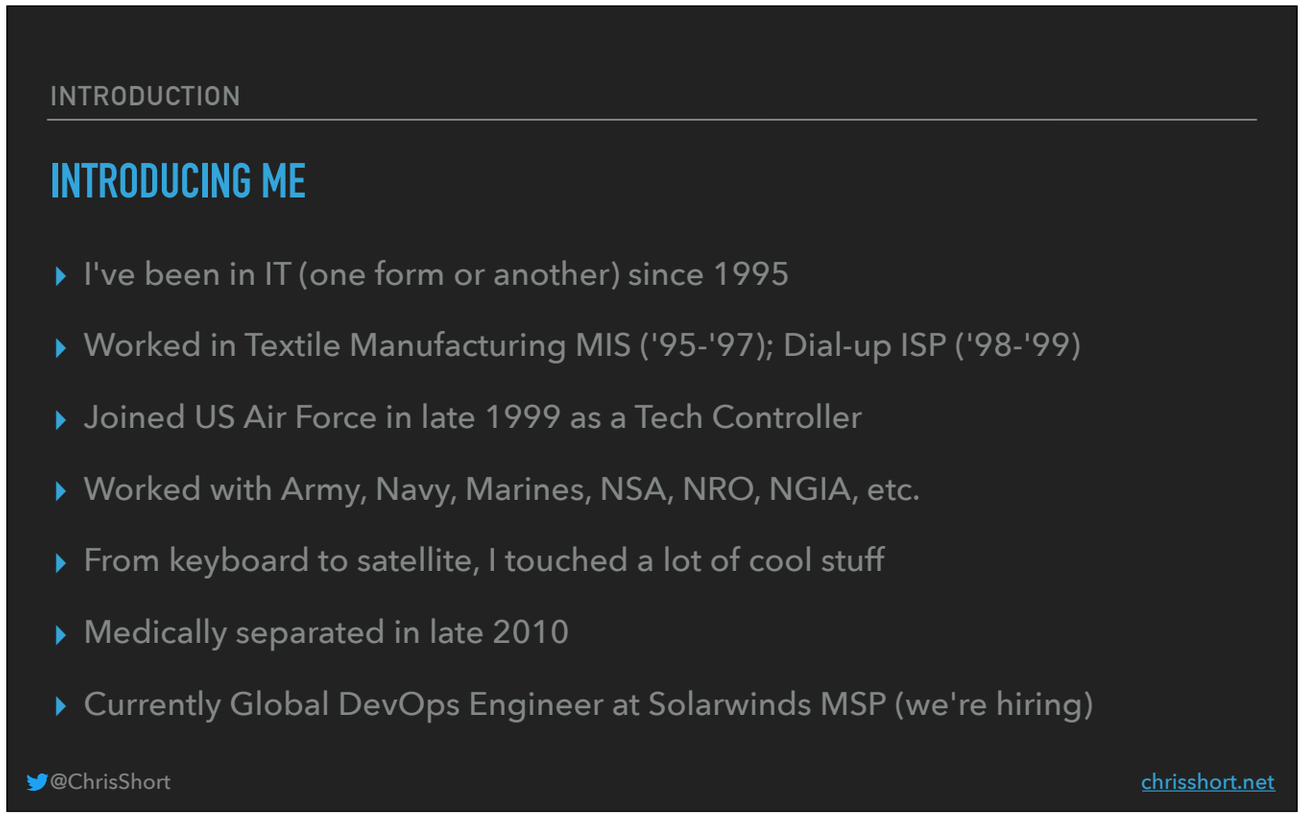
CHRIS SHORT

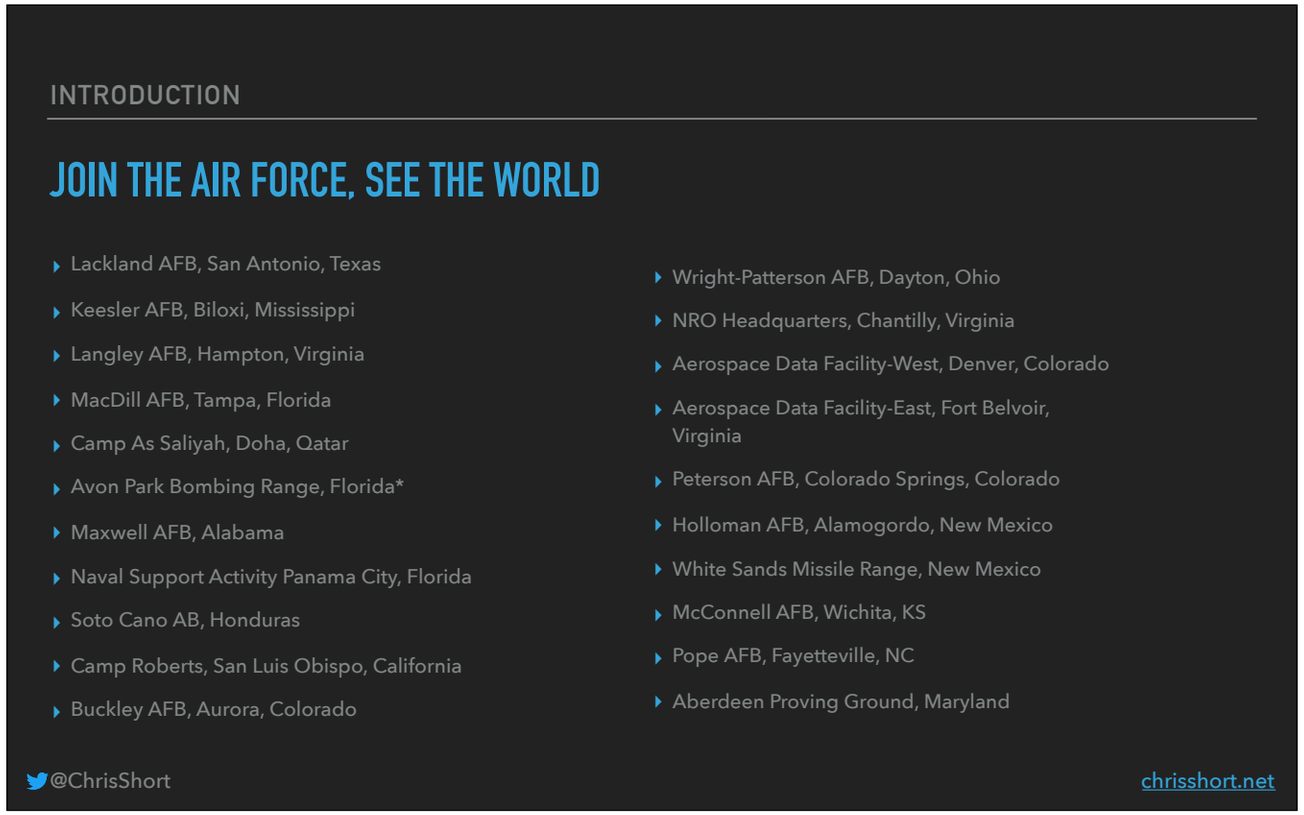
WHAT THE MILITARY TAUGHT ME ABOUT DEVOPS

