



Order 23313 by Kara Joyce on June 2, 2020

# Disrupting Data Governance

---

A CALL TO ACTION

Laura B. Madsen, M.S.

Technics Publications  
BASKING RIDGE, NEW JERSEY

[Order 23313 by Kara Joyce on June 2, 2020](#)



2 Lindsley Road, Basking Ridge, NJ 07920 USA

<https://www.TechnicsPub.com>

Cover design by Lorena Molinari

Edited by Lauren McCafferty

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without written permission from the publisher, except for the inclusion of brief quotations in a review.

The author and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

All trade and product names are trademarks, registered trademarks, or service marks of their respective companies and are the property of their respective holders and should be treated as such.

First Edition

First Printing 2019

Copyright © 2019 Laura Madsen

ISBN, print ed. 9781634626538

ISBN, Kindle ed. 9781634626545

ISBN, PDF ed. 9781634626569

Library of Congress Control Number: 2019952193

Order 23313 by Kara Joyce on June 2, 2020

# Process

I was fortunate enough to learn the fundamentals of agile from the venerable David Hussman. David was an institution in the agile community in the Twin Cities. He also worked tirelessly all over the world helping organizations and individuals better understand what the agile framework could do for them. He could weave together the agile concepts with a unique storytelling style that left me both exhilarated and exhausted at the same time. His insights were so prolific, I often found myself wishing for a real-life pause button to process them to the extent they deserved. David left an indelible mark on me. It's difficult to condense what I learned from David in just a few words, but what resonated with me the most was the idea that there is *no inherent value in the process*. You don't get points for checking a box or jumping through flaming hoops. The work and what you build are what matter. Why do we spend so much time creating a process around the work?

Too many organizations spend too much time creating difficult to follow processes, often with corresponding software, to prove that they are working. These convoluted processes are nothing more than a method for management and stakeholders to

“check-in” on the people that are doing the work. Much of what is done, particularly in the IT space, takes time and feels abstract to people outside of IT. To create a sense of control and transparency, we identify all these steps and processes so we can collect the data and prove we are doing the work. This helps to fill the gap between the time we start and the time we can deliver, because sometimes the work won’t be delivered for months at a time. Unfortunately, in many organizations, there is a lack of trust between IT and the business. This often stems from a history of perceived slowness or not delivering what IT said they would deliver. IT spends a boatload of time in traditional waterfall methods proving that they are doing the work rather than doing the work.

My perspective on process has changed a lot over the years. Fifteen years ago, you would often hear me say, “Follow the process because the process makes you safe” but safe shouldn’t be your goal - delivering work should be your goal. Obviously, if your leader or organization requires these process steps to be completed so they can see you are working, there’s a breakdown in trust. As an individual contributor, you can’t fix that for the organization. But if you are leading a data team, perhaps now is the time to think about the transition to an agile framework.

This chapter will not teach you how to “Do Agile” or “Be Agile,” for that you will have to go elsewhere (see the reading list in the

appendix). What it will do is introduce some agile concepts you can consider in your data governance efforts.

The issues with data governance are layered. First, it's a definitional problem. We throw everything at data governance from definitions and usage to protection and security. We hang on to parochial methods of data governance with documentation, such as policies and procedures. We keep committees at the top, bury the people that do the work in a department and scatter decision-making rights between the people that know (i.e. stewards) and the people that have leadership responsibility (i.e. the executives). Most people realize that these methods don't often work; because it's a framework that we are familiar with, though, we stick to it. However, the way you're comfortable with isn't always the best way.

Another significant (and perhaps the most impactful) issue with data governance in a modern data platform is the volatility of the data itself. The data is coming into your systems at a high rate of speed and is always changing. So, if the purpose of governance is to create a lock and dam system to control the uncontrollable, how do you do that when volatility is all over the place? In other words, one day you're in the middle of a drought and the next day it's a 100-year-flood.

It is high time that we use modern methods to address our modern problem. Adopting agile methods, and leveraging DataOps and DevOps frameworks, we can begin to shore up the

challenges we face in our broken governance programs. DataOps (see above) capabilities are gaining steam for good reason, and I believe this framework is a good starting point for a modern approach to data governance.

