



build great
products™

A MODERN TAKE ON THE AGILE MANIFESTO

Introduction to Derwyn Harris



In 2001, when the Agile Manifesto was published, workplace technology looked very different from how it does today. Communication channels were more limited in variety, and sharing of information was nowhere near as easy as it is today with high-speed internet, mobile data and a multitude of cloud storage options. So what does today's agile look like? Do the values of the Manifesto still apply? When did agile become more about process and less about the mindset? How can we merge the intention of the Manifesto with the new way we work? How can we evolve agile concepts to tackle the challenges of today in a new way?

During my career as a developer, consultant, and now as a product manager, I have lived the experience of the challenges faced by Waterfall, Agile and hybrid teams alike, both from the technology and business sides. With great respect for the Manifesto, I have long considered the implications of rethinking its tenets in light of the massive shifts we've seen in technology and the way we now work. Read on for an opportunity to rethink yesterday's Manifesto in a new light and deconstruct which concepts were home runs and which still need to evolve.

The Manifesto in 2016

The Agile Manifesto has been great for software teams, but has presented real challenges for business analysts and other stakeholders involved in the broader context of application development. The problems addressed by agile in 2001 still exist today; in fact they are magnified by the changing landscape of product delivery. As a result:

- More information than ever is available, maybe too much
- Context, conversations and decisions go undocumented
- Communication gaps widen due to geographically dispersed teams
- Time to market has dramatically shortened
- Customer needs continue to go unmet

Why Rethink the Manifesto?

- 1 The World has Changed
- 2 Software is Everywhere
- 3 Complexity has Increased
- 4 Projects Still Fail



THE WORLD HAS CHANGED

We are radically connected as a result of modern technology. We have invented ways to connect that were unthinkable in 2001 – from online storage in the cloud to social networks – from Facebook to LinkedIn and Twitter. This true for how we work. It is also true for how we communicate with customers in general. The customer voice is more empowered than ever and customers vociferously share opinions about products all the time.

As a result of this radical connectedness, hierarchies of communication have flattened and reshaped the power relationship between rulers and masses, or between management and workers.

Modern Technology in 2001

MOBILE:

The Ericsson R380 had just been released, laying claim to the term “smartphone.” Camera phone technology was still three years away from North America.



SOCIAL:

Social platforms were in their infancy in 2001, evolving from message boards to instant messaging and Wikipedia. Even MySpace was not around yet.



MUSIC:

iTunes was first released in 2001 as was the first generation of the iPod, beginning the biggest disruption ever in music acquisition and storage.



COMPANIES:

Google had 400 employees, and a new idea of the creative technology company was beginning to emerge. Amazon turned its first profit, laying the foundation for major growth in the coming years.



The State of Technology in 2016

MOBILE:

Smartphones are now the norm, with more sold worldwide than feature phones for the first time in 2013. We're also adding tablets and wearables to the array of connected devices.



SOCIAL:

The hardware devices we've started using since 2001 help us connect to each other through a number of highly adopted global social networks. It's easier than ever to share information and opinions across the world instantly.



MUSIC:

The music industry has gone through huge changes. The iPod still exists but is no longer the force that it was. The music industry is switching to subscription models (such as Spotify) and is integrating with hardware ecosystems built around personal devices and home systems (such as Sonos).



COMPANIES:

Google is now a global leader and is shifting from just software towards more complex product systems, such as self-driving cars, Google Glass and smart home systems. Amazon has released Kindles, Fire streaming entertainment devices and more recently entered the mobile phone market.