@ChrisShort

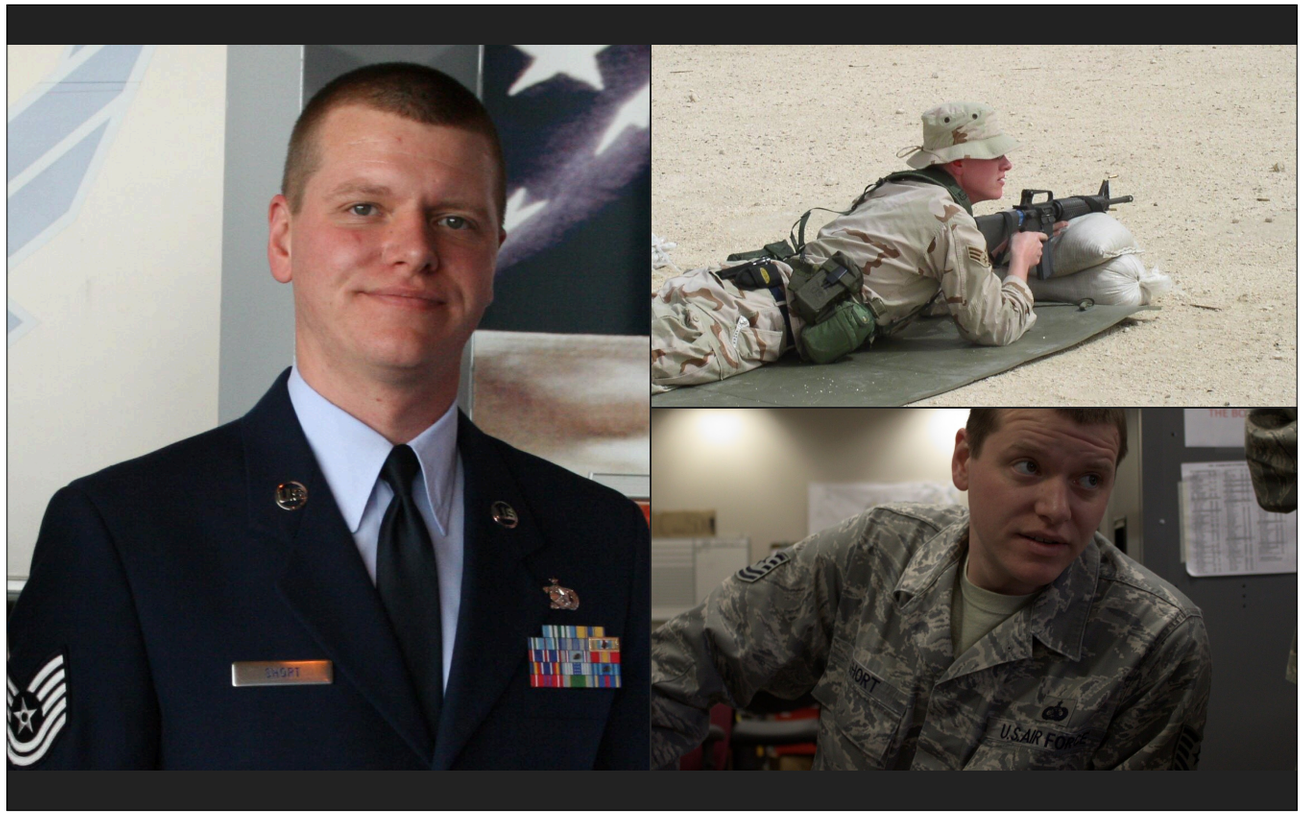
chrisshort.net



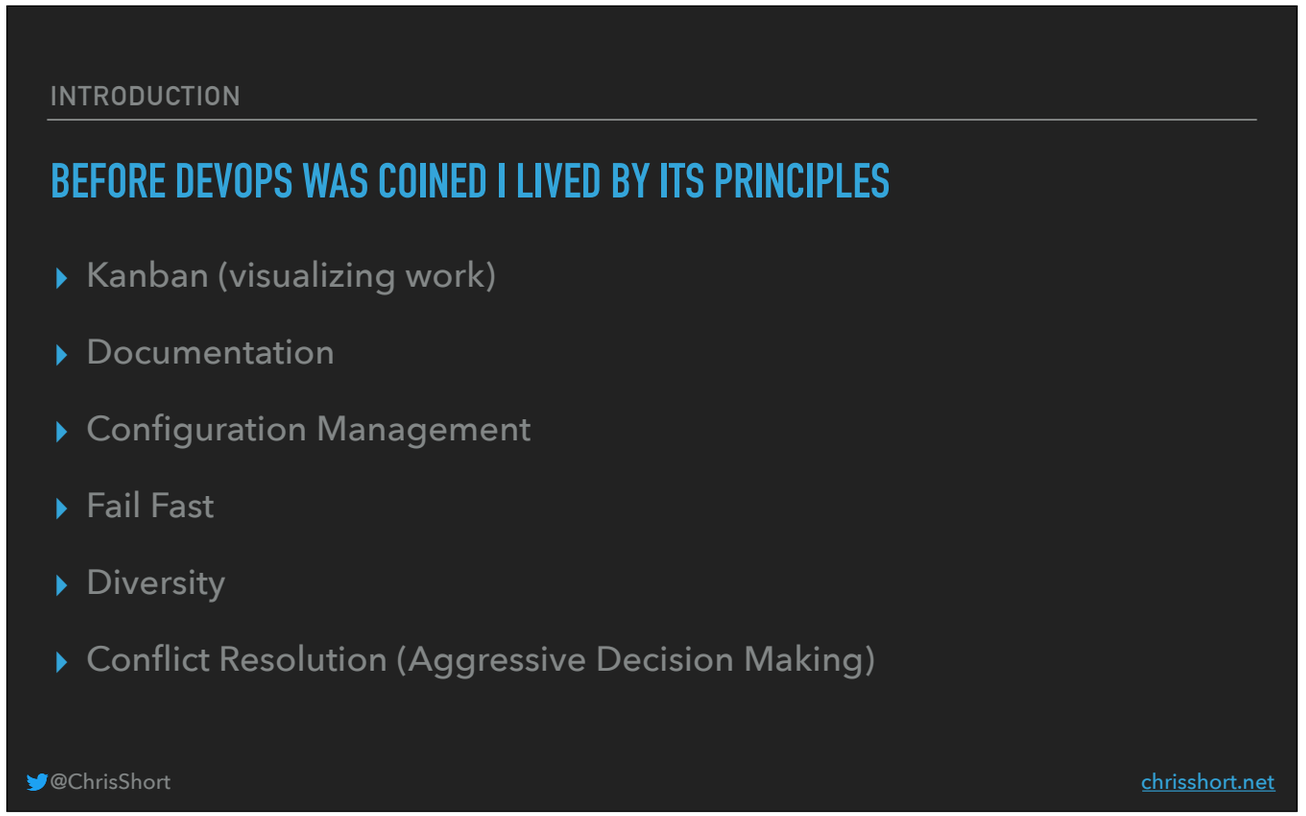
1. So Who Am I?
    1. I'm Chris Short and I've been in IT (in one form or another) for over 20 years (OMG)
    2. Yes... I know my last name is Short. People need to laugh, right?
    3. I had a great career ahead of me while I was in high school
    4. In 1995 I worked in the MIS department of a textile manufacturer in western North Carolina
    5. When I was 18 I worked at a dial-up Internet Service Provider
    6. While working at this ISP I learned about two things:
        1. How cutthroat business could be
        2. Linux and Open Source Software
            1. My favorite distro was Mandrake Linux
            2. My favorite tools were vim and nmap
    7. Then the dot com bubble began to pop
    8. After going to three different high schools in four years I did not want to go to college; I was done with school
    9. What was a kid to do?
    10. I Joined the Air Force
2. I worked in the technology field (which was called communications at the time)
    1. No. I did not fly planes
    2. My official Air Force specialty was Communications-Computer Systems Controller
        1. That is the most government name you could give to any IT job ever
        2. We shortened it to Tech Control for what I hope are obvious reasons



1. Join the Air Force! See the World!
2. I was deployed three times
    1. Qatar (twice)
    2. Honduras (once)
3. I worked for or along side several organizations while I was in:
    1. Air Combat Command
    2. Central Command
    3. Special Operations Command
    4. Southern Command
    5. Northern Command
    6. NSA
    7. National Reconnaissance Office
    8. Air Mobility Command
    9. 82d Airborne
    10. Joint Special Operations Command
4. While I was at Aberdeen Proving Ground I worked on what is called the Joint Biological Point Detection System:
    1. This was a $900 million defense program
    2. It was designed to detect biological weapons attacks and alert military personnel
    3. It was about to horrifically fail Air Force certification spelling almost certain doom for the project
    4. The team asked my leadership if anyone could help them and I volunteered

