



# Delivering BI Projects Using Agile

SYED RAYHAN, Accenture

GURUSHYAM MONY, Department of Taxation, VA

SCOTT NEWBY, Z Inc

---

Are you wondering how you could deliver your BI project using Agile? In this experience report, we share our experiences building a BI solution using Agile. Our hope is you will be able to tackle your next BI project using Agile and enjoy similar successes as you see from Web and Mobile projects.

---

## 1. INTRODUCTION

Unlike Web or Mobile projects, why do BI projects typically struggle to adopt the principles of agile methodology? In retrospect, in our project we realized the following challenges.

1. Lack of clear data ownership and governance
2. Correct definition of data can be difficult to capture
3. Lack of consistent development patterns for BI artifacts
4. Lack of test automation tools to drive TDD practices
5. Lack of CI (Continuous Integration), early CD (Continuous Delivery) practices
6. Constant pushes and pulls between immediate needs and long term priorities for customers

Traditionally waterfall seems to be the preferred approach to manage BI project challenges. It may cost organizations a lot of time and money to realize substandard or inadequate BI solution.

Faced with above challenges, we learned and matured over time in adopting Agile methodology to deliver our BI solution. In the end, we managed to incrementally deliver working BI solution to our customers. In this paper, we will share our journey with you. We'll broadly divide our journey into three phases. In each phase, we share things we did that went well, and things that needed improvements as we progressed through each sprint. Our goal is to help you, recognize similar challenges and adopt Agile principles to deliver working BI solutions that meet your customer's expectations.

## 2. BACKGROUND

The Department of Motor Vehicles (DMV) serves approximately 5.6 million drivers and ID holders, and 7 million registered vehicle owners. The DMV wanted to see its data better used to help reduce the number and severity of traffic accidents in the state. But that idea required efficient and timely data capturing, improved data accuracy, integrating with databases at other state agencies, and flexible reporting accessible by the right people across many state agencies. The department's aging mainframe couldn't do all that.

So, the department created a custom solution called Traffic Record Electronic Data System (TREDS) to integrate its data with third-party and custom software data provided by both the Department of Transportation and the Virginia State Police. The result enabled better data-driven decisions such as pinpointing locations for better signage, efficiently deploying police patrols at the right locations during the right time of the day, implementing corrective actions to reduce accidents on the road, adopting right policies,

---

Author's address: Syed Rayhan, San Francisco; email: [srayhan@gmail.com](mailto:srayhan@gmail.com)

Second author's address: Gurushyam Mony, 5821 Maybrook Court, Glen Allen, VA; email: [gurushyam.mony@gmail.com](mailto:gurushyam.mony@gmail.com)

Third author's address: Scott Newby, 14406 Camack Trail, Midlothian, VA; email: [scott.newby74@gmail.com](mailto:scott.newby74@gmail.com)

Copyright 2016 is held by the authors.



### 3.1 Phase 1

The initial backlog in this phase had stories each representing an entire report with broad acceptance criteria. Struggling to break the stories by some meaningful business criteria, we broke them further by development, testing, and release efforts. For example,

For a feature of completing Crash Facts 2009 that contained many BI reports, the team broke it down into three epics,

1. A development effort represented as a story such as “I as TREDIS developer would like to develop the reports needed for Crash Facts 2009 so that they can be provided to Highway safety office for testing.”
2. The testing effort was represented such as story such as “I as a TREDIS tester would like to test the reports needed for Crash Facts 2009 so that they are of good quality,” and
3. Release effort of reports to production such as “I as a TREDIS developer would like to publish Crash Facts 2009 to the website so that customers can access it.”

These stories were sized independently, and consumed in different Sprints. Most of the development, and testing Product backlog Items (PBI's) were epics. These epics were never broken down during backlog grooming sessions, nor when they were prioritized into the Sprint backlog.

