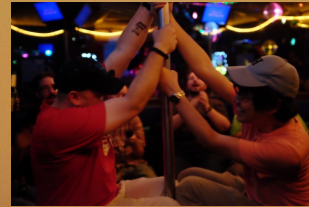


The Pony Express

And How Technology Moves Fast

The Pony Express & How
Technology Moves Fast

About me



PJ Hagerty | @aspleenic | Mattermost



The Pony Express & How
Technology Moves Fast

Where I Work



Mattermost

mattermost.com

The Pony Express & How
Technology Moves Fast

Some History



PJ Hagerty | @aspleenic | Mattermost

Many people who watch Westerns and modern takes on the romantic American Western movement believe in certain things. The good guys always wore white hats and always won the day, everything west of the Mississippi was lawless, some bank robbers weren't such bad people, and the Pony Express was how the mail got delivered.

To many people, these are "facts". Dearly held beliefs that have been allowed to become the stories we build other stories upon.

But if we look closely, there are some holes in this story...some really big ones.

Some History



Before even the Civil War, some folks decided to head out to seek their fortunes. The United States was trying to achieve a goal called Manifest Destiny - expanding from East Coast to West Coast. For some people, this meant opportunity - be it to find farm land, to establish new freedoms, or to find the gold lying in the hills across America.

When prospectors found gold on the California Coast, towns began to pop up all along the coast. It was necessary for people, both companies who owned land or paid prospectors and families of these adventurous souls, to get mail as quickly as possible. These remote locations were growing. Between 1848 and 1860, the population of California grew to 380,000 people. With very few ways to get information from one end of the country to the other, a new solution had to be found.

Some History



While railroads could get messages a good portion of the way across the continent, they couldn't make it much further than Missouri. When the United States gained New Mexico and Arizona as states in 1853, we could only just start to consider bringing the coasts together via rail.

And let's be honest. The railroad back then was no guarantee of speed or safety. It was kind of like having a server in a closet being maintained by someone with a general idea of how to run Linux or Ubuntu and making their own cat-5 cables...which were entwined in unsafe ways through out this closet.

Unsafe, unstable, prone to robbery - the railroad couldn't be the right answer for this growing country.

Some History



Add to all these aggravating edge cases a small inconvenience in the idea of who was doing all the work. While some people wanted to bring the country together, others wanted to remain on top of the pile by oppressing other human beings.

War was on the horizon. If we wanted to continue to be successful and to move forward as a country, we needed a way to communicate quickly. On the eve of the American Civil War, the Pony Express was born.

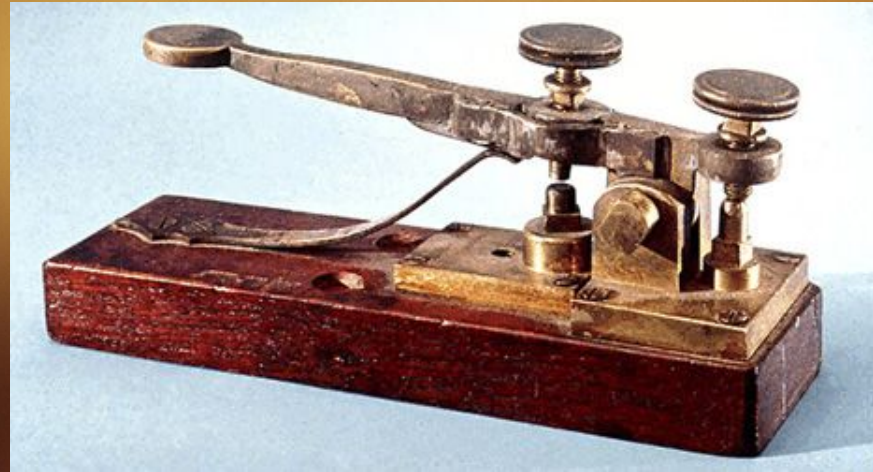


A network was built. Riders, generally young, often orphans or escaped slaves, would be hired to ride like the wind. The pony express began its ride April 3rd, 1860. Suddenly, you could get a message from New York City to San Francisco in just 10 days!!! It was a miracle in modern communications!!!

If we've learned anything from the movies about the Pony Express it's about how ubiquitous and long lasting it was. Seems every Western has some reference to the Pony Express. There have been movies and television shows dedicated to these brave riders and their horses - flying across the plains, outwitting dangerous animals, bandits, and Native Americans who were kind of upset we kept taking their lands and territories (rightfully so).

The Pony Express - the paragon of innovation in April 1860.