DataOps is an amalgamation of agile, lean and DevOps specifically geared to support data and analytics efforts. In the “DataOps” manifesto it shares these values:

- Individuals and interactions over processes and tools
- Working analytics over comprehensive documentation
- Customer collaboration over contract negotiation
- Experimentation, iteration, and feedback over extensive upfront design
- Cross-functional ownership of operations over siloed responsibilities<sup>1</sup>

I’ll refer to this new method as Data Governance Operations (DGOps); it values usage of the data over the protection of an asset. It values the ability of teams to self form to address issues in the data, it sees all people across the organization as stewards of the data at different times for different purposes. It encourages questions of the data and it’s quality because that only increases the knowledge and value of the data itself. In chapter seven we will delve a little deeper into the DGOps concept.

---

<sup>1</sup> <https://bit.ly/2k0O3yc>.

---

## Stuff to stop doing

If we've learned anything over the last decade or two working in data governance it's that semantics, or what we call things, is important. Before I dive deeper into the "how" of using agile concepts for data governance, I want to update the language that is often used in governance.

It's time to get rid of the word "control." It implies something that's not achievable in a modern data platform. In addition, rather than creating a "standard" definition for our metrics, we should strive to create a "working" definition (WD), which allows for changes to occur, because changes will inevitably occur. Our goal should be to increase resiliency and the ability to respond to supposed anomalies occurring in the data. If we frame our governance efforts around creating standards, then we are starting off on the wrong foot.

As I was thinking through this work, I reached out to agile coach Kevin Burns. Kevin is a prolific lean and agile product development coach in the Twin Cities. He has a passion for helping teams turn their product ideas into implementations. As he and I discussed these concepts in the context of data governance, Kevin offered this: "Rather than a command and control frame, strive instead for adaptability and resiliency using visibility, pattern recognition, problem identification, decision



criteria, problem-solving methods and deployment (release) using agile methods.”

The best thing we can do is aim for improvement, and that’s what agile methods bring to bear.

---

### **Stuff to start doing**

I am still a proponent of defining your key organizational metrics. You should do that proactively. Just don’t get wrapped around the axle and don’t have too many metrics to start with. Create a list of key metrics the organization considers valuable and supports the decisions the organization needs to make. The executives should identify this list. The goal for the key metrics is to create a WD people can agree on. The intent is not to force everyone to use this one definition, but rather reach a level of agreement that encourages different parts of your organization to use these WDs when talking with each other. It addresses the part of governance that often gets touted, but rarely works well; making sure your executives are all looking at the same well-defined information.

There are some things in data governance that are still important and do work well. One of the key tenets to good DG is visibility and communication. Find like-minded people throughout your organization that either currently uses or want to use the metrics

that are on your shortlist. I call this your analytic community. Every organization has them; few use them wisely.

Pull together this team of cross-functional analysts to discuss how different groups use these pre-defined organizational metrics – but beware of *analysis paralysis*. It is really easy to analyze and re-analyze and find yourself down a rabbit hole or on another topic altogether. Analysts, in particular, want to make sure that they have considered and vetted every conceivable use of the metric and every nuance of the data. Recently, I was talking with a former CIO of a hospital who said it took them eighteen months to uniquely define and measure “weight of a patient”. I asked her, “What were people using while the governance team was working on the definition?” They continued to use the “old” definitions, or something that was probably *good enough*.

It’s very likely that this team of cross-functional analysts will get together and then it will spiral. That’s why it’s critical that you identify your definition of done, success criteria, and/or good-enough-for-now criteria before you start. These are long-standing concepts in agile methods and ones that will bring the most value to our new approach to DG. We need to recognize that we’re in a continuous state of learning, adapting, adjusting, and evolving. This helps us avoid analysis paralysis and helps get you to a minimally valuable product (MVP). In our new DGOps language, that’s our “good enough” WD. This WD has to be

based on our current understanding. If we agree that there is no inherent value in the process and that, in all likelihood while we're off analyzing data to get the definition perfect, the end-users are using good enough definitions anyway, the decision to adopt these agile principles should be obvious.

Another way to ensure that the team won't go off the rails, especially as you start this new way of thinking, is to consider a time limit or time-box for the discussion. If you follow the standard agile approach, you will have "sprints" or short cycles of work effort. Within a sprint, identify a limit of time in which you are willing to analyze the metric. You can use a few time-boxes first to do the data profiling and then test a couple of definitions, but then you should be prepared to propose a WD. Drive the discussion using the questions you need to be answered to make decisions to move forward (i.e. what business areas need to use the metric for decision-making). Then there should be some analysis related to questions you're unable to answer from the discussion (a little data mining to see if there are obvious things you've missed). This will likely involve data profiling work that will be done by the Quality Control (QC) team members. The importance of the analysis work is two-fold; first, to make sure the data supports how it will be used (sad to say that sometimes it doesn't!), and second, an understanding of how to implement changes once the team is ready to release the working definition.