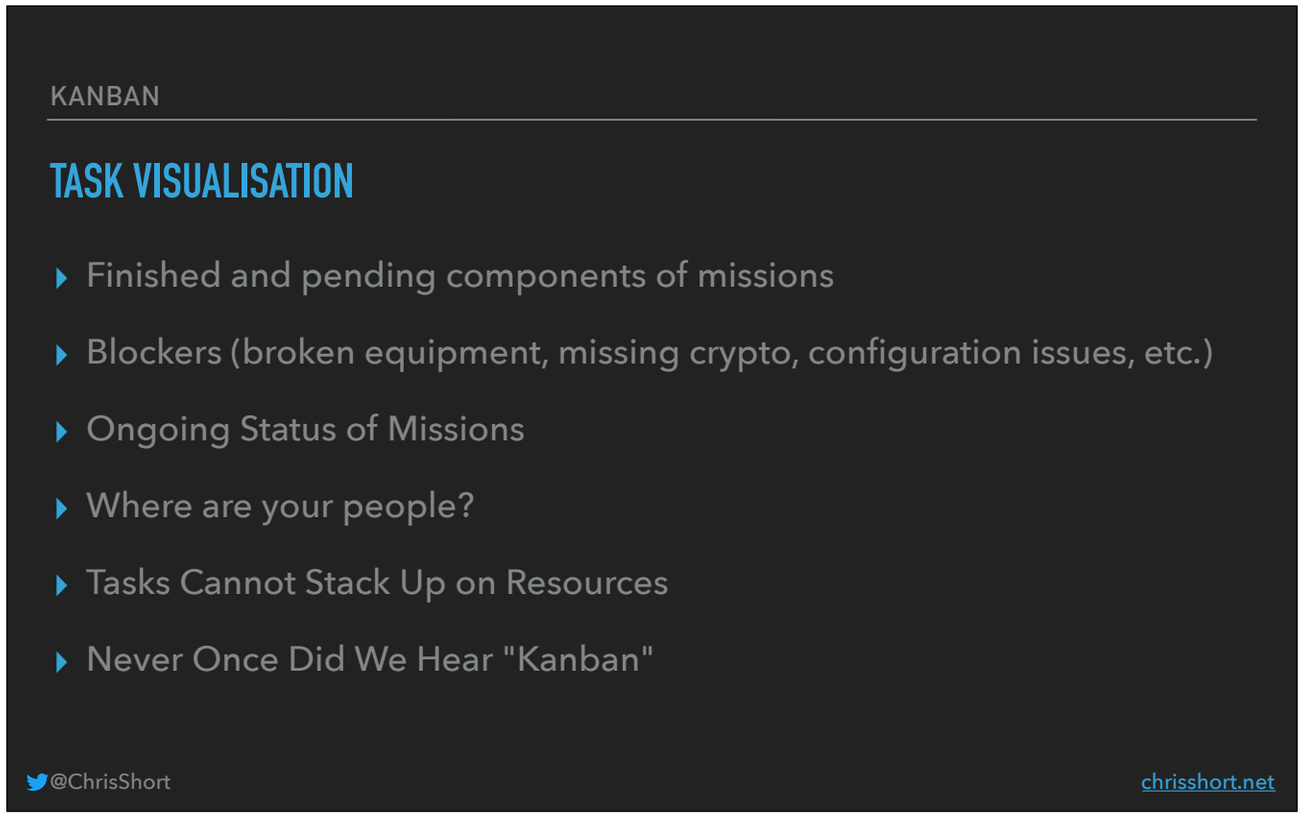


1. Here are some pictures of me from over the years
2. Wow... Look how short my hair was
3. Left: Me at my last promotion ceremony to E-6 in 2009
4. Top Right: M16 qualification course outside Doha, Qatar in 2003 (note the multitool and flashlight)
5. Bottom Right: Me at Pope AFB, NC in 2010

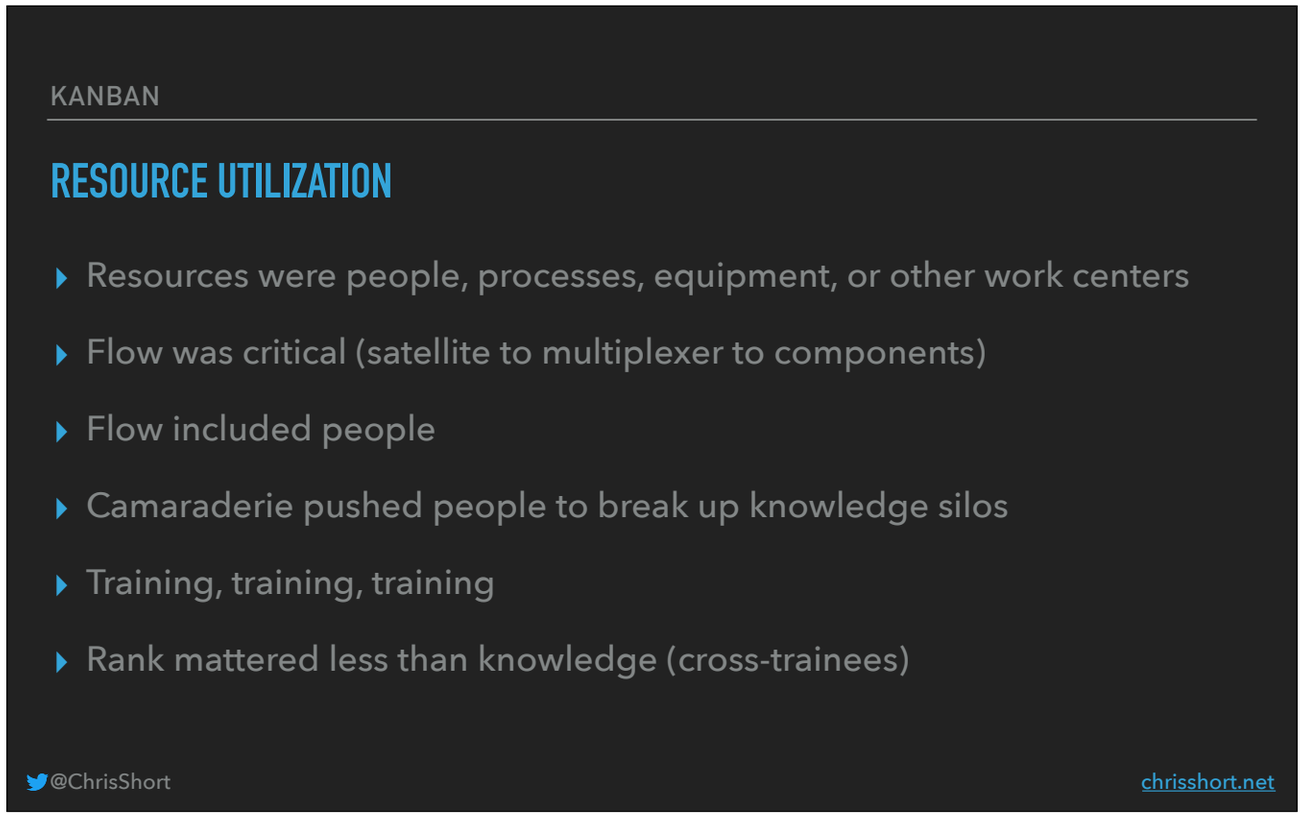


1. What is DevOps? Are you dev'ing? Are you ops'ing?
2. Is there really a perfect definition of DevOps?
3. No, but I think we can all agree on a few core tenets of DevOps
    1. Kanban or visualizing work
    2. Documentation
    3. Configuration Management
    4. Managing Failure
4. There are also some intangibles that the military taught me that have helped me on my DevOps journey
    1. Diversity; both human and idea diversity
    2. Conflict Resolution

1. How do you visualize tasks in the military at the turn of the century? WHITEBOARDS!!!
2. And later on flat screens
3. But let's talk about the whiteboards we were using in the early 2000s



1. Every "mission" consisted of multiple people and components or resources
2. There was often a whiteboard for each division of responsibilities or career specialties
3. We provided status updates on everything, all the time
4. The blockers were very clearly visualized
5. All these whiteboards insured tasks did not stack up on one resource or team
6. You could get the sense of the overall mission status by simply walking around
7. Finishing something was lauded and provided a great feeling
    1. There was a physical event that took place; updating the whiteboard
    2. Going from red to green was like spiking a football and was usually celebrated
        1. I have a love for high fives and I think this is where it comes from
        2. I'd really like some celebrations in JIRA. That would be awesome.
8. We even had whiteboards to tell you where people were:
    1. At a meeting
    2. Lunch
    3. Somewhere else
9. One unintended consequence of this constant status updating was cross-training and cross-utilization of skill sets
    1. If you were in maintenance mode after completing your core components of a mission you helped do something else
    2. That something else could be any number of things from setting up antennas to running cables
    3. It had to get done and we were all on the same team