Some History

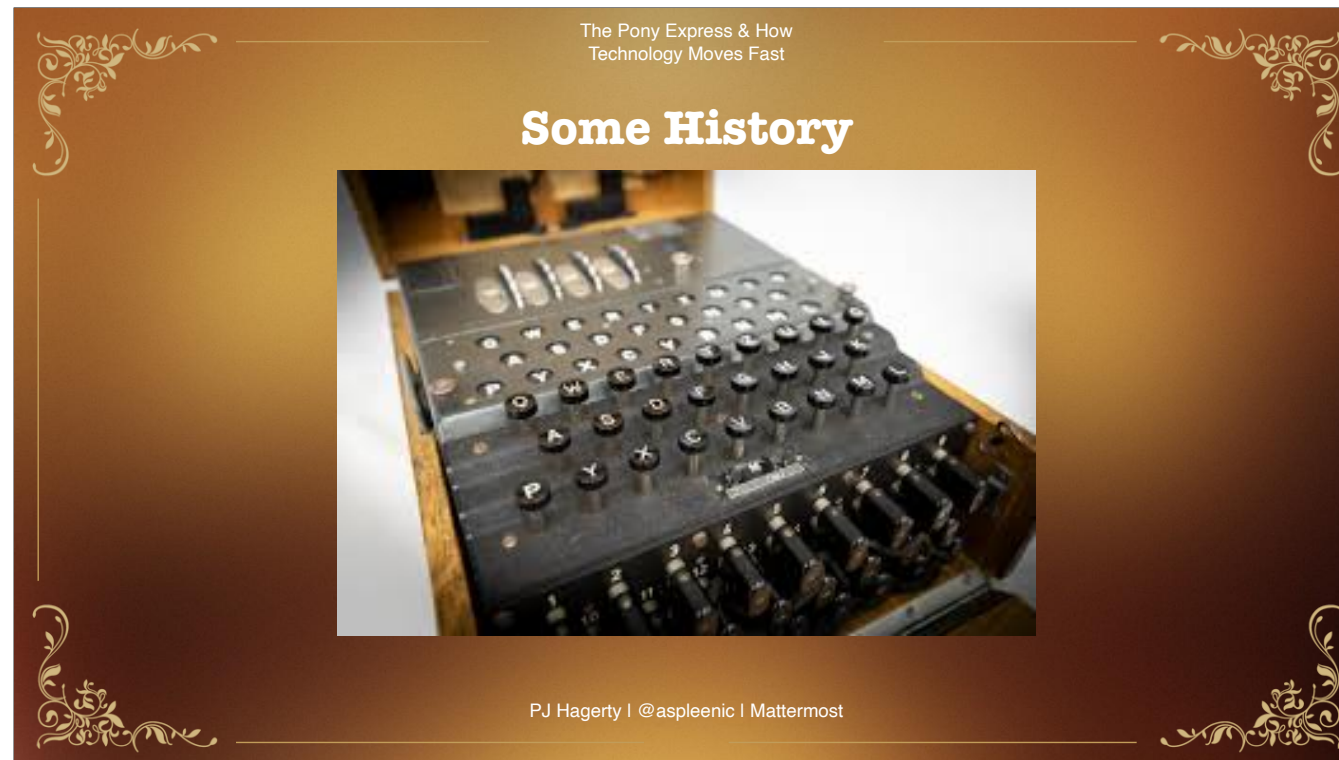


PJ Hagerty | @aspleenic | Mattermost

However, this great technique for moving messages coast to coast was outdated after a mere year and a half. The needs of the people and the speed of technology overtook these Western riders. So instead of innovation, the Pony Express was really just a stop gap method, and would've remained a footnote in American history, were it not for the so called penny dreadfuls that raised the Pony Express rider to heroic status.

A simple device at the right time, a time when war was starting and communication needed to move faster than ever before. All this culminated in the end of the Pony Express after a mere 18 months of operation.

As a side note, think of the telegraph and the Pony Express the next time you have a conversation about machines replacing people. Automation is not a new issue. When is the last time you tipped your elevator operator?



If the Pony Express was a stop gap, how long did the telegraph last?

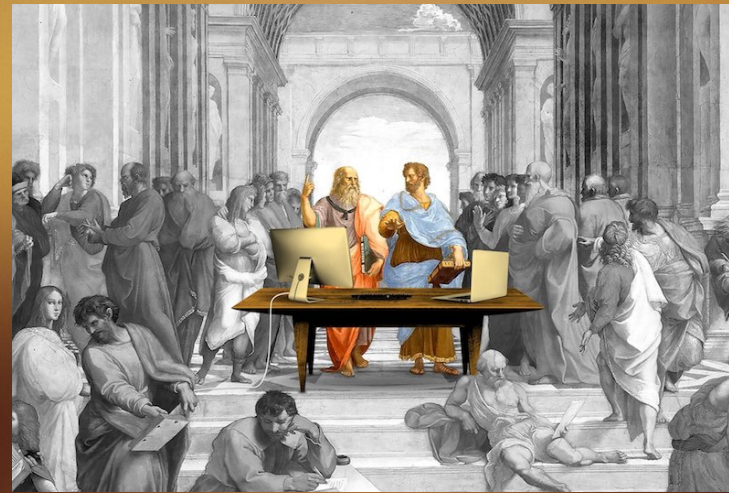
Funny you should ask. Perhaps you've heard of a British person, sometimes called the father of modern computer science. In World War II, about 80 years after the Civil War, Alan Turing was in England trying to decipher telegraph messages that were encoded using the above machine. Known as Enigma, it was a way to encrypt and decrypt messages sent via telegraph.

Some pieces of technology last a long time. Others, fade quickly or have limited use and disappear. We see this all the time in the application world. Unless all of you plan to share your thoughts on this talk on your Beebo page or LiveJournal.

So let's take a look at some things and see their lasting power. Maybe we can speculate on what has staying power and what is more a flash in the pan (by the way, that is literally an 1849 gold prospecting reference).