SOFTWARE IS EVERYWHERE

Automotive

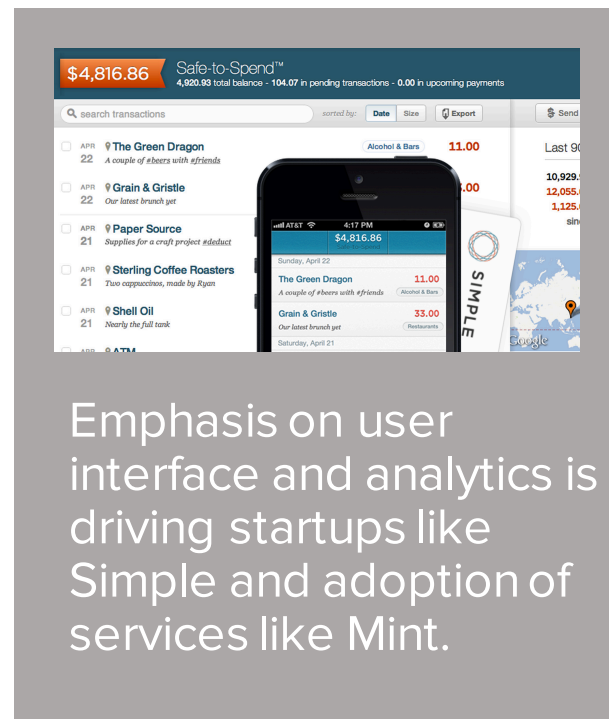
- In 2001, cars had a minimal amount of code in them. A current new car has about 100 million lines of code. ^[1]
- More than 150 million connected cars are expected on American roadways by 2020. ^[2]
- Remote Recalls: In March 2014, Tesla addressed a known fire risk in its cars in part by providing a software update to existing vehicles. This helped mitigate the risk without owners needing to visit dealerships or service centers. ^[3]
- Cars are also beginning to connect to each other. The U.S. Department of Transportation is working on a system that has the potential to reduce accidents as a result of an intelligent connected car network. ^[4]



Tesla is using information sent directly from its cars to better inform product development decisions. In addition, Tesla is sending frequent updates to improve the user experience.

Banking

- According to a report by Accenture, full-service banks in North America could lose 35% of market share to digital competitors by 2020. [5]
- Four of the ten most hated brands by millennials are banks. [6]
- Banking experiences are shifting away from brick-and-mortar, face-to-face interactions in favor of more digital banking experiences: mobile banking apps let customers budget, transfer money and deposit checks via camera phone.
- Peer-to-peer lending and payments make sending and receiving money easier than ever. Retail solutions including Square and apps such as Venmo and PayPal are connecting customers in constantly evolving ways.



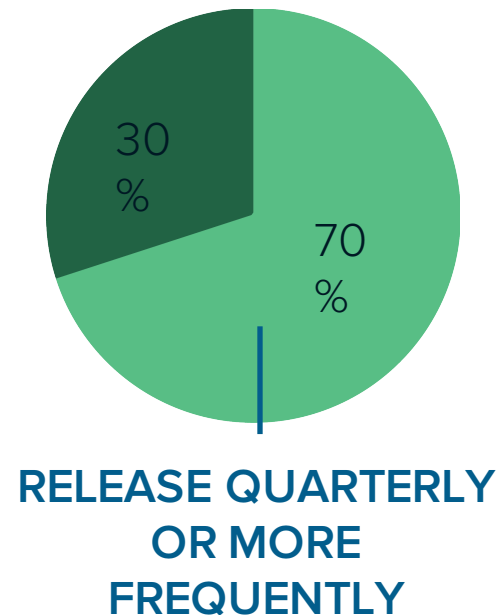


COMPLEXITY HAS INCREASED

Product

Building products used to be simpler; hardware products could only handle so many lines of code; web applications only had to worry about a couple of browsers and monitor sizes; and there were fewer programming frameworks.

Today hardware contains a lot more software. Software products need to consider many more devices and situations, and open source has provided the development community with many more libraries, frameworks and languages from which to choose. A greater number of platforms and browsers must be supported. This has introduced a huge amount of complexity. According to “The State of Modern Product Delivery,”^[7] a report by Forrester Consulting commissioned by Jama Software:

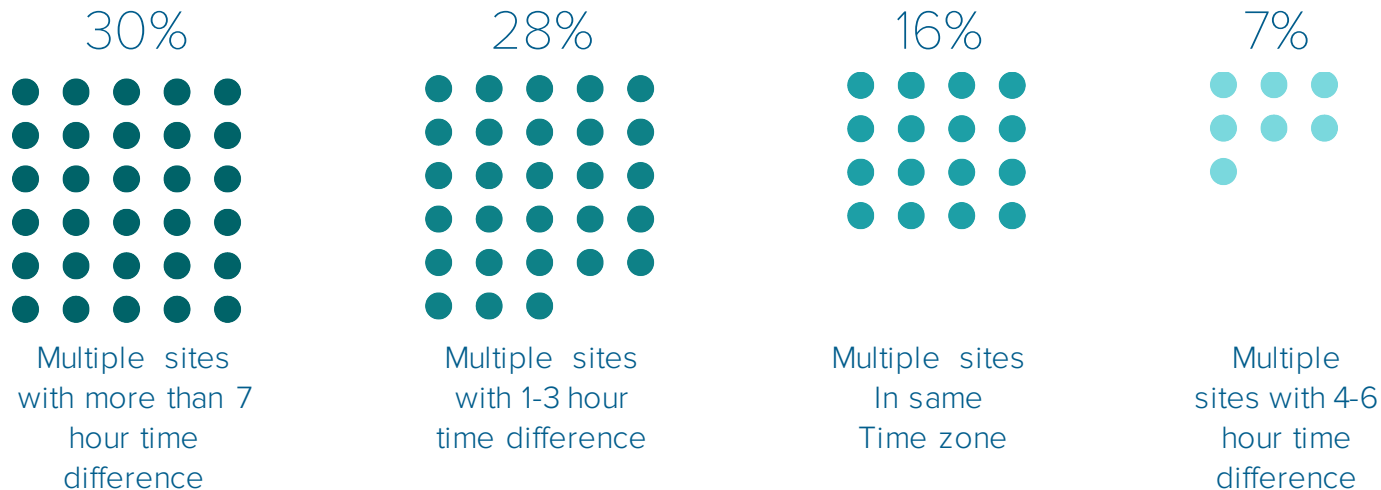


Organizational

We can now communicate across teams in many ways that were unavailable in 2001, enabling modern distributed teams to work together more efficiently. However, the growth in distributed teams also amplifies the organizational challenges of product delivery.

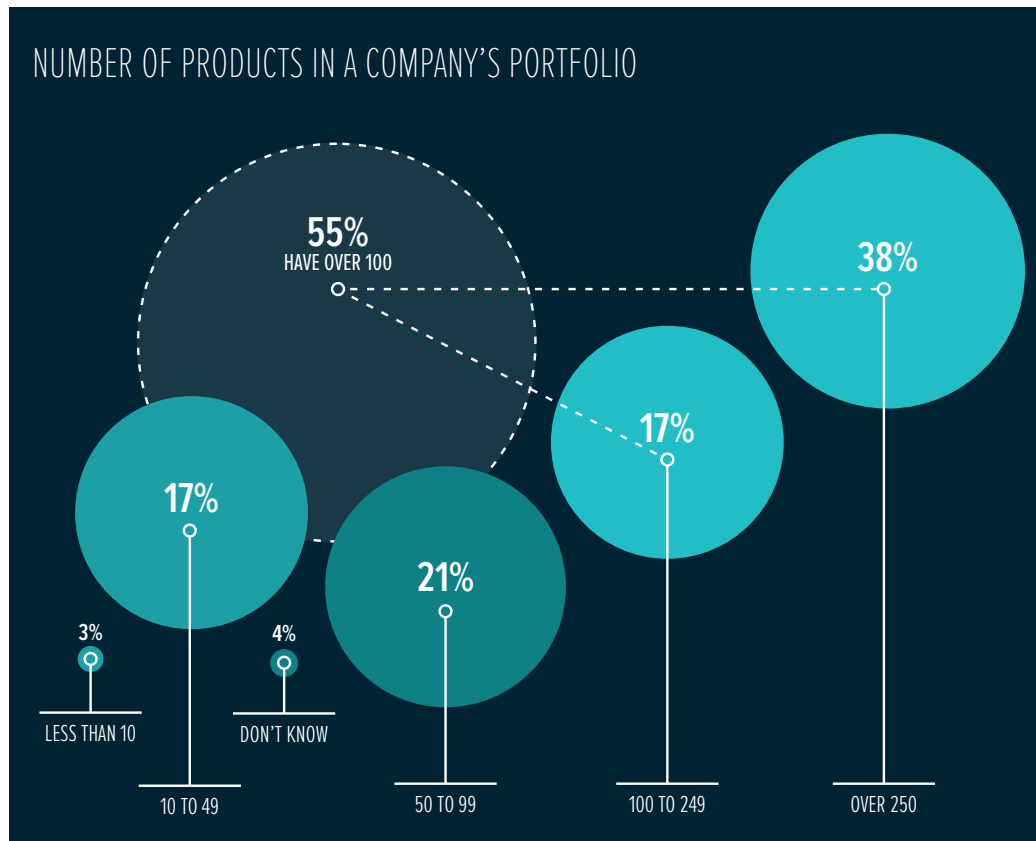
According to “The State of Modern Product Delivery,”^[7] a report by Forrester Consulting commissioned by Jama Software:

81% of teams are distributed across multiple buildings



Organizational

According to “The State of Modern Product Delivery,”^[7] a report by Forrester Consulting commissioned by Jama Software, companies have massive numbers of products under development:



4

PROJECTS STILL FAIL

Why Projects Fail

TEAMS ARE OUT OF SYNC:

Forty-one percent of business leaders identified “disagreement on requirements due to conflicting priorities or opinions” as the No. 1 problem they face bringing products to market^[7].

NO SILVER BULLETS:

What Fred Brooks wrote in his 1986 paper, “No Silver Bullet – Essence and Accidents of Software Engineering” remains true. The problem: we continue to seek out the silver bullets and fail to recognize the nuances of human behavior and complexity.

DECISIONS HAVE BIG IMPACTS:

There are hundreds, if not thousands, of decisions that take place during a product lifecycle. Robert Goatham, in his paper “The Story Behind the High Failure Rates in the IT Industry” outlines the number of individuals involved—compounded with the complexity of the project as primary reasons IT projects fail more frequently than others.

PEOPLE MATTER:

You could have the best process in the world, but if the people involved don't care or don't get along you are at risk. It's not to say that people are the only thing that matters but they're an important factor in product creativity, innovation and success.

The Definition of Failure has Shifted

DEFINING PRODUCT SUCCESS IS IMPORTANT AND CAN BE OVERLOOKED:

The Agile Manifesto promoted the idea of working software over big bang releases, but working software can have different meanings in organizations. It's important to establish what it means in yours. Evaluate the correct amount of context you need to provide your stakeholders to ensure you can maintain a working software process.

FAILURE CAN PROVIDE USEFUL DATA:

If done right, the data collected from failure can be valuable and actionable feedback. Product management is not only about defining what needs to be built but also about designing customer and stakeholder expectations. By defining outcomes and managing expectations with the rest of the organization, you'll be better able to utilize the data obtained from product failures.

IT'S WHAT YOU DO NEXT THAT MATTERS:

Success is theoretically about getting it right. Yes, agile and the idea of working software is about putting it out there for feedback, which means you are not always going to be right. What's key is to be prepared to react to the feedback so you can act quickly and continually. This builds loyalty and admiration in a world where customers are highly engaged and demand interaction.

2001 MANIFESTO

VS

2016 MANIFESTO

IS THE MANIFESTO AGILE OR IS AGILE THE MANIFESTO?

Given the vast changes that have taken place since the Manifesto was written, consider how the concepts or methodology of agile should adapt. Going through the Agile Manifesto line by line, what parts of it remain true today and which need to evolve to fit the times?

INDIVIDUALS AND INTERACTIONS **OVER** PROCESSES AND TOOLS

WORKING SOFTWARE **OVER** COMPREHENSIVE DOCUMENTATION

CUSTOMER COLLABORATION **OVER** CONTRACT NEGOTIATION

RESPONDING TO CHANGE **OVER** FOLLOWING A PLAN

“That is, **while there is value in the items on the right,**
we value the items on the left more.”

INDIVIDUALS AND INTERACTIONS **OVER** PROCESSES AND TOOLS

WORKING SOFTWARE **OVER** COMPREHENSIVE DOCUMENTATION

CUSTOMER COLLABORATION **OVER** CONTRACT NEGOTIATION

RESPONDING TO CHANGE **OVER** FOLLOWING A PLAN

“That is, **while there is value in the items on the right,**
we value the items on the left more.”



