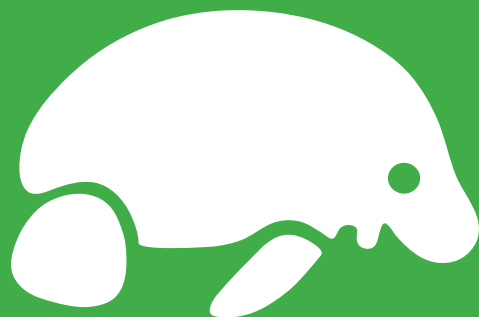


Terryberry Development Team and Studio 3T: A Case Study with ROI.



Studio 3T

“Be Recognized” ... for ground-breaking technology.

Terryberry celebrated their 100 year anniversary in 2018. For a century the business has been pioneering employee recognition programmes and awards and their slogan is naturally, “Be Recognized.” Also fittingly, as the direct links between workplace well-being, recognition and achievement have been widely demonstrated over the last few decades, Terryberry has evolved from a manufacturer of awards and trophies into a leading cloud service provider.

In 2009 under the leadership of 4th generation managing partner Mike Byam and CIO Dan Egan, Terryberry launched their prototype of a hosted platform that could manage all the different elements of an in-house employee recognition program, in a single sign-on environment. Hence the name: 360 Recognition. Full visibility. The Development Manager Dan Cumings joined the team in 2010 and has been helping Terryberry push the innovation envelope for nearly a decade.

Scaling Issues

“We came from a fairly traditional relational data background,” explains Dan, “So when we started out, we built it on SQL Server. All was good to start, but by the time we got up to fifty customers and more, each with some thousands of their staff logging into the system concurrently, we ran into scaling issues.”

The stored procedure that the 360 Recognition platform relied on for loading the Recognition Feed for each of the customers had gotten progressively slower and finally started locking up. Dan and his team looked around for a database that could address the rapid scaling and performance issues they were facing. MongoDB looked like a promising candidate for two reasons.



Right First Time

“One was performance. The other was that MongoDB was so much easier to develop with. C# is our main language and in MongoDB it’s very simple. You just store your user info in a C# object and then you pull it out when you need it again. MongoDB does all the work for you. Once we discovered that efficiency and ease-of-update, we quickly started switching everything over to it.”

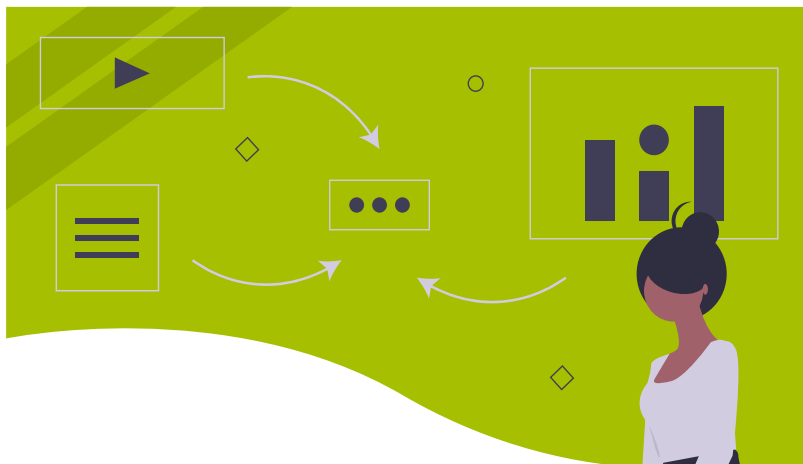
As Terryberry migrated their first customers from SQL to MongoDB, the benefits were immediately apparent: application responsiveness and ease of deployment.

“We wanted to get it right first time, every time.”

“Our whole shop was on SQL Server originally and so it took a good 6 months before we were able to do the first customer site. Once we’d templated it, it took another 18 months to migrate everyone else over. We wanted to get it right first time every time, so we decided to do just one customer at a time.”

Within eighteen months all their existing fifty or so customer sites had been migrated over to MongoDB. But their growing ability to develop faster and deliver new features and benefits to customers meant growth was accelerating.

They are now serving hundreds of thousands of users all over the world, delivering recognition programs for anything between 50 staff and 30,000 employees on each customer site. Scalability is not optional - it’s integral to the service.





Team Tools Make it Work

“I’ll wager with you that if we went and asked right now, everyone on the team will have Studio 3T open on their desktop. We all use it *constantly*, it’s used as extensively as Visual Studio. We use it for all sorts of things.”