The Pony Express & How
Technology Moves Fast

Tech Philosophy



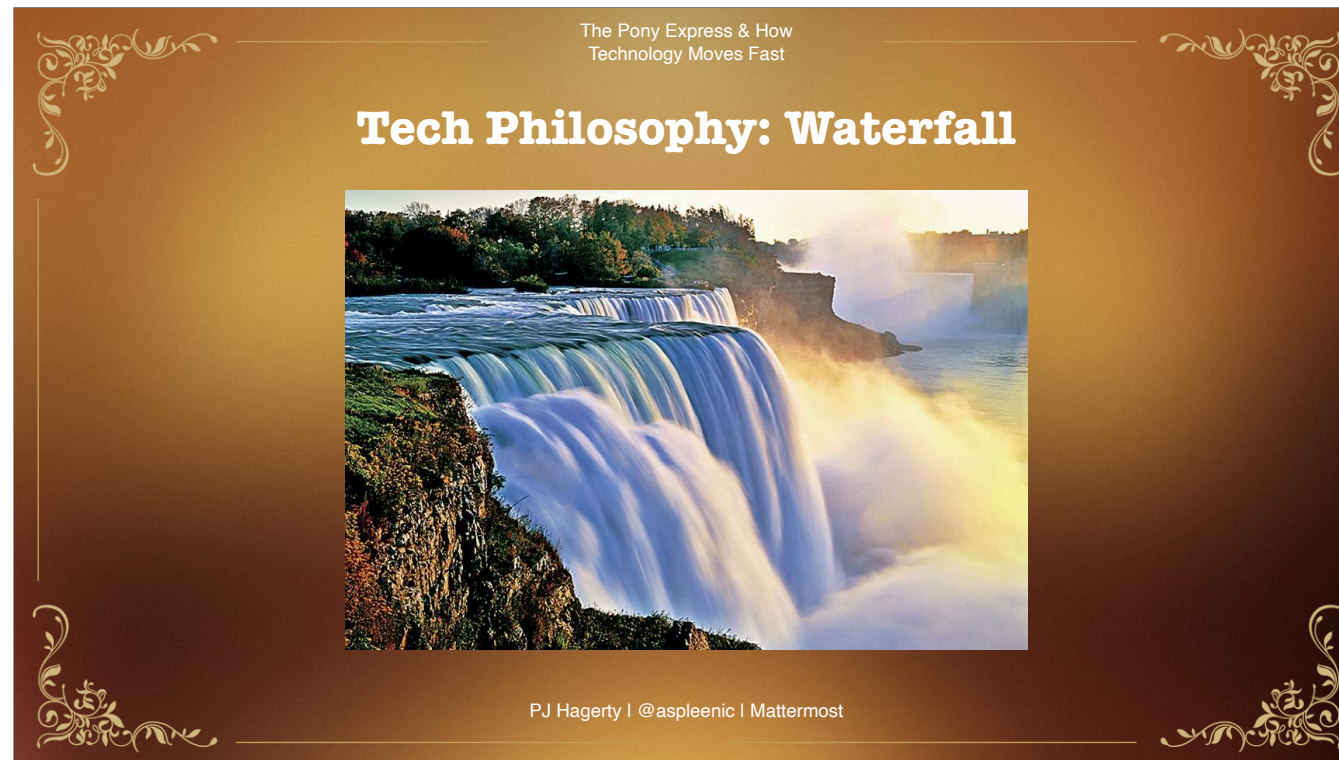
PJ Hagerty | @aspleenic | Mattermost

We'll start with a Philosophy lesson.

Philosophies around technology are not a new phenomenon. Whether it's the idea that mainframe computing is the only true computer science solution to the idea that DevOps can be certified and assigned as a team - there have never been a shortage of ideologies that accompany the work we do.

Most philosophies are really meant as best practices to allow our work to flow easily and produce better applications or run better networks or data centers, on the earth and in the cloud. Some are only suggestions because something "worked on my machine" so it must work for everyone else.

Let's take a look at the most common modern era philosophies and see what the strengths are and where the pitfalls might be.