INDIVIDUALS AND INTERACTIONS

OVER

PROCESSES AND TOOLS

-
- Waterfall did not reward interactions
 - Individuals were considered resources
 - Interaction options were limited
 - Tools were cumbersome
 - Process was heavy and manual
 - Tools didn't drive process



INDIVIDUALS AND INTERACTIONS

OVER

PROCESSES AND TOOLS

In 2001, we worked more independently and our workplace interactions were limited by the technology of the times. Waterfall at its core was designed to limit interactions and focus on set, defined goals.

Agile's goal was to encourage teamwork and collaboration. Because options with technology and communication were limited, working more closely together was encouraged by teams sitting physically close together. This did not scale well across the organization. Roles such as business analysts and project managers were often left out of this process.

Process and tools meant something very different in 2001. Tools were few and process was very heavily manual with lots of steps and detail. Process focused more on "CYA" and quality than it did on getting the product right. The tools didn't really drive process in the way they do today. Rational was driving Rational Unified Process, but tools were immature in comparison to those that exist to support the process today.



INDIVIDUALS AND INTERACTIONS **COMBINED WITH** PROCESSES AND TOOLS

- Teams are more complex
- Options to interact have increased
- Tools have matured
- Many more processes exist
- Agile is a process



INDIVIDUALS AND INTERACTIONS **COMBINED WITH** PROCESSES AND TOOLS

In 2016 teams are more complex, as are the channels we use to communicate. Our understanding and acceptance of interacting through a medium has dramatically changed and continues to change.

One idea to consider is the notion that the Manifesto, in attempting to describe one OVER another (albeit with a disclaimer at the bottom) may have caused some of the confusion. In this instance, “combined with” is used in order to indicate a partnership.

This is a perfect example of the Manifesto favoring process over tools. Complexity of products and teams has increased. Tools have matured

greatly and should be valued more than they were in 2001. In fact, today’s tools are tightly coupled with individuals and interactions.

Some argue that the reason agile teams fail is that they did not follow the process strictly enough. This logic goes against the Manifesto. Process is an important consideration. It’s impossible to not have some kind of process. Tools are important and today it’s the combination of tools and process that is most important.

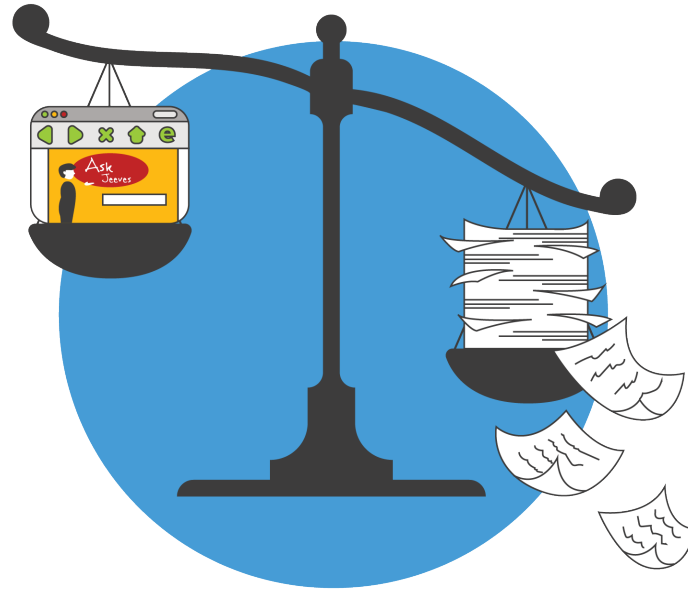
INDIVIDUALS AND INTERACTIONS **OVER** PROCESSES AND TOOLS

WORKING SOFTWARE **OVER** COMPREHENSIVE DOCUMENTATION

CUSTOMER COLLABORATION **OVER** CONTRACT NEGOTIATION

RESPONDING TO CHANGE **OVER** FOLLOWING A PLAN

“That is, **while there is value in the items on the right,**
we value the items on the left more.”