The acceptance criteria for the epics were simple, such as “Duplicate the Crash Fact 2009 reports published from TREDIS application.” Single definition of data for developing the reports was difficult to ascertain. Product Owners were not actively engaged with the development team to be able to provide clarifications in a timely manner. Sizing and tasking of stories was challenging, as the scope of the PBIs was broad and vague. The team followed story point estimations for sizing stories, but relative estimation was hard as the team did not have any mechanism to compare stories.

The acceptance testing of the stories remained as a trailing sprint's product backlog item to work on. Automated acceptance test practices did not exist and all levels of testing were manual efforts. The testing strategy was to use already published production reports to new reports generated from the TREDIS application and compare to find anomalies. There was no Continuous Integration setup.

The team was operating more in a mini Waterfall fashion within the Agile framework. As Sprints progressed, the backlog was filled with three categories of items - 1. PBI's for new reports to address immediate customer needs, 2. Bugs found during Acceptance testing, and 3. Production bugs raised by the customers as they trusted the existing published reports to be the single definition of truth. The DMV customers did not want published reports in production to report different results when the reports were accessed at different timelines.

All of these challenges led to constant de-scoping of stories. We were unable to deliver our Sprint commitments regularly. We also had mounting frustrations in having to juggle between prioritizing fixing bugs and delivering new reports in any given Sprint.

### 3.2 Phase 2

By early 2009, the team started to recognize patterns across new reports. The team observed that these reports were similar in nature such as “I as a DMV Analyst want to report the Motorcycle Fatalities by time so that customers can view that data” and “I as a DMV Analyst want to report the School bus fatalities by time so that customers can view the data.

To address the challenges mentioned in phase 1, the team during this phase, decided to stop the mini Waterfall within the Agile Sprints, and started breaking down the epic stories along common dimensions and measures observed. Stories defined in the backlog were written such as “I as a DMV Analyst want to report the Crash Fatalities by time so that customers can view that data”. A standard acceptance criteria template that contained details for the product owners to define was developed to avoid ambiguity in the definition of data. The acceptance criteria template would capture details such as a. pre-conditions, b. definitions of any measures included, c. conditions that classifies a crash time, d. any UI mock-up as necessary or web front-end development etc. Based on these patterns observed, we developed dimensions and measures common across all reports first. This helped reducing rework. Implementing measures and higher priority dimensions allowed the team to demonstrate the capabilities of data warehouse to our product owners. The product owners were then able to define their acceptance criteria better for the reports.

If during estimation it was determined that a PBI was an epic, we broke them down such as 1. Stories for developing dimensions, measures, and queries needed for the BI solution 2. Stories to design the front-end reports integrating the dimensions and measures needed to derive the data definition.

## Crashes by Primary Street - Report

**Date:** June 16th  
**Layout:** Landscape  
**Expected Output Format:** Excel/PDF  
**Page Size:** 8.5 inches width/11 inches height  
**Margins:** .25in top/bottom/left/right  
**Title of Report:** Crash Count by Primary Street  
**Link name :** Crashes By Primary Street  
**Banner Image:** No  
**User Filters (User Input):**

Filter Caption	Filter Values	Default Value	Source	Required	Multi-Select	Comments
Time(Year)	Populate from Data ware house	Between '2012' To Getdate()	Crash Year from Treds Star -> Time table	Yes	No	NA

**System Filters (Static filters across entire report):**

Filter Description	Filter Values	Source	Comments
Department	NA	User logged department Id.	NA

**Include Selected Filter Values on Report:** Year  
**Include Report Executed Date & Time:** Yes  
**Include Page Numbers:** yes  
**Use Conditional Formatting:** Yes (Gold/White)  
**Include Grand Total:** No  
**Sort By:** Primary Street Name  
**Output Field List (example included):**

Field Caption	Field Format	Order	Calculation	Header Style	Data Style	Source	Comments
Primary Street Name	Text	1	None	Bold	Normal Text	Crash Snap Shot - VT Street, VT Location	If any
Crash Count	Text	2	None	Bold	Normal Text	Crash Snapshot - Crash	If any

**Web Roles given Access to this Reports:** Police Officer, Police Officer Supervisor, Law Enforcement Clerk  
**Report Server Folder:** Law Enforcement  
**Subscription:** No.