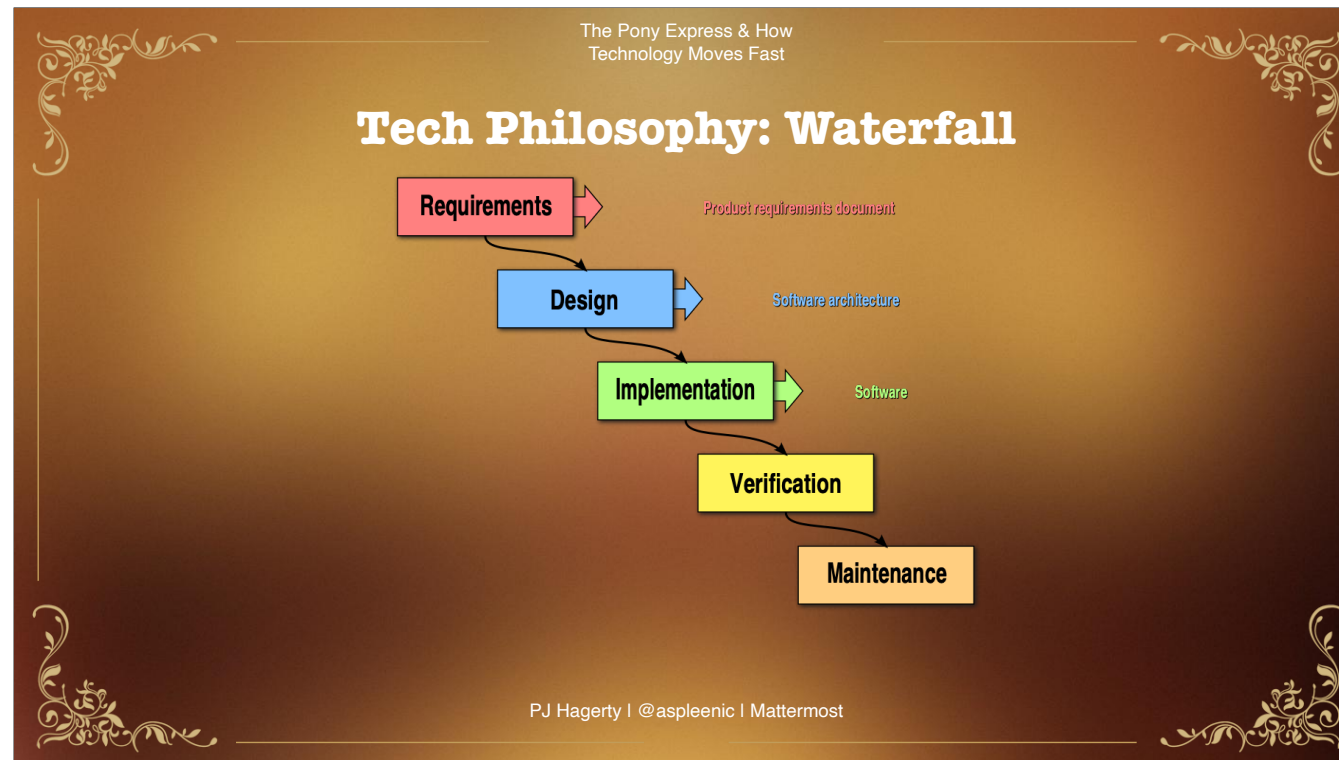


Waterfall is probably one of the most well known, and much maligned, development methods in the world today. How many of you have been involved in an organization using the Waterfall technique?

For those not in the know, here is the definition of Waterfall:

The waterfall model is a relatively linear sequential design approach for certain areas of engineering design. In software development, it tends to be among the less iterative and flexible approaches, as progress flows in largely one direction ("downwards" like a waterfall) through the phases of conception, initiation, analysis, design, construction, testing, deployment and maintenance.

Pretty thorough for an idea first developed in the 1950's. A mere 10 years or so after Turing was dealing with the telegraph we were trying to find engineering techniques in computer science. Innovative for its time, though even shortly after people saw the flaws.



Waterfall is simple. As our programming teams began to build applications for business and scientific enquiry, we needed to establish a way to do so in an organized way.

Is it in any way surprising our first thought was to take a page from modern manufacturing techniques at the time. To build a car, you establish requirements, design it using those requirements, build the car, test to see it works, and maybe do maintenance when it's out in the world.

Easy, right? Why wouldn't software work in the same rigid way?

The Pony Express & How
Technology Moves Fast

Tech Philosophy: Waterfall



PJ Hagerty | @aspleenic | Mattermost

The problem with this philosophy, like so many first ideas, like the Pony Express, is its short sightedness. What works in one industry does not necessarily work for another.

In a few minutes we'll discuss the Agile methodology, much of which focuses on the concept of LEAN, an idea brought over from Japan's car manufacturers. There is a difference in how these ideas were adopted. For Agile we borrowed. With waterfall we just took it all.

Waterfall has no iteration, no feedback loop, no structure for movement outside of the specs. Most people know when an organization is not flexible it spells doom. Waterfall was a doomed philosophy before it got off the ground.

Yet, parts of it are still used to this day in large scale corporations.