1. Resource was a generic term for just about anything:
    1. people
    2. processes
    3. equipment
    4. teams
    5. work centers
2. Everything as a resource allowed you to diagram or visualize a work flow at any given moment
3. If a flow included one person and NOT a team or higher level resource it identified a problem
4. Teamwork and "the struggle" encouraged people to break up silos and share knowledge
5. Training was the name of the game
    1. If there was no live mission work you were training
    2. We wanted your job to be almost muscle memory
        1. If X happened, do Y
        2. If Y failed, do Z
6. The military has a concept of cross trainees:
    1. These were people who trained in one career field and for manpower reasons volunteered to re-train into another career field
    2. These people were given the same foundation of training everyone else in that career field
    3. But, since they had typically been serving for four or more years they would typically be mid-level managers with no on the job technical knowledge
    4. Rank mattered a lot less as a result with cross-trainees since they relied on the younger folks to train them

DOCUMENTATION

SO MUCH DOCUMENTATION

- Binders existed for everything
- Needs Based, Hands-On Guides
- Shared drives with backups
- Thumb drives (before the ban)
- Well Documented Requirements Established Before Work
- What's The Fix?

Photo: Seth Sawyers

1. Moving on to the topic of documentation
2. Show of hands... Who has fantastic documentation in their organization?
3. The military had documentation on documentation
    1. They had regulations dictating how to categorize documentation
    2. They had regulations dictating how documentation should be made available in the work center
4. All military equipment came with guides or manuals
5. All relevant procedures were available
6. Bookshelves full of technical orders
    1. Technical Orders were detailed maintenance guides
    2. These were life lines when things broke or went wrong
7. Locally generated documentation was frequently bound and shelved (later stored digitally)
8. One good thing the military did was document all requirements for mission completion
    1. These were detailed and documented as well
    2. Agreed upon between all involved parties prior to execution
    3. This did not prevent the odd curveball but it minimized them
9. Digital local and centralized document repos were often implemented
    1. If these failed you had the binders (which were updated regularly)
        1. There was a regulation on how to update documentation
    2. Before they were banned in the mid-2000s folks typically carried a thumb drive with docs and guides
10. Think of a few dozen people wandering from Windows machine to Windows machine plugging and unplugging USB drives

DOCUMENTATION

WHAT'S THE FIX?

- Wiki-based (SharePoint... Eww)
- Searchable
- Solutions to odd problems
- Documentation for one off fixes
- Uniquely Named
- Not Forgotten
- #WTF

1. What's The Fix documented solutions to those odd or every once in a while problems
2. "What's The Fix" was an unclassified Microsoft-style wiki on a SharePoint site
3. Wiki based documentation was not common surprisingly
    1. The concept of anyone generating good documentation was new-ish to military leadership
    2. "What's The Fix" Became invaluable as an index for weird issues
    3. This trained people to search the wiki for documents as opposed to a share drive
4. What types of odd problems would live on What's The Fix?
    1. Something like fixes for Heartbleed would have been documented there
5. Why did it get the name "What's The Fix"?
    1. It was catchy, easy to remember, and maybe... slightly... unprofessional
    2. Comedy during a problem solving endeavor is often helpful

MAKING SENSE OF THE RIGIDITY

- Military regulations are intentionally explicit
- Technology outpaced regulation re-write cycles
- Formal training only provided a foundation to build on
- Understanding the regs and instructions only took you so far
- Common sense is not common

1. Most people view the military as this rigid, regemented structure
    1. That's very true at a high level
    2. But on the ground the structure had to give way to specialized skills
    3. The Air Force realized they couldn't regulate technology space too tightly or it wouldn't be adaptable in wartime
2. Formal Training was the foundation you built from
3. Real training came from learning systems and skills on the job
4. On the job knowledge and training evolved at the speed of technology
5. Just going by the regulations only took you so far
    1. Consider military regulations the bare minimum
    2. There was frequently ways to go above and beyond
6. For example, putting a $2,000+ Cisco switch in a bathroom above a toilet was a by the book install job
    1. You can't regulate commonWater and electricity not mixing was not in the regulation apparently
    2. One bad plumbing incident and this part of the building loses network connectivity
    3. On the job knowledge often won the day
        1. That was not the case in this picture4
        2. Common sense is not common

CONFIGURATION MANAGEMENT

VERSION CONTROL???

- We are **so** lucky to have Mercurial, Git, Subversion, etc. NOW
- Versioned configs were difficult to maintain
- Access controls had to be maintained very tightly
- Standards were maintained and documented but not programmatically enforced
- CTRL+C, CTRL+V could crash networks (thanks Cisco)
- Your disaster recovery plan was based off little more than hope