WORKING SOFTWARE

OVER

COMPREHENSIVE DOCUMENTATION

-
- Software was simpler
 - Software was typically “just” software
 - Big reveals were scary
 - People reacted to cumbersome documents
 - Documents were created in silos
 - The value of documentation was up for debate



WORKING SOFTWARE

OVER

COMPREHENSIVE DOCUMENTATION

In 2001 the notion was that documentation should be replaced by working software. Of course, back then “software” was a simpler concept. Certainly some software was highly complex, but overall, software products have grown greatly in complexity. The mindset at the time of the Manifesto was to document everything upfront, then go build. Often, the result was that teams built the wrong thing. In that context, eliminating comprehensive documentation seemed like a good idea.

Alistar Cockburn, signer of the Manifesto, has spoken about the word “comprehensive” and the decision to use it. According to Cockburn, this term was highly debated. The creators didn’t want people to think that documentation in and of itself was unnecessary because they did believe it was important. The intent was to call out *exhaustive* documentation as overkill.



WORKING SOFTWARE

**BALANCED
WITH**

COMPREHENSIVE DOCUMENTATION

-
- Software is no longer just software
 - What does “working” even mean?
 - How do we account for continuous delivery?
 - Documentation no longer means “Word docs”
 - We are “documenting” all the time
 - Documenting includes decisions



WORKING SOFTWARE **BALANCED WITH** COMPREHENSIVE DOCUMENTATION

Today we have more complex software, we also have realities of “MVP” and continuous delivery. The idea of working software is much more of a reality. This does not change the importance of documentation. What does change is the idea of the word “document.” A white board, sticky notes, wiki, or collaboration software, these are all documentation.

This is a critical and necessary aspect of the process. The ability to respond to change, to interact, in fact everything the Agile Manifesto believes in relates to communication and collaboration around something. The ideas, stories, epics, and decisions written and made everyday.

INDIVIDUALS AND INTERACTIONS **OVER** PROCESSES AND TOOLS

WORKING SOFTWARE **OVER** COMPREHENSIVE DOCUMENTATION

CUSTOMER COLLABORATION **OVER** CONTRACT NEGOTIATION

RESPONDING TO CHANGE **OVER** FOLLOWING A PLAN

“That is, **while there is value in the items on the right,**
we value the items on the left more.”



CUSTOMER COLLABORATION

OVER

CONTRACT NEGOTIATION

- Required customers to be onsite during development
- Proxy customers failed to accurately convey customer needs
- Collaboration meant something different

- Contracts feel final and distant
- Mismatched expectations between business and engineering teams



CUSTOMER COLLABORATION

OVER

CONTRACT NEGOTIATION

At the time of the writing of the Manifesto in 2001, the expectation in the software world was that customers and developers defined budgets and timelines of what was expected to be delivered and when. The developer was then obligated to deliver to specifications and on time. This arrangement left little room for change: the challenges or opportunities that inevitably arise during the development process. Contracts may have kept some projects on track but they also could stifle features or improvements that served the customer well. The Manifesto's solution was to have the customer on-site in lieu of making contracts. This was a noble idea, but not practical for all organizations or industries.

The reality is that in some companies and industries, contracts are still

necessary. Sometimes contracts are part of the vendor selection process or related to regulatory requirements. Another problem is that having the customer on-site often results in the unintended creation of “proxy” customers, individuals who are supposed to convey the concerns and interest of the customer. Instead, proxy customers created a point of failure. Relying on one individual to understand and comprehend what customers need and want can result in communication breakdowns. In many organizations, the role of business analyst helps to bridge the business and the technical sides, to translate the customer's problem into a set of requirements. They can help define what the business problem actually is, which might not be in the same words customers would use to describe their needs.



CUSTOMER COLLABORATION

**COMBINED
WITH**

CONTRACT NEGOTIATION

- Collaboration has changed
- Speed and innovation has changed our interpretation of the customer
- Collaboration vs. feedback loops

- Contracts have not gone away
- It's about communication and alignment
- Negotiation is synonymous to collaboration



CUSTOMER COLLABORATION

**COMBINED
WITH**

CONTRACT NEGOTIATION

Today when we say “collaboration” it means something different than it used to. Modern collaboration tools make it easy to convene, discuss and agree to decisions with individuals around the globe. Collaborative processes are now a reality for dispersed teams and make it possible to be in constant communication with our customers.