- Checking that our data is as we expect it to be
- Writing and running new mongoshell queries
- Building aggregations in Studio 3T, which we can then later write up in C#, or use Query Code to generate the equivalent driver C# code for.
- Writing and running queries in SQL. Naturally we’re comfortable in JSON but sometimes it’s great to use SQL.
- Confirming data after we run new application code, e.g., “Hey, how did that new app code change the data?”
- Test manipulations of the data, e.g. setting a temporary status on a field so we can rerun a test.
- Bulk data and/or schema updates. There are many different ways to update the collections and documents. So for example, we might want to say, “For this iteration, update this one field value for all the documents.”
- Renaming, editing and removing fields.
- Depending on what we’re looking at, we may need to swap the data view between Table, Tree and JSON views.

“I’ll typically stick to the Tree-view if there are a lot of nested fields,” says Dan Cumings. “But if I’m debugging, and I might have to drill-down deep within some other object, often I’ll switch to the Table view and drill down that way. Then I’ll find the row I’m on and reverse out to find which document I’m dealing with. Then we’ll switch to JSON view if I just want to understand what the underlying data really looks like. So we use all of the views - Table, Tree and JSON - depending on the job in hand.”

“The phenomenal growth of 360 as a hosted service over the last few years has meant that we’ve had to step back and ask ourselves, “So how do we enable the next phase of growth?” And we’ve made the big decision to go the microservices route.”

Dis-Assembling the Monolith

The development team at Terryberry is lean. It is comprised of a team of software developers, QA, as well as a support team who also take care of on-boarding customers, maintaining the programme, rolling out new features and and sharing with customers best practices and tips.

Dan Cumings describes their method as 'agilish' - encompassing sprints and issue tracking using JIRA, and regular scope meetings with business stakeholders, and the overall leanness and speed of development means that they've yet to see a need for a dedicated DBA or even someone with DevOps in their job-title:



"It doesn't feel really necessary right now. MongoDB is simple enough to understand and work with. And Studio 3T makes it really easy to navigate around the data. We're using a hosted, paid-for subscription on MongoDB Atlas. They do all the management and the sharding and all that stuff. And when we switch to regional provisioning, they can handle all that too, which is a huge relief."

"Where you could say we are more Agile, is that we've got a bespoke system for deployments, which minimises the impact on customers. We have a monolithic application. So we do a lot of heavy testing before any release. Then we give it to QA who give it another hammering."

"And Studio 3T makes it really easy to navigate around the data."



And the Rise of the Microservices

“Lately, in one of our main projects this year, we’ve been separating out functional chunks of this monolithic app and turning them into microservices. These we can release much more frequently, in a much more resilient and agile way. We’re a couple of months away from the end of a 12 month project to to break up the monolith into microservices.”

“Depending on which features customers have enabled, the new microservices architecture maps, as practically as possible, on to the subscription modules.”

Studio 3T tools used by the Terryberry Development Team

Visual Query Builder
Table, Tree and JSON views
Edit data in-place
Aggregation Builder
Import/Export

Index Manager
Intellishell
Query Code
Data Compare & Sync
Tab and connection coloring

“The thing that’s pretty unusual is that we’ve got several *thousand* collections. We’re kind of switching away from that approach, but in the past we gave each customer its own ‘wall-post’ collection. A social feed is what we call a wall-post and each one is its own collection.

“Studio 3T definitely saves time, when considering that we all work in it all the time. Even just finding that a particular collection exists is pretty useful. So the collection would be prefixed with the customer’s name. Go to that prefix and you’ll get all the collections for that customer. When you have several thousand collections, it’s crucial to be able to find stuff fast. Studio 3T is a lifesaver for that.”

“When you have several thousand collections, it’s crucial to be able to find stuff fast. Studio 3T is a lifesaver for that.”



Productivity boost enjoyed by Terryberry team:

While different team members use different features with varying frequency, the following are survey answers from 4 team members to the question:

“How many hours per week (on average) do you save with Studio 3T?”:

- Dan 1 - 5 hours
- Sr Developer 1 5 - 10 hours
- Sr Developer 2 7 hours
- Sr Developer 3 10 - 15 hours

Altogether an average of **7.5 hours is saved per week by each developer** on the team, or roughly a day a week per developer.

A quick ROI calculation:

Studio 3T licensing per developer (7 x Core, 2 x Pro): \$3.33 per day per developer
Average US software engineer salary : **\$400** per day

Each developer saves a day per week when using Studio 3T, the 9 strong team derives 9 x \$396 per week in productivity value: **\$3,564**

Annually that adds up to a **\$185,796** saving on the bottom line.