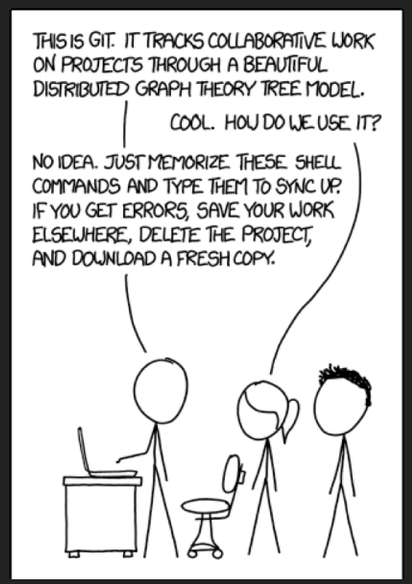


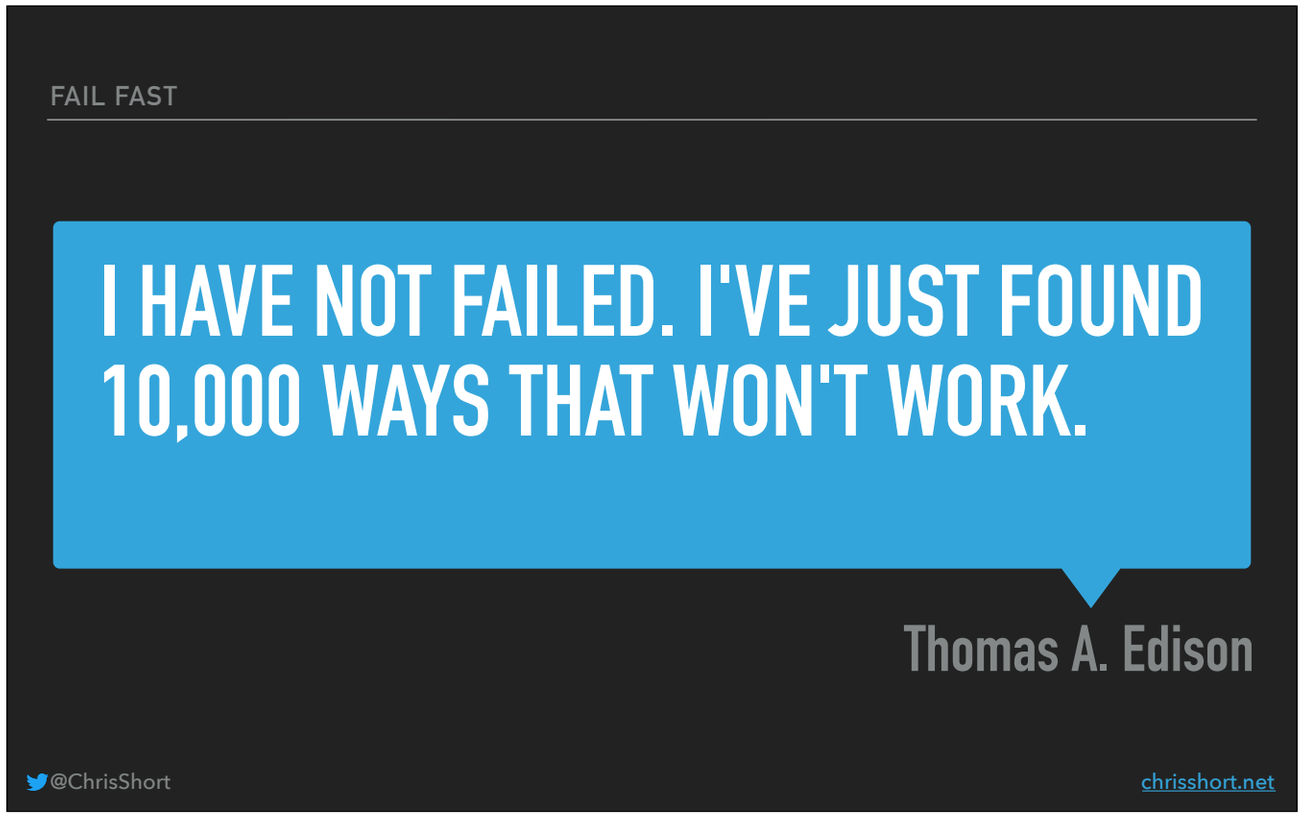
Photo: XKCD

1. Let's discuss what configuration management was like during my time in the Air Force
2. First thing to keep in mind, good configurations were like gold nuggets
    1. If you maintained a good config for something people clamoured for it
    2. You would make copies upon copies of configs to make sure you had it when needed
3. I DID NOT have Mercurial, Git, or Subversion
    1. The concept of versioning changes in files was new in the Windows enterprise
    2. We are so incredibly lucky to have these version control tools available
4. We did our best with Windows File Servers
    1. git branch was a right-click, make new folder
    2. We would have all kinds of access control problems because of personnel changes and changes in Active Directory
    3. Even more challenging was enforcing naming standards for folders and files
5. An error or omission in a stored configuration could crash a network with a simple copy/paste
    1. Cisco equipment activated configurations with every line break
    2. Murphy's Law struck frequently
    3. If something broke due to a copy paste error it would break on a Friday at 4 PM during some unique event or mission