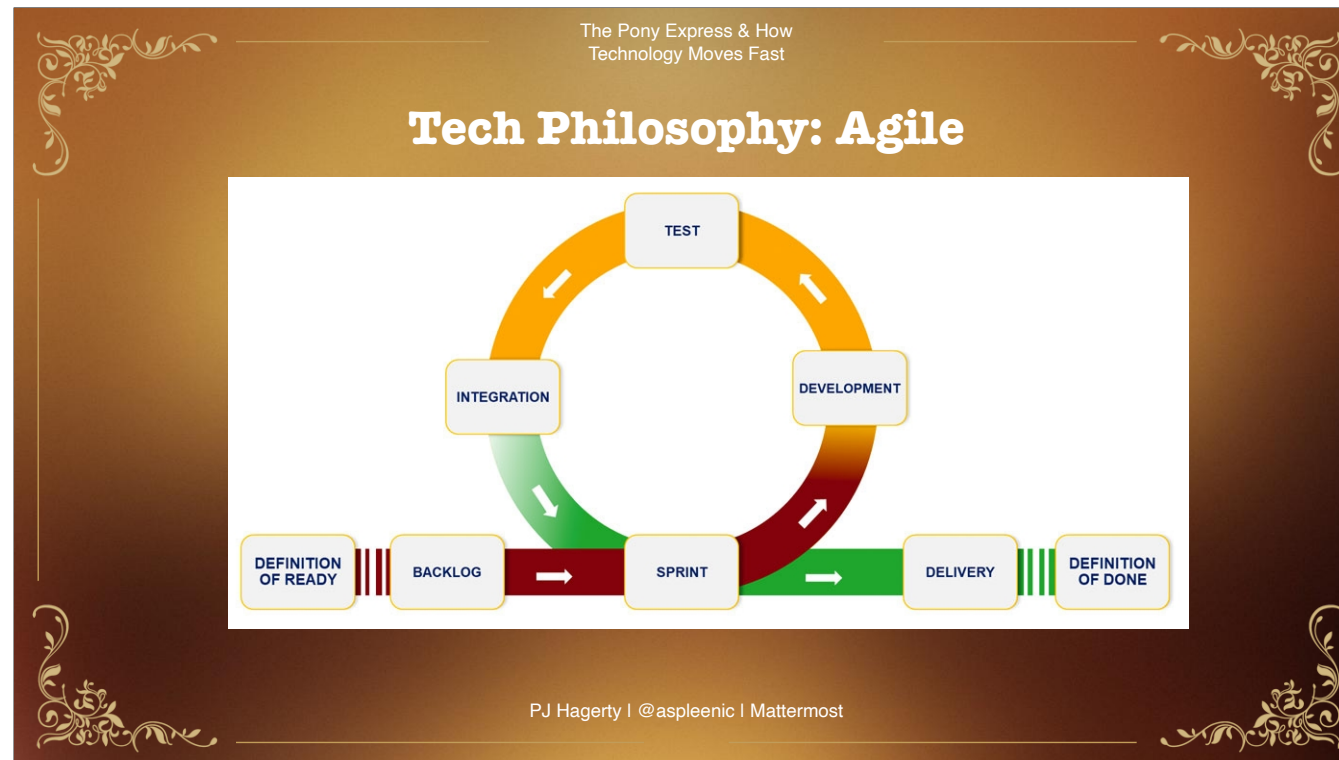


Agile is more modern, more widely accepted philosophy in the world of tech. While the word conjures up images that may or may not work along with the ideals of the philosophy, let's define it before we make any decisions.

Agile is an approach to software development under which requirements and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and their customers or end users. It advocates adaptive planning, evolutionary development, early delivery, and continual improvement, and it encourages rapid and flexible response to change.

This sounds amazing after waterfall. Self-organizing teams? Iterative development? Customer input?

What could go wrong?



Agile is the space age solution. Everything we are looking for is there: iteration, collaboration, flexibility

At the advent of Agile, which was, believe it or not the 1970's, there still wasn't much need for adherence to a "software development" philosophy. Most software was still built in and for large scale corporations. Computers weren't common in homes and even entertainments systems like Colecovision and Atari were only coming into being.

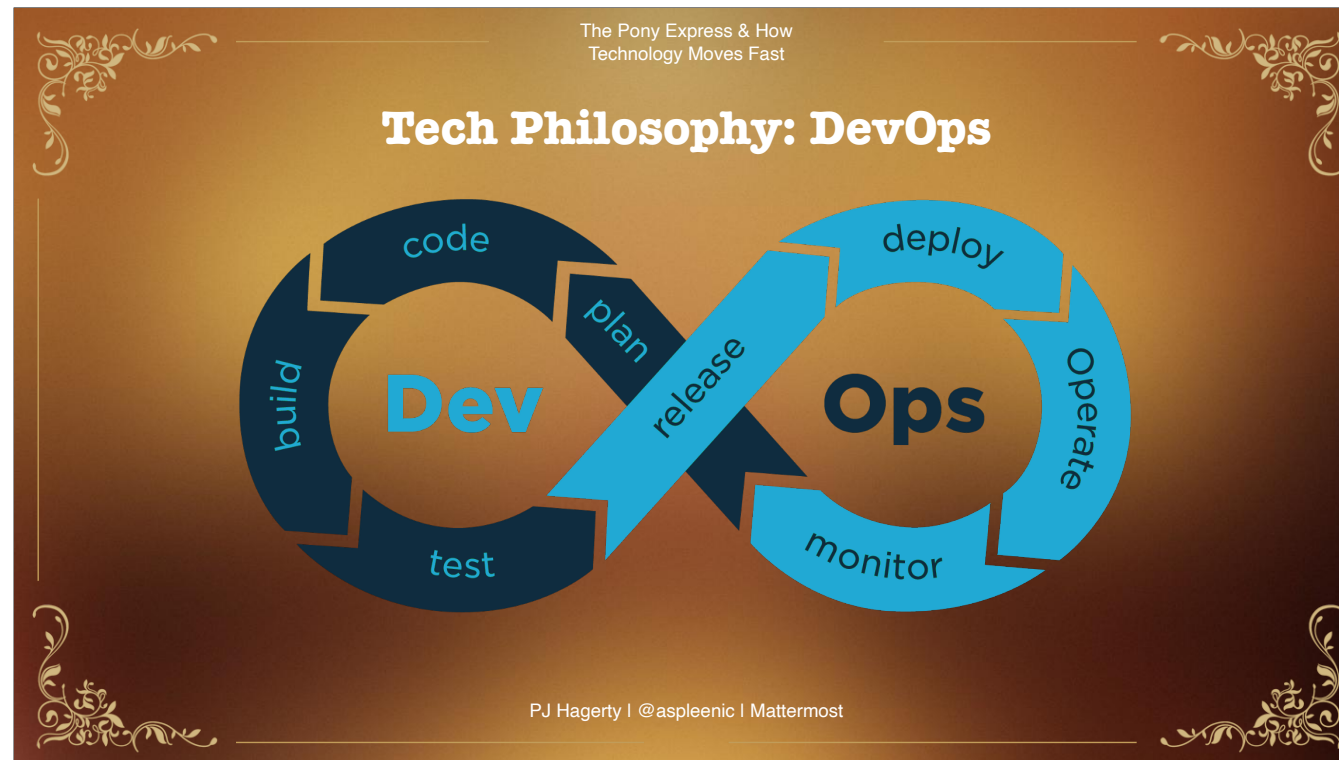
Agile was a sleepy philosophy for a while. Then, as more modern application development and deployment grew in the 90's, it began to take off. We couldn't just apply normal manufacturing techniques to building and deploying software, we needed real world ways that applied to the problem at hand. Something modern and different.