Figure 2: An example of TREDs Report Definition using a standard Acceptance criteria template

Definition of “Done” included both the development and the acceptance tests to be performed within the same Sprint. These allowed us to do better point estimations, as we were able to compare the stories better. Specific resources of the development team were allocated to handle production bugs in an effort to avoid all team from distractions that would impede sprint deliverables. This helped the team stay focused on the Sprint deliverables resulting in better Sprint results.

As we were delivering more and more reports and bi solutions based on our crash data views, we no longer could use the same testing strategy followed earlier (e.g., comparing a legacy report to new one manually). Plus, we were delivering to our customers in regular increments end of every 4 sprints. Test driven development practices were adopted to avoid rework and to reduce the number of bugs in the product backlog. Acceptance tests were defined for the customer to review as part of each sprint and the testing team ran them before certifying a story to be released into production. The data required to be able to run an acceptance test was designed as mocks. We did not have a test automation suite for our Data Warehouse yet.

The team conducted in addition to the acceptance tests, a separate user acceptance test phase before integrating the features for delivery.

Delivery of the BI solutions and reports were consumed as stories to be prioritized into a release/hardening Sprint. We had a 1-week of hardening Sprint after a 4 development Sprints before a BI solution increment was released into production.

Attribute of Vehicle Body Type Dimension	Scenario
<b>Passenger car</b>	Create a new dimension named Vehicle Body Type Contributing to the <b>Crash - Passenger car</b> When any vehicle involved in a crash has a vehicle type of Passenger car Then all persons involved in the crash are indicated as Yes for this new dimension. ELSE persons involved in the crash are indicated as No for this new dimension
<b>Truck - Pick-up/Passenger Truck</b>	Create a new dimension named Vehicle Body Type Contributing to the <b>Crash - Truck - Pick-up/Passenger Truck</b> When any vehicle involved in a crash has a vehicle type of Truck - Pick-up/Passenger Truck Then all persons involved in the crash are indicated as Yes for this new dimension. ELSE persons involved in the crash are indicated as No for this new dimension
<b>Van</b>	Create a new dimension named Vehicle Body Type Contributing to the <b>Crash - Van</b> When any vehicle involved in a crash has a vehicle type of Van Then all persons involved in the crash are indicated as Yes for this new dimension. ELSE persons involved in the crash are indicated as No for this new dimension
<b>Truck - Single Unit Truck (2-Axles)</b>	Create a new dimension named Vehicle Body Type Contributing to the <b>Crash - Truck - Single Unit Truck (2-Axles)</b> When any vehicle involved in a crash has a vehicle type of Truck - Single Unit Truck (2-Axles) Then all persons involved in the crash are indicated as Yes for this new dimension. ELSE persons involved in the crash are indicated as No for this new dimension
<b>Motor Home, Recreational Vehicle</b>	Create a new dimension named Vehicle Body Type Contributing to the <b>Crash - Motor Home, Recreational Vehicle</b> When any vehicle involved in a crash has a vehicle type of Motor Home, Recreational Vehicle Then all persons involved in the crash are indicated as Yes for this new dimension. ELSE persons involved in the crash are indicated as No for this new dimension
<b>Special Vehicle - Oversized (Veh/Earthmover/Road Equip.)</b>	Create a new dimension named Vehicle Body Type Contributing to the <b>Crash - Special Vehicle - Oversized (Veh/Earthmover/Road Equip.)</b> When any vehicle involved in a crash has a vehicle type of Special Vehicle - Oversized (Veh/Earthmover/Road Equip.) Then all persons involved in the crash are indicated as Yes for this new dimension. ELSE persons involved in the crash are indicated as No for this new dimension
<b>Bicycle</b>	Create a new dimension named Vehicle Body Type Contributing to the <b>Crash - Bicycle</b> When any vehicle involved in a crash has a vehicle type of Bicycle Then all persons involved in the crash are indicated as Yes for this new dimension. ELSE persons involved in the crash are indicated as No for this new dimension
<b>Moped</b>	Create a new dimension named Vehicle Body Type Contributing to the <b>Crash - Moped</b> When any vehicle involved in a crash has a vehicle type of Moped Then all persons involved in the crash are indicated as Yes for this new dimension. ELSE persons involved in the crash are indicated as No for this new dimension
<b>Motorcycle</b>	Create a new dimension named Vehicle Body Type Contributing to the <b>Crash - Motorcycle</b> When any vehicle involved in a crash has a vehicle type of Motorcycle Then all persons involved in the crash are indicated as Yes for this new dimension. ELSE persons involved in the crash are indicated as No for this new dimension