6. Disaster recovery was hoping no one screwed up your backups or access to them
7. As a result, a lot of us carried around thumb drives with what we thought were the latest versions of configurations

TESTING... HA, HA, HA

- Dev? Test? Stage? No budget.
- There was once a world with no Ansible, Puppet, Chef, etc.
- Lucky to have spare equipment to test on
- Find lowest impact users and deploy

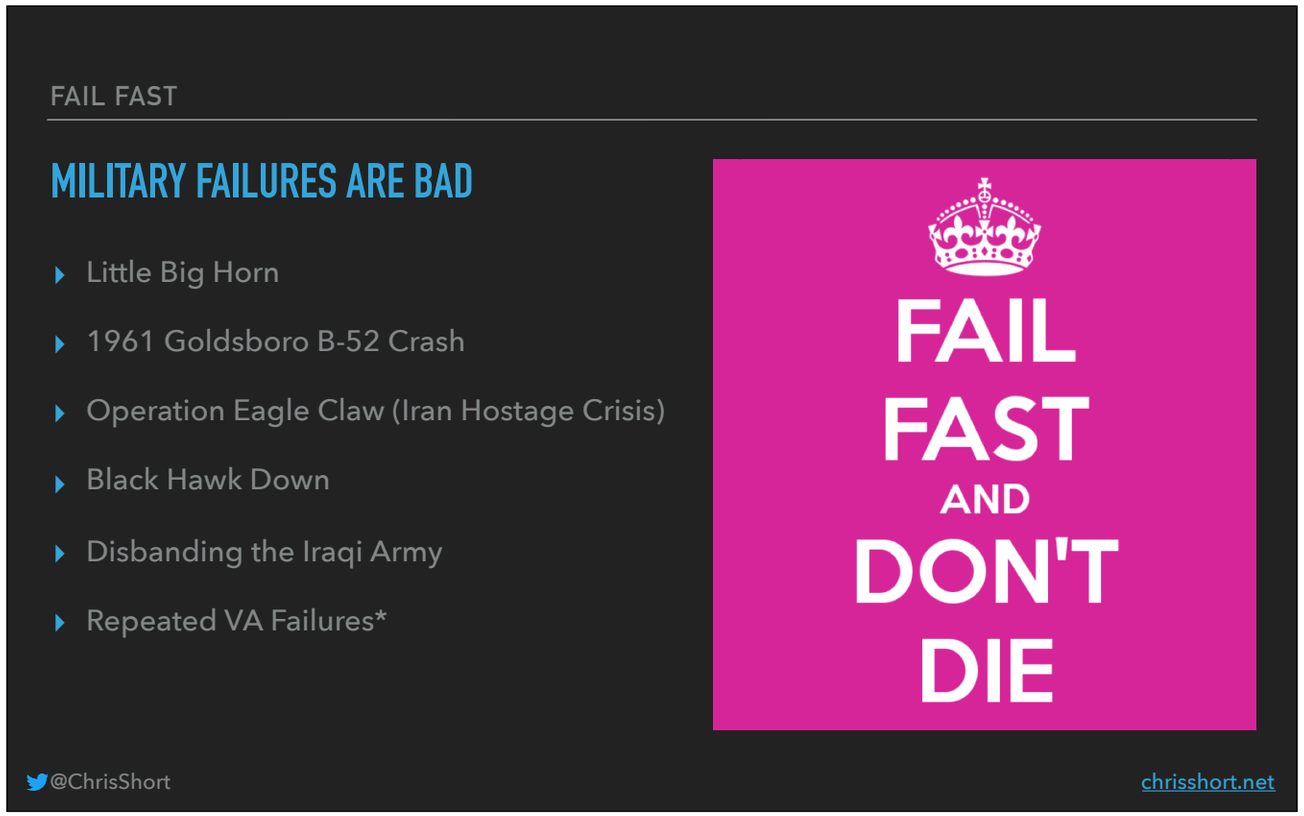
1. We all know someone like this, right?
2. Sadly the military did not have large budgets for labs or test equipment or test environments
3. If you wanted to test configurations you generally had two options:
    1. Spare equipment used in case of failures or
    2. Equipment that is about to be retired
4. Testing at scale or under production like load just did not happen
5. We could not promote dev to stage and catch issues before deploying to production
6. There was no automation either
    1. If you were pushing out changes often times it had to be done manually
    2. As a result changes had to be pretty bullet proof
7. So when you did go to production?
    1. You had to know that Test Subject Customer A was not going to be greatly impacted or was notified well in advance
    2. And you always, always had a back out plan