Your QC team should create a data quality dashboard of (at a minimum) the key metrics so that at any given time, anyone can see how the data quality lines up. That level of visibility drives trust.

### **The steps**

1. Work with your executive sponsor(s) to identify a finite list of metrics the organization will use (ideally, less than fifteen)
2. A group of executives should rank order those metrics
3. Find your analytic community
4. Bring together a cross-functional team of analysts and data quality resources to discuss the metric of choice
5. Use business terms to define the metric
6. Use math terms to define the algorithm that supports the metric
7. Run standard data profiling methods on all the data that make up the metric
8. Review the data with the cross-functional team for errors, nuances, and insights
9. Edit or modify either the definition or the algorithm based on what you learned with the data (important step!)
10. Publish your working definition
11. Gather feedback, edit and repeat

These steps can be included in your product backlog, a list of all the work that you are undertaking to achieve a product. In this case the product is a set of pre-defined working definitions for the organizational metrics commonly used for decision-making.

Working definitions and the idea of “good enough” is actually difficult to operationalize. Recently, in a conversation with a client, they asked “what about the changes? If we focus on good enough and we realize that it wasn’t, how do we make the change?” It’s a good question, but even the question hints at a waterfall mindset. The question assumes there’s no value in the process. If we revisit the earlier example about the hospital defining weight in eighteen months and apply the DGOps method to it, the scenario might go something like this:

The weight of a patient in a pediatric hospital is a significant issue. It’s often difficult to track reliably and critical for patient safety. Most pediatric hospitals have many ways of defining weight, both categorically (i.e. grams versus pounds) and process (babies are weighed laying down, older children on a traditional scale). Comparisons are difficult and for all of these reasons, it’s important to govern and have consistency. Now that we’ve established weight as a key metric, within our DGOps method what’s next? We review all the data typically associated with weight and where it is sourced from and where it goes to, including all the steps in between. During this time, we communicate frequently to as many stakeholders as possible (i.e.

our self-creating teams). If we meet, it's to review the data - not theories about the data (i.e. everyone's definitions). During this phase we learn from one another about the metric (weight in this case) but are likely still using the "old" way of defining it until we can find a better way.

The difference between the DGOps way and the previous method is that we encourage people to continue to use the data. We review the data as it exists in our systems and we take the time to learn about the data and how people use it AS-IS. Everyone is a steward in DGOps because at one point or another, everyone is using the data for different reasons. This creates an opportunity for process changes where they are most impactful (for instance, if one unit develops a better way to take a measurement, other units could adopt it). It also alleviates the pressure on everyone to agree to one standard definition for all cases, because there is no value in that.

After all versions of the data has been vetted, and broken processes are fixed, the team has learned and shared a lot about the data, all while using it. Then and only then can we all agree on a definition that the organization can use to compare. That definition would pass all of our tests (litmus and coded) and doesn't violate how the data is generated (defining and using data as it exists). Just as in any agile or Ops method, testing and visibility are the keys to acceptance of the data.

---

## **You break it we fix it**

Because we will only be creating a WD for the limited set of metrics, you will find that there will be situations in which your users will discover data issues. It's important to know that this can be a good thing! That means they are using the data and it gives you an opportunity to dig into any data issue and determine what the problem really is.

For break/fix issues you will also want to take advantage of your backlog. Not all issues are created the same, nor are they of equal importance and it will not be feasible to track down everything. You will need to create a mechanism for your organization's response and ability to prioritize these things. This typically entails a quick review to determine the impact and severity of the issue.

For example, if one of your WDs is found to have an issue, it will likely carry more priority than an obscure metric issue that only impacts one person in one department. Try hard to avoid the fireman phenomenon (running from fire to fire putting them out as fast as you can) when you're structuring your break/fix capability. Rely heavily on your product owner to help triage the urgency and importance of requests.

