visual Studio Team services(VSTS). For the first time on a Logistics' project, implemented Continuous Integration (CI), automatically deployed to their non-prod environments. From checking in code, CI, deploying to the dev, qa (with Selenium/SpecFlow tests) and staging boxes have been completely automated.

QA was a big focus this year. Integrating QA into the sprint and automating tests has been the single biggest impact in achieving the agile dream.

Transforming from the traditional manual testing to automated tests was a fundamental change. Rather than test plans and extensive documentation, QA has become a dynamic, adaptable and easily repeatable code driven process.

All code is guilty until proven innocent.

We now literally have a shippable product at the end of each sprint. The International team completed 13 deployments to production this year, without a UAT week. Many people have worked to make automated functional test a reality. The impact is huge.

For our business partners, it frees them up to focus on driving value. Our Product Owners have been able to devoted to authoring new user stories and planning next set of features.

From April 1, 2015 to April 30, 2016 we tracked the number of SEV 1 tickets for 28 projects.

"Testing has to be an integral part of developing software and not a separate phase. When this approach is taken, product quality is owned by everyone on the team. It is easy to state, but hard to put into practice because of long-standing preconceived notions that developers and testers are better kept apart."

James Sivak

> “Focus and simplicity. Simple can be harder than complex: You have to work hard to get your thinking clean to make it simple. But it’s worth it in the end because once you get there, you can move mountains.”
>
> STEVE JOBS
> [BUSINESSWEEK, MAY 25, 1998]

So, what’s the impact.

The result for all projects:

- 155 SEV 1 tickets
- Average number of tickets per project: 5.54
- Most tickets for a project: 43
- Least tickets for a project: 1

Two projects used QA, TDD and SOLID:

- CRM had 1 ticket
- Pricing had 3 tickets

** not included in the tracking was XPOTrack. From Feb 2016 to December 2016, the project only had 1 SEV-1 ticket.*

The number support tickets for applications using QA, SOLID and TDD were significantly lower than ones not using these techniques.

A SOLID foundation…

We continued our second year of training on SOLID and TDD. The combination has proven to reduce the complexity of maintaining our applications. Keeping things simple has built a foundation for us to mature in our agility as a development team.

We matured our agile practices in areas such as:

> *“When in doubt, simplify.” - Eric Ries, The Lean Startup*

Collective ownership: No one person owns or is responsible for individual code segments.

Continuous Deployment: Software builds goes through the pipeline and automatically deployed.

Continuous Integration: Software builds and unit tests are completed every time code is checked in.

Coding Standard: All team members write code in the same way, using the same styles and formats.

Simple Design: The best design is the easiest one that works.

Rise above the clouds…

This year we began moving our applications to the cloud. Moving to Azure opens up new solutions, opportunities and efficiencies that were not possible before.

Here is a list of technologies we starting using in Azure this year that helped us be successful.

Visual Studio Team Services – the SaaS version of Team Foundation Server.

Release Management - A service in Visual Studio Team Services and Team Foundation Server that helps you automate the deployment and testing of your software in multiple environments.

Machine Learning – A tool to model predictive analytics as well as a fully managed service you can use to deploy your predictive models as ready-to-consume web services.

Application Insights - an extensible analytics service that helps you understand the performance and usage of your live web application. It's designed for developers, to help you continuously improve the performance and usability of your app.

Service Bus – A cloud-enabled communication with enterprise messaging and relayed communication that helps you connect on-premises solutions with the cloud.

Redis Cache - Azure Redis Cache provides high throughput and consistent low-latency data access to power fast, scalable Azure applications. The following teams are using this technology

Work Roles - a Cloud Service Role is a collection of managed, load-balanced, Platform-as-a-Service virtual machines that work together to perform common tasks. Cloud Service Roles are managed by Azure fabric controller and provide the ultimate combination of scalability, control, and customization.

And that's a wrap…

I want to thank each person for your hard work in making 2016 an outstanding year of achievement.

Looking forward to see what we will do in 2017.

> "A company's ability to respond to an unplanned event, good or bad, is a prime indicator of its ability to compete."
>
> Bill Gates