Figure 3: An example of dimensions grouped using similar patterns represented using a common acceptance criteria format

We started to perform better in terms of estimating our capacity and velocity and meeting Sprint deliverables by the end of 2012. As sprints progressed we gained maturity with BI solutions we delivered into production. This success was not enough for the team.

### 3.3 Phase 3

In 2013 we realized that our maturity did not translate to improved efficiency on business operations and user's ability to automatically harvest the data. We found out that our customers still did not grasp the full potential of a Data Warehouse. The team was motivated to solve this problem and analyze ways to be more effective.

The question became how do we design and deliver good solutions more efficiently to provide a greater positive impact to business operations and access to data. Our solution was to create consumable and impactful Sprint stories to increase the usefulness of the Data Warehouse, leading to increased client confidence and access to data. We also wanted to facilitate more effective Sprints.

So, we started doing the following things -

1. Started proposing stories such that they incrementally increased dimensional and fact coverage resulting in incremental Data Warehouse expansion. This would also mean reducing the number of reworks needed from story to story. Some of the examples were -
  - a. "I as a Highway safety office data manager would like to see the heat map index of crashes along major highways so that VA citizens can be made aware of high crash zones".
  - b. "I as a VA citizen want to see the fatal, injury or property damage crashes along all intersections of VA to determine risk score of intersections."
2. Established clearly defined data ownership and data governance within the department. For example, we set clear closure times that were controlled by the data owner. Once a reporting period was closed, data prior to that point could not be modified in the transactional system and effectively locked changes to data in that period. This was done to ensure consistent reporting from a consistent set of data. This also slowly eliminated the fear our customers had with respect to data integrity.
3. Followed business driven story definitions while writing stories to let our customers understand its value in the backlog. This gave the stories and epics with more direct business value and abstracted the technical aspects of delivering them away from our business users. While analyzing a story during Sprint planning, we derived the technical tasks required to meet the business need.
4. Started using prototyping as a way to help clarify the business needs in the minds of the customers. Sometimes, the business users will not fully grasp what they need or how we could generate certain insights. Prototyping stories helped resolved those situations instead of prolonged story discovery sessions.
5. Started collaborating with the business users in defining acceptance criteria and tests. It became part of the "Definition of Ready" for stories to size and commit to a Sprint. A template for providing acceptance criteria for stories was established that required business team to provide behavior specifications. One such example of acceptance criteria is  
"Pre-Conditions: Crash is final stage passed approval from crash processing office. Rule: If a crash has any driver, passenger or pedestrian and if any of the people in that crash were reported dead and if the crash has a vehicle that was classified as Commercial then that crash needs to be classified as commercial fatality" Other Conditions: Add dimension to Warehouse".  
Using a non-technical format for writing acceptance criteria which clearly captured pre-conditions, conditions and other details meant that the stories had clear scope, and were easier to verify and validate.
6. Developed and documented repeated patterns in building BI artifacts for our development team to follow. Some of these patterns were -
  - a. Follow standard report templates along with pre-defined tasks and steps,
  - b. Use a set of standard development steps required to add a new dimension or measure into TREDs data warehouse such as- dimension generation, alternate data sources and/or environments, or data visualizations,
  - c. Leverage standard tasks to create new test crash artifacts to be used for running acceptance tests and for adding them to the regression test suite.
7. Once patterns were established, during each Sprint we incrementally developed simple tools to reduce development time across the BI portion of the project and put it to use. We designed a simple star-schema dimensional table generator to use when faced with a large set of menial or simple script tasks. The team sought creative use of tools, such as excel, to create traditional star-schema database table structures and/or repetitive data generation statements for running complex MDX queries.