Agile is an amazing modern work flow. With a few caveats, that really come from the peripherals. Things like Scrum and Kanban, the add-ons that aren't really necessary and don't really contribute to the success of the team.

Agile means flexible. If your team can work without the overhead of complicated necessities like daily stand-ups and RAFT teams, agile will probably work for you. As with any philosophy, zealotry and strict adherence stagnates growth. If you need to follow the rules as strictly as possible, that's not very flexible...not very agile.

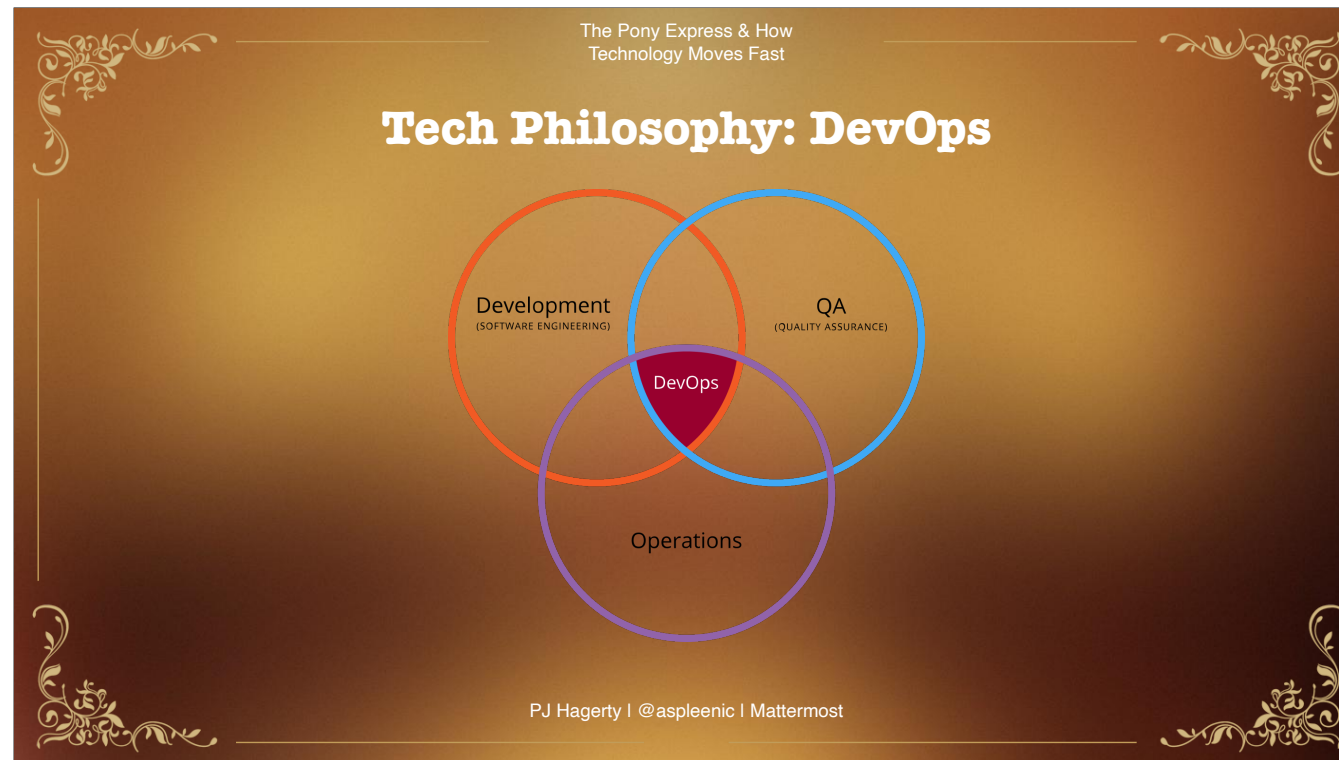
As a great bonus, Agile lead us to something new. The reason we're all here. DevOps!!



DevOps is a set of software development practices that combines software development (Dev) and information technology operations (Ops) to shorten the systems development life cycle while delivering features, fixes, and updates frequently in close alignment with business objectives. Different disciplines collaborate, making quality everyone's job.

Personally, I love this definition. It's really what the other philosophies are striving for as far as how a tech team should function. All the best parts of Agile, the few good ideas from Waterfall, applied to all teams across an organization. Everyone is part of the conversation, everyone is responsible.

This is our modern answer to how best deliver software - mobile, web, OS's - doesn't matter what. We've learned from the past and this is what we have.