While collaboration has not eliminated the need for contracts across industries, it has changed our ability to combine collaboration with the contract. The contract keeps teams in alignment with the original vision and core business value of what is being built. If the vision changes, continued communication with the customer means the contract is a living agreement that can adapt to change.

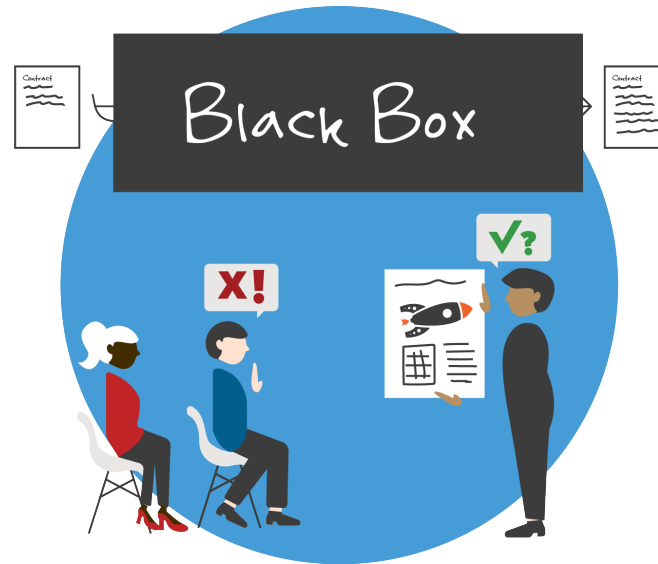
INDIVIDUALS AND INTERACTIONS **OVER** PROCESSES AND TOOLS

WORKING SOFTWARE **OVER** COMPREHENSIVE DOCUMENTATION

CUSTOMER COLLABORATION **OVER** CONTRACT NEGOTIATION

RESPONDING TO CHANGE **OVER** FOLLOWING A PLAN

“That is, **while there is value in the items on the right,**
we value the items on the left more.”

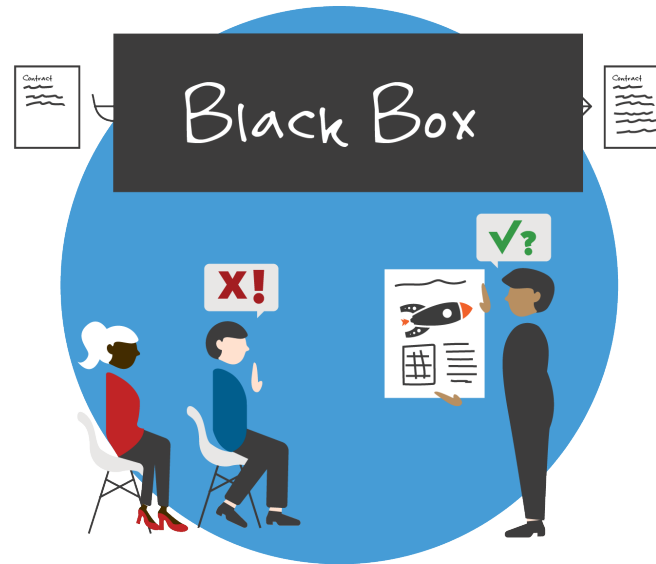


RESPONDING TO CHANGE

OVER

FOLLOWING A PLAN

-
- Change is up to interpretation
 - Requirements help define the intent
 - “Plan” was synonymous with “rigid”
 - Outcome will be bad if the plan is bad



RESPONDING TO CHANGE

OVER

FOLLOWING A PLAN

In 2001, the perception—and often the reality—was that a plan represented the final work. It was treated like a gate into the Black Box of engineering. You had to have a plan or you couldn't solve the customer problem. The engineering team then went into the Black Box with the plan and built the product based on their interpretation. Of course when the product failed, many blamed it on a faulty plan. Others blamed engineering for misinterpreting the plan.

A lot of effort went into defining requirements upfront, so they were crisp and clear to the developer team. Entire books have been written about how to write requirements in an effort to reduce confusion around “the plan.”