**Given:** Crash is completed its workflow and it is not a non-reportable crash. Crash contains multiple vehicles with different combinations of driver, passengers and pedestrians described in scenarios below.

**Scenario1:**

**When:** EMV In Service value of Yes is selected for a vehicle in crash

**Then:** All Persons in EMV vehicle will have EMV in Service = Yes and Pedestrian will always have a value of Not Applicable- Pedestrian

ID	DWPerson Type	DWCrashID	DWVehicleID	EMV In Service
1	Driver	1	1	Yes
2	Passenger	1	1	Yes
3	Passenger	1	1	Yes
4	Driver	1	2	No
5	Passenger	1	2	No
6	Pedestrian	1	-	Not Applicable – Pedestrian

**Scenario2:**

**When:** EMV In Service value of NA is selected for ALL crash vehicle in crash

**Then:** All Persons in EMV vehicle will have EMV in Service = NA and Pedestrian will always have a value of Not Applicable- Pedestrian

ID	DWPerson Type	DWCrashID	DWVehicleID	EMV In Service
1	Driver	1	1	Not Applicable
2	Passenger	1	1	Not Applicable
3	Passenger	1	1	Not Applicable
4	Driver	1	2	Not Applicable
5	Passenger	1	2	Not Applicable
6	Pedestrian	1	-	Not Applicable – Pedestrian

**Scenario3:**

**When:** EMV In Service value is not provided for ALL crash vehicle in crash

**Then:** All Persons in EMV vehicle will have EMV in Service is not provided and Pedestrian will always have a value of Not Applicable - Pedestrian

ID	DWPerson Type	DWCrashID	DWVehicleID	EMV In Service
1	Driver	1	1	Not Provided
2	Passenger	1	1	Not Provided
3	Passenger	1	1	Not Provided
4	Driver	1	2	Not Provided
5	Passenger	1	2	Not Provided
6	Pedestrian	1	-	Not Applicable – Pedestrian

Figure 4: An example of test acceptance criteria written in behavior specifications

8. For increasing the quality of features/stories delivered, we started to follow Test Driven Development (TDD) practices and implemented test automation for our BI features. This enabled us to test early and often. For this, re-usable and re-buildable test data sets using carefully engineered crash data was designed. This test data bed would satisfy all the possible scenarios for all dimensions and measures that existed in the Data Warehouse. A custom test automation suite was designed using SSIS tool. It was used to run complex MDX queries as assertions, based on the test bed data loaded, leveraging the predetermined outcomes of the dimensions and measures that system should calculate as.
9. Established Continuous Integration that helped us keeping all BI code and artifacts tested and working all the time, so that we could deliver at consistent intervals. We delivered most of the BI stories into production at the end of the sprint. If a large feature was developed, we delivered them on the 1-week release Sprint following a 2-week of development Sprint. The reason we still needed a short hardening Sprint was that there were large number of regression tests that had to be run across all layers (web, service, backend and bi layer) of the TREDs application, before we deployed the changes to production. We released on the 4<sup>th</sup> day and monitored production for any potential issues the fifth day of the hardening sprints. This mechanism ensured a high sprint capacity for the team.