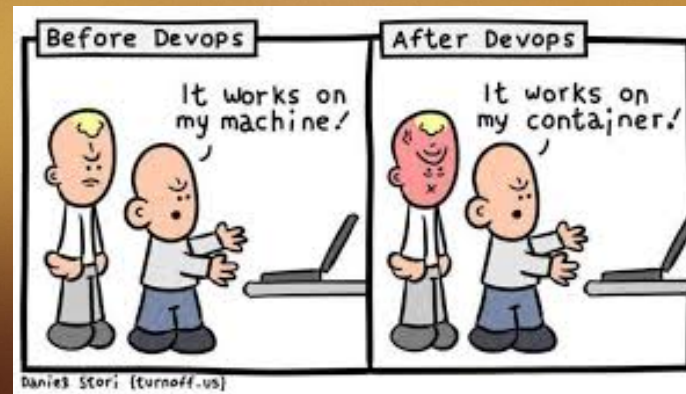
DevOps, according to some, should be the end all be all of development. We introduce the concepts of interaction with developers and ops or IT. We add in the folks from QA and testing - a great cycle. And what do we focus on?

AUTOMATION!!

DevOps goals (according to some) is to automate as much as possible to remove risk. I mean, isn't that really what happened to the Pony Express? We needed to move messages faster. So instead of doing it manually on horse back, we automated it. We built a system that made the old system less useful.

Makes one wonder if part of the point to DevOps is to make DevOps unnecessary.

Tech Philosophy: DevOps



PJ Hagerty | @aspleenic | Mattermost

Like all theories, DevOps is great in theory. Everyone works happily together now because we are employing DevOps. Right? All problems are resolved, and we all get along and the world is at peace and the applications work perfectly.

DevOps - it works 60% of the time every time.

Like waterfall and Agile before it, DevOps has one major downfall. It is based on the “perfect world” idea. In a perfect world all these things work. When faced with the reality of organizations functioning at modern speed, some of these things fall down.

This doesn't mean we should abandon hope all ye who DevOps...but we need to see where these philosophies work and where they don't. We need to take our organization's perspective into account to achieve our goals. We need to be critical of the things we do and find the parts that don't work.

All things in tech should be a constant evolution - not an “end goal”. Who ever writes software that is done?



Beyond our methodologies for how the infrastructure of our team works, we need to build things. Whether in ops or dev, these ideas come through.

We'll start with Test Driven Development. This is the concept that everything you build, software or hardware, application or infrastructure or database, must be thoroughly tested before going out into the world. This is a great idea...except that it takes nearly twice as long to build anything. Additionally, since you never REALLY know how people will use a thing once it's in the world, it can be needlessly pedantic.

So maybe Behavior Driven Development is better?



Unlike Test Driven Development, Behavior Driven Development is based on how users interact with, and break, the things we do. Forward thinking right? Let's see how users do things and build what we need based on that information. Brilliant. Less time spent than TDD right?

Maybe...but it also leaves a huge possibility of falling over again and again. Luckily, from the BDD movement, DevOps culture has been able to adapt and adopt the concepts of Chaos Engineering. This is great and behavior based, so even out of misguided philosophies we can cull some good.



What really goes on the world, in every organization, every team, is Shame Driven Development. People only write as many tests as they can get away with so the next developer doesn't think they skipped it all together. The ops person only takes their time setting up physical hardware so it seems there time was worth what they want it to be worth. The SRE holds everything together with duct tape and bubble gum, but documents a well running system.

Whether DevOps or Agile or what have you, we generally build things so we won't be shamed by the next person to see it. This is the truth of where most things live in the world of tech today.

Kind of like how the Pony Express was built "good enough" for what the United States needed at the time. We innovate, but often the focus is on Minimal Viable Product, Minimal Viable Setup...Minimal...Viable.

Tech Philosophy: Language Zealotry



Just the language we develop in comes with philosophies and ideals that matter to some, but seem to be less and less important as time marches on.

Compiled languages vs non-compiled. Open Source vs “Enterprise” languages like C++ or .Net. We’re seeing now a return to functional programming practices while also seeing the growth of JavaScript tools focused on building single page web applications.

These choices lead to things like not needing a proper server - deploying a Github page with Jekyll or MiddleMan or GatsbyJS - you don’t even need an Ops team, right?

But these aren’t innovations so much as iterations on the same circle of concepts. We aren’t building new things, just solving the same problems in different ways.



These are all concepts. Ideas. But most are based on the ideas of others. What is it like to truly innovate? How do you change an industry that has been moving in a certain direction with a certain momentum for so long?

When DevRelate came about, it was the process of 2 years of consideration on the back of 7 years of experience. We were the first Developer and Community Relations as a Service company. Now, there are a handful of companies that have worked on the same model. I won't say they are the same, but they are competing in the same space.



The interesting thing about most of the innovation we hear and see is that it's about technology. The Pony Express was about technology, kind of - using the resources available to complete a task.

Also, it's clear the Pony Express was a bubble. One that burst quite spectacularly.

We hear this term in tech and investment fairly regularly. The Bubble - will it burst, are we safe?

It's hard to tell. We know last time, VC backed firms fared poorly while bootstrapped companies survived (for the most part).

Will the new bubble burst? It's hard to tell, though it seems we may have learned from last time and hopefully, we know how to keep our heads above water if things go south.



The issue, however isn't only resiliency through a bubble burst, or playing it safe so we can be sure to rake in the cash. The issue is innovation.

We need to continue to push forward - I hate to use this word, but it works - we need to disrupt industries as often as possible to ensure we innovate.