As you read this quote think about what would have happened if Steve Jobs failed 10,000 times making the Apple II.

Apple and the PC market as we know it would probably look a lot different

Let's talk about failure and the what I think is a great term "failing fast"



1. Listed here are some well known US military failures
2. I think we can all agree that all of these are failures of one sort or another
3. The 1961 Goldsboro B-52 Crash is an interesting case we will discuss further in a minute
4. Operation Eagle Claw was the Iran hostage crisis rescue operation that failed
    1. In this case, think of the old silo mentality like the dev and ops of old
    2. The mission planners (developers in this example) failed to create a solid delivery process
    3. The developers intended operations do things a certain way
    4. The ops team showed that the reality was not what the developers intended
    5. The mission failed as a result
    6. Lives were lost
    7. The CEO (President Jimmy Carter in this case) was then let go by the board of directors (the US voters) because of the handling of the Iran Hostage Crisis

FAIL FAST

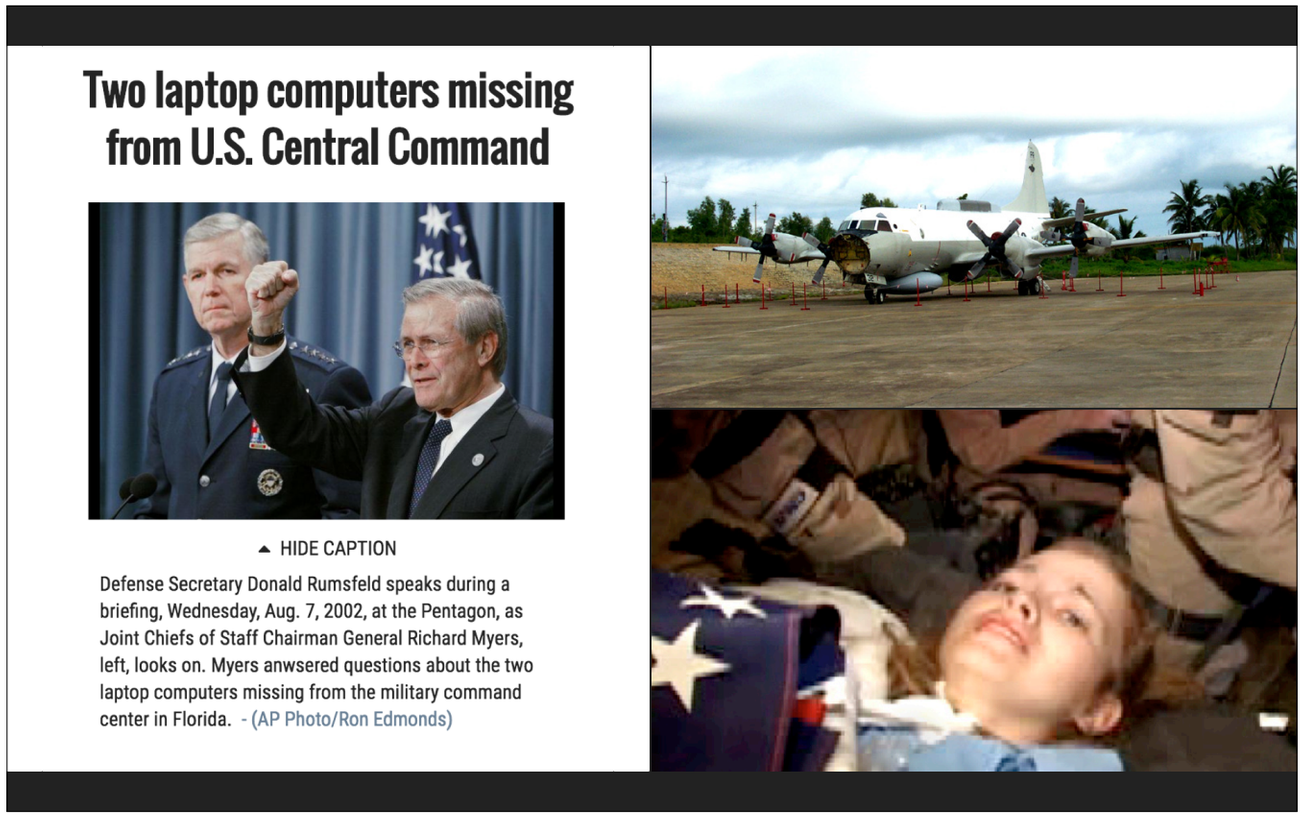
WHEN FAILING FAST GOES WRONG

- 1961 Goldsboro B-52 Crash
- B-52 broke up in mid-air
- Was carrying two Mark 39 thermonuclear bombs (3-4 megatons each)
- One bomb deployed a parachute (only one step away from detonation)
- The other bomb plummeted to earth
- One high-voltage switch prevented that bomb from detonating
- Amazing postmortem