---

## DataOps

A few years ago, on my agile journey, I turned to Google – like most people do these days. I was looking for information regarding the use of agile concepts or examples of using agile methods for data programs. I came across the DataOps Manifesto ([www.dataopsmanifesto.org](http://www.dataopsmanifesto.org)). I remember reading it and filing it away in the back of my head. Then as I started research for this book, I re-visited the Manifesto and had the opportunity to talk with one of the co-writers, Christopher Bergh, CEO and Head Chef of DataKitchen. I talked with a lot of people about this book, but the conversation with Chris felt like it lasted ten minutes, but I took the full hour. We were in what I like to call “violent agreement” about the value of DataOps for data programs.

In the manifesto there are eighteen principles; you should read them all. I cherry-picked a few because this isn’t a chapter on the manifesto, it’s a chapter on how to take the concepts and use it for a process update in governance.

The gist of the manifesto for our purposes is this: analytics is a team sport. Focus on getting the data or results of the analysis into the hands of users, something data governance should address as well. Visibility to the process (including testing and code) is integral to success, because it helps build trust. As I outlined in the first paragraph, most organizations actually place



value on the process itself. They turn to the steps in the process and check them off like they've accomplished something. They use that to pretend they're providing visibility into the work they do. They won't show you the work, but they will show you that they spent ten hours on step fifteen in the project plan. The transition to agile and DGOps requires a degree of visibility and transparency that makes the people who have been using the old methods for a long time very uncomfortable. Particularly in analytics, we have come to believe that the quality of the data is OUR job, and ours alone.

Many of us don't believe that other people in the organization have the capability to understand what we are doing or how we are doing it. But that's beside the point. Understanding isn't always the goal. If you can't provide transparency to the process, anything you say about a delay becomes a trust issue. As Chris said in our conversations "Hope and trust are important feelings to have, but they don't belong in analytics." Our users shouldn't "hope" for great work and we shouldn't force them to "trust" us without the willingness to back that up. There is no question that the transition to DGOps and agile methods will be disruptive to your organization and to your staff. You've already done it the other way and it hasn't worked. What's wrong with trying it a new way? Even if DGOps doesn't work for your organization, you will still probably learn a lot. Keep making improvements, because that's all any of us can ask for continuous improvement.

It might be worth your time to consider an even bolder step, one that requires you to blend together the changes in the people and roles, with the changes in the process. Most agile experts will tell you that change in the roles is as important as changes in the process. Self-creating teams and product owners are the things that catapult most agile efforts to the next level of velocity. I've seen this firsthand; through a humbling experience I had as I managed my first agile team.

One of the things I really like about agile, at least how we were doing it in this particular project was the exposure to the data. I had data points showing how much time someone thought something was going to take against how long it actually took. Every two weeks I would happily sit down with my Excel sheet and calculate out the accuracy of individuals and make (what I thought was) helpful adjustments to the work to ensure that we improved our accuracy of estimates. My thinking was the closer our accuracy, the better the projections of the model. The better the projections, the easier time I had sharing it with executives. After a while I noticed our completion or close rate start to diminish at a much higher rate than I could rationally explain. Our due date kept pushing out and I was getting nervous. I'd call a meeting together and we would fiddle with the model to see what we were missing. The team would sit there quietly and listen to me pontificate on the importance of velocity. A few brave souls would raise their hand and tell me why something

took longer, and I always had a (useless) management-like response.

After a while, I was in full panic mode, so I called in David (the big guns). By this point David had been diagnosed with cancer and was actively in treatment. I was only hoping for a bit of an email exchange, but he offered to come in. There's a lot I can say about that day: the energy in the room, the discussions we had, and the fact that David was tired and weak, but as insightful as ever. But what I learned that day changed everything for me. I learned that I was the problem. I was so busy being helpful I was messing people up. The problem with our velocity wasn't the team or their ability to accurately determine the work effort... it was me! I was in their hair all the time mucking up their flow. Talk about an exercise in humility. I had always prided myself on being the person that helped my team members become the best versions of themselves. On that day, though, I realized that I had been forcing them to be the best version of what *I thought* they *should* be, slowing down the project in the process.

I took a deep breath and I stepped back. I worked with good, smart people, and I needed to let them do good, smart work. Immediately, they began to self-organize, create the backlog, and work through their own issues. I was always there when they got stuck, and I still prioritized the backlog, but it was different. They felt it and I felt it. And you know what happened? Our velocity tripled.

---

## Prioritization and product owners

Now here's the thing about having lots of analysts doing lots of things for lots of different people in your organization: it produces a natural conflict for prioritization. Every organization has limitations to what they can do; whether that's with the number of people they can hire or with the amount of money they can invest in a data asset. Eventually, you will find yourself in a situation where analysts (or ambassadors) have competing projects. They might all have great endpoints, but you cannot get them done at the same time.