The Agile Manifesto tried to fix this by eliminating documentation, i.e. moving away from “the plan.” Teams also began to interpret the idea of responding to change as meaning “just do what the customer wants.”



RESPONDING TO CHANGE

**COMBINED
WITH**

FOLLOWING A PLAN

-
- Quickly understand and resolve
 - Requires connections and collaboration
 - You say plan, I say vision
 - Follow together vs. follow blindly
 - Communication and alignment



RESPONDING TO CHANGE **COMBINED WITH** FOLLOWING A PLAN

How you respond to change is key to determining whether your outcome is success or failure. Understanding how to get to the core of the change can be an indicator for success.

Change is inevitable. Regardless, there must be a plan. It's important to refer back to the plan often to avoid drifting away from the core business value your product needs to deliver. Any drift needs to be within acceptable bounds for the project and must be necessary. If it isn't, then the plan needs to evolve.

Ultimately change and vision go hand in hand. We must work together to constantly evaluate and understand what we are building and why. One can even argue that it's not change we are responding to. We are responding to the many decisions made throughout the process. People can't make good decisions in a vacuum. They must have a clear understanding of the business value they are trying to deliver and a crisp grasp of customer needs.

SO WHAT'S THE FUTURE?



MODERN PRODUCT DELIVERY

Future Methodologies

- Data and Science will Drive Decisions
- Culture will be a Focus
- Individual Behavior will be a Driver
- Tools will play a Larger Role

RECOMMENDATIONS

Pause and Rethink

- Decouple the Manifesto from agile
- Continue to shift our understanding of failure
- Evaluate your current communication methods
- Track metrics and use them to iterate
- Think of agile from an organizational perspective

Communicate

- Find ways to constantly provide visibility to the organization
- Allow communication and opinions to flow freely
- Identify where decisions are made and captured
- Constantly ask if people understand not only what they are doing but *why*

Evolve

- Open up the dialogue about culture
- Embrace process
- Communicate, Communicate, Communicate
- Start with too much information, then work your way back to find the balance

The world looks very different today than it did in 2001 when the Agile Manifesto was written. Many of its core tenets are still relevant today, but there are also opportunities to incorporate modern tools and technology to further evolve our development processes.

By rethinking the Manifesto for today's work environment you can better communicate with stakeholders to provide more visibility and continually evolve your process.

Share your thoughts on rethinking the Agile Manifesto by finding me on Twitter [@Derwyn](#) and connecting with other product professionals in the [Product Delivery Group](#) on LinkedIn.



build great
products™

ABOUT JAMA SOFTWARE

From concept to launch, the Jama product delivery platform helps companies bring complex products to market. By involving every person invested in the organization's success, the Jama platform provides a structured collaboration environment, empowering everyone with instant and comprehensive insight into what they are building and why. Visionary organizations worldwide, including SpaceX, The Department of Defense, VW, Time Warner, GE, United Healthcare and Amazon.com use Jama to accelerate their R&D returns, out-innovate their competition and deliver business value. Jama is one of the fastest-growing enterprise software companies in the United States, having exceeded 100% growth in each of the past five years, during which time both Inc. and Forbes have repeatedly recognized the company as a model of responsible growth and innovation. For more information please visit www.jamasoftware.com.

Sources

[1] <http://www.wired.com/2012/12/automotive-os-war/all/>

[2] <http://press.ihs.com/press-release/country-industry-forecasting/big-data-drivers-seat-connected-car-technological-advance>

[3] <http://www.nytimes.com/2014/03/29/business/safety-agency-ends-investigation-of-tesla-fires.html>

[4] http://www.washingtonpost.com/local/trafficandcommuting/direct-communication-between-car-computers-may-reduce-accidents-by-up-to-80-percent/2014/02/03/b55e9330-8d1a-11e3-833c-33098f9e5267_story.html

[5] <http://www.accenture.com/us-en/outlook/Pages/outlook-journal-2014-digital-disruptors-how-banking-got-agile.aspx>

[6] <http://www.fastcompany.com/3027197/fast-feed/sorry-banks-millennials-hate-you>

[7] <http://go.jamasoftware.com/rs/078-EIF-407/images/jama-state-of-modern-product-delivery-forrester-thought-leadership-paper-EDU.pdf>