We can't stop, we won't stop.

A lot of companies say they are building the things they build in order to see "a better world", but few can define what that means. Maybe it's something for them, personal, but not expressed. For me, I'd like to see a world with equitable pay for everyone, where people who aren't white males can innovate and build awesome things without worrying about being looked at as different or less than. I want to see a tech landscape that is vibrant with ideas. I'll use my platform to do as much as I can to make it happen.

That's just one form on innovation. You need to ask what your motivation to innovate is.

The Pony Express & How
Technology Moves Fast

Innovation - Moving Forward

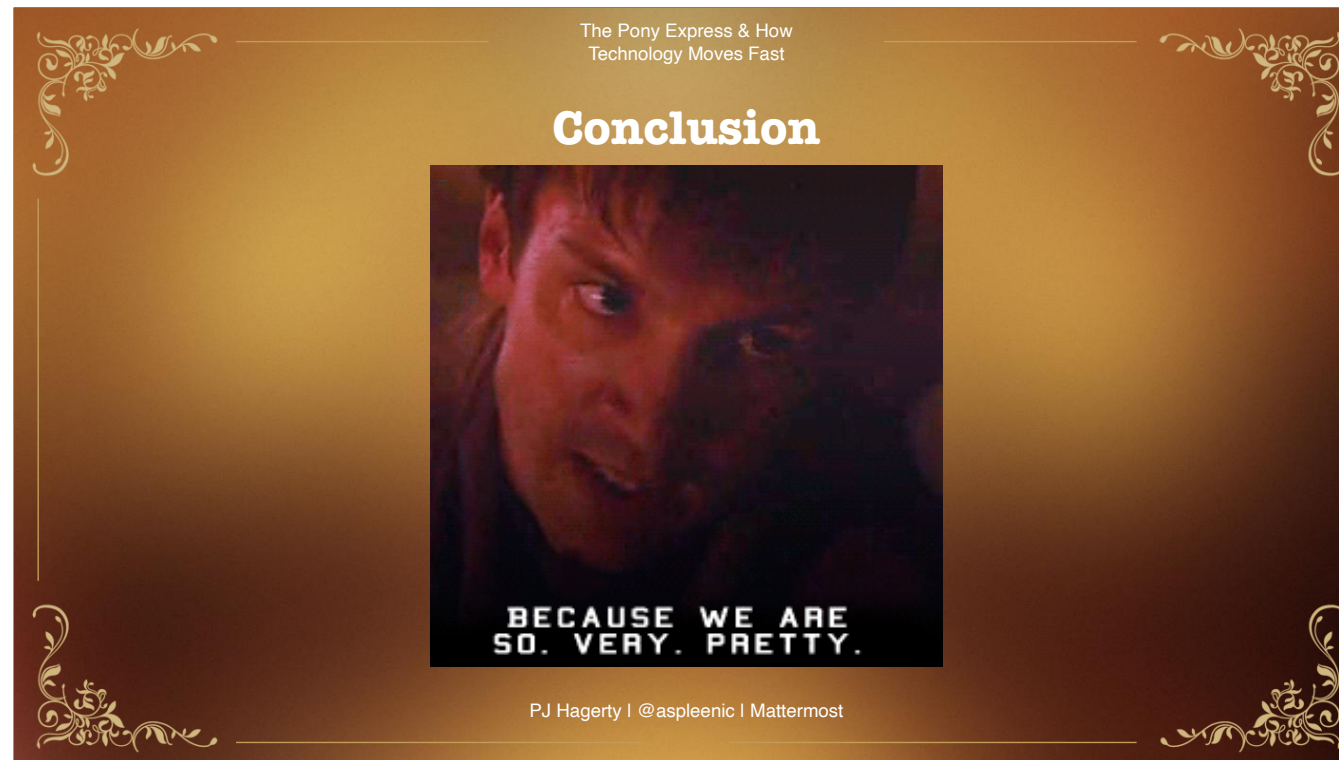


PJ Hagerty | @aspleenic | Mattermost

We strive to keep moving forward. We might be the current Pony Express. Will we last longer? Will we become the Telegraph of this generation?

When we think of the word innovation, we think of building something new, discovering some territory. We've done that by bringing teams together toward the common goal. DevOps has taken us a step further by getting rid of those accoutrements to Agile, all the rules, and saying hey, we all have the same goal. Let's work together to reach it.

That's innovation. That's riding to outlast the competition. Technology moves fast. We need to learn, adapt, and innovate if we hope to keep up.



So we see things come and go. We see innovation built on failure, philosophies grow from simple, adopted ideas, concepts borrowed from somewhere else then retrofitted to our needs. We see that what was the best solution today might be outdated tomorrow.

When adopting DevOps culture and ideals, we seek to find the best solution to our teams and our development and ops environments. We should never accept these ideas blindly. We must question every step of the way.

At some point, there was a young man on a horse racing across the hills of California to deliver a message. He never read it, that wasn't his job. His job was to deliver. That message read: "TELEGRAPH SERVICE REACHES SAN FRANCISCO: PONY EXPRESS RETIRED"

Every point of our job in tech is to innovate and build. We continue to do so knowing that the future is our goal, not the nearby goal.

The Pony Express & How
Technology Moves Fast

Thank you kindly, folks



Mattermost

mattermost.com

PJ Hagerty | @aspleenic