CrashID	DocumentNumber	CrashTypeID	PersonID	PersonTypeID	DriverActionID	Driver Distraction - Navigation Device	Driver Distraction - Texting	Distraction - No Driver Distraction	Vehicle Body Type - Passenger car	Body Type - Truck - Pickup/Passenger	CrashInvolvedAlcoholID
1	99999901	2	1	1	2	2	2	2	2	2	1
1	99999901	2	2	1	NULL	2	2	2	2	2	1
1	99999901	2	3	1	NULL	2	2	2	2	2	1
1	99999901	2	1	2	NULL	2	2	2	2	2	1
1	99999901	2	2	2	NULL	2	2	2	2	2	1
1	99999901	2	3	2	NULL	2	2	2	2	2	1
1	99999901	2	1	3	NULL	2	2	2	2	2	1
2	99999902	1	4	1	1	2	2	2	2	2	2
2	99999902	1	5	1	NULL	2	2	2	2	2	2
2	99999902	1	4	2	NULL	2	2	2	2	2	2
2	99999902	1	5	2	NULL	2	2	2	2	2	2
3	99999903	1	6	1	NULL	2	2	2	1	2	2
3	99999903	1	7	1	NULL	2	2	2	1	2	2
3	99999903	1	8	1	2	2	2	2	1	2	2
4	99999904	2	9	1	NULL	2	2	2	2	1	1
4	99999904	2	10	1	NULL	2	2	2	2	1	1
4	99999904	2	11	1	3	2	2	2	2	1	1
4	99999904	2	12	3	NULL	2	2	2	2	1	1
5	99999905			1	4	2	2	2	2	2	1
5	99999905			2	NULL	2	2	2	2	2	1
5	99999905			3	NULL	2	2	2	2	2	1
6	99999906			1	5	2	2	2	2	2	1
6	99999906			2	NULL	2	2	2	2	2	1
6	99999906			3	NULL	2	2	2	2	2	1
7	99999907			1	6	2	2	2	2	2	1
7	99999907			2	NULL	2	2	2	2	2	1
7	99999907			3	NULL	2	2	2	2	2	1

Figure 5: A Snippet of the Test Data Bed used for loading Mock test data for running automated acceptance tests

All these measures translated to an increased velocity and quality. At our peak, the team was able to consume a BI feature that included designing and developing UI reports along with 83 dimensions, all within a span of one sprint. The team was able to complete and deliver into production.

#### 4. WHAT WE LEARNED

When we look back on our experiences, we realize we did certain things that are frowned upon among Agile Evangelists – breaking User Stories by effort type (Development, Testing, Release) in phase 1, or doing a release Sprint in every 4 Sprints in phase 2 of our maturity, or even we continued to have a need for a 1-week release Sprint followed by a 2-week development Sprint in phase 3 at our highest agile maturity. However we matured in our agile practices and were successful in following its principles, to deliver working software that met our customer’s requirements.

A few things that stand out that helped us deliver the BI solution to our customer’s satisfaction are:

1. The use of a BI Story templates to help our Product Owners to write better BI stories,
2. Breaking stories by identifying common patterns across stories,
3. Standardizing BI development tasks and templates,
4. Use of prototyping as a way to clarify BI requirements,
5. Building a custom testing as well as rollout tools that helped us do Continuous Integration and reduce effort taken to deliver new changes to production.

If we were given an opportunity to do it all over again, we would certainly put efforts from the get go, to:

1. Build test and release tools,
2. Identify commonalities among stories to build the higher level dimensions and measures
3. Eliminate the need for a release sprint by implementing build projects for our test suites using CI tools.

#### 5. ACKNOWLEDGEMENTS

We are grateful to have been a part of the TREDs project and learn from it. We would like to thank Ajay and Azmath for providing us with approved samples of work, for inclusion in this paper. We also sincerely thank David Grable for his guidance and being our scrum master along the way of writing this paper. This paper would not have come together without your shepherd’s keen insights, questions, and edits: *Thanks, David, we couldn’t have done it without you!*