Typically, this is the point where I would recommend that you create a committee of people that can review all the requests and prioritize them based on the value they provide and/or the effort they take. I still make that recommendation to lots of organizations but if you are ready to take on a truly agile way of thinking about delivery incremental deliverables are key. The idea of holding onto things until a committee can get together and review them flies in the face of many of the key attributes of agile and DataOps Manifesto and our newly minted DGOps. What's a product owner to do?

First, make sure you really have a conflict. Sometimes things seem difficult, labor-intensive or contradictory on the surface, but turn out to be none of those things upon digging deeper. Instead, it's the people involved artificially introducing barriers,

for different reasons. Here's one example: you have a small request from the business that's being pushed out 3 -4 sprints from now. The business doesn't understand why it takes that long, and frankly neither do you. You realize the issue isn't the work itself, it's that there's only one analyst with the skill set to complete it – and they are fully utilized. You have a bottleneck issue, not a complex problem requiring a team to solution and debate. Do what you need to do to solve the problem in the short term and make a note that you have some cross-training to do in the future.

Agile is designed for the work to be able to shift between resources as needed, but sometimes in data, particularly in smaller organizations, you don't have the luxury of that level of redundancy. Even if you've been savvy in hiring and know very well that someone has those skills, there is a cost to context shifting (see box below) that you need to be aware of. The main thing you need to consider is this: you value progress over committee reviews and delays.

Prioritize the work that you think has value and can be done quickly, move work around to different resources when you can and help your stakeholders understand the challenges you are facing. Changes can be made both in deliverables and prioritizations when you all agree on a minimally viable product. That's harder to do when your organization or team is all geared toward traditional waterfall methods.

“When you focus on one particular type of task, challenge, or information set, then switch to something completely different, you’re shifting contexts. Sometimes, the transitions are huge and jarring. Other times, you don’t even notice them.”<sup>2</sup>

---

## Love Your errors

Probably the biggest change I’m proposing to traditional governance is the idea of radically democratizing access to data. You have to get data out there, let people use it, find things “wrong” with it, and even ask terrible questions. You must learn to love the errors inherent in the data and stop trying to prevent them after the fact. I know that sounds categorically insane for a data professional to claim, but it’s true. Governance has become too much about protection from people using the data and not enough about promotion. The only way that we can get the full capability from the data assets we are so worried about protecting is to let them go a little and be there to pick up the pieces when terrible stuff happens. And it will happen. Adjusting your mindset toward ambassadorship and first responders allows you to protect what you can and prepare for the rest/worst. Using the DGOps model helps take incremental steps to improve the data,

---

<sup>2</sup><https://bit.ly/2zuzgzV>.

but it also helps us frame the usage of data into a functioning data governance capability.

Love the errors you find in the data. Don't hide them, this only makes your job more difficult and lets the people or process responsible for the error off the hook. As so eloquently described in the DataOps Cookbook, analytics is a team sport and for too long we have let our executives and users get away with nominal engagement. We brushed their lack of participation off by saying it was too difficult or that it was "our job," but data is nothing without context. The only way to provide context is through business knowledge. Active participation in a self-created team to deliver incremental value helps everyone understand their role in stellar data governance.

The shift to an agile or DGOps mindset is not insignificant. In the appendix of this book, you will find a curated list of recommended reading, based on my many great conversations with others on this subject. Take the time to read some of those other books; talk to other people that have made the shift, and help your organization prepare for some of the changes, or at least acknowledge that there could be huge value in doing so. Then, hire a coach to help you through the transition. Don't worry about software or tools right away. Take the time to get the process stuff figured out; tools should come *after* you understand the value you need from them. Give yourself time as the change happens. Some of the biggest lessons I learned during

my agile transition had to do with how I led the team. Leading and supporting your staff through this transition looks a lot like parenting. You give them the foundation they need to make good decisions, and you're there to dust them off when they stumble and encourage them to keep going.

---

### **Wrapping it up**

There is no end to data governance. It will run long past your tenure. Building a resilient program requires resilient processes. Instead of proving you're doing the work you're just doing the work. The one thing I want you to remember from this chapter is to keep making incremental progress. Data governance, perhaps more than any other data function, is an unrelenting foe. The only way to tame it is to be vigilant, don't expect a big bang, just keep moving forward and focusing on building minimally valuable products, showing value and improving every day.