@ChrisShort

chrisshort.net

1. 1961 Goldsboro B-52 Crash
2. A B-52 Stratofortress was flying a Cold War alert flight
3. These were rapid response flights ready to strike the Soviets if they attacked
    1. This was the US answer to what the military believed was a superior ballistic missile force maintained by the Soviets
    2. To improve response times some safeties were disabled once the aircraft was in flight
4. The B-52 had a structural failure and broke up in mid-air
5. The two Mk39 bombs plummeted to earth
    1. One parachuted to earth
    2. 3 of 4 safeties were somehow flipped off during the plane breaking up
6. The other bomb became a 700 MPH nuclear lawn dart
    1. The parachute on this bomb did not deploy
    2. In 2011, one researcher stated during an interview that **one** high voltage switch prevented the creation of a "Bay of North Carolina"
    3. The core (or pit) of the bomb was safely recovered
    4. Full recovery was next to impossible given the swampy conditions
7. Some where buried about 180 feet down in a marshy part of eastern North Carolina is some weapons grade plutonium and uranium that the US government could never recover
8. Needless to say this "fail fast" did not go as intended
9. The subsequent recovery efforts and internal reviews created quite the postmortem
    1. As a result of the crash the military (Operations) finally woke up and listened to the weapons developers who were adamant about making nuclear weapons safer

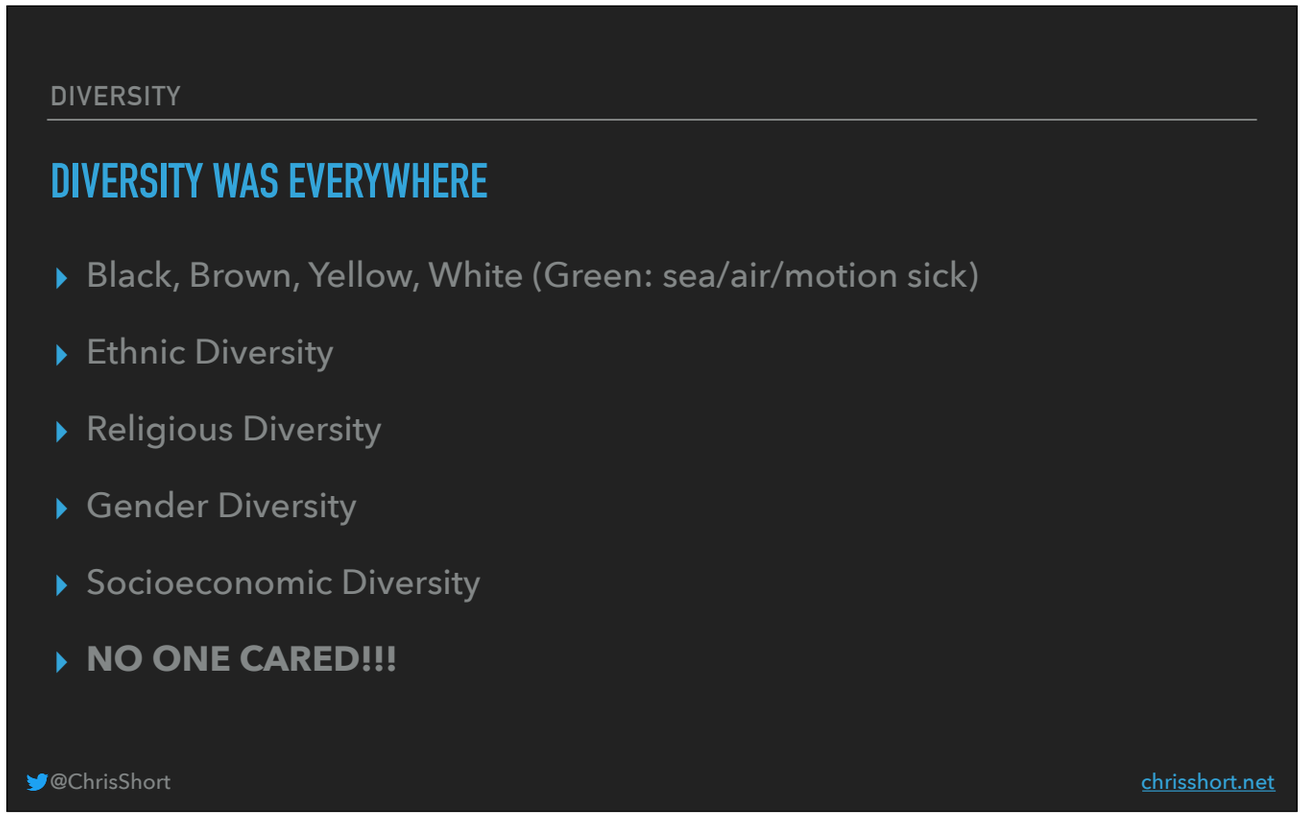


1. This slide represents some failures that happened while I served
2. Top Right:
    1. Hainan Island Incident
    2. In April 2001, a US EP-3 collided with a Chinese J-8 fighter
    3. This forced the landing of the EP-3 in China
    4. There was highly sensitive classified material in use by a large swath of the Department of Defense
        1. One of my jobs was to distribute the message to multiple units destroy the classified material
        2. I also worked to phase out my work center's use of the compromised cryptographic key material
    5. We practiced such a failure numerous times so we were ready for it but had never done it in real life
3. Bottom Right:
    1. During the Jessica Lynch rescue mission the power went out at my site inside Central Command's Forward Headquarters
    2. We had to safely shut down equipment
    3. None of the cool video feeds or interactive maps were available for the team at my site
    4. We directed mission staff to use radios to interact with mission personnel
    5. We had previously requested generators but were not provided any until after this power outage
    6. One officer was under the impression a UPS would last for several hours not minutes
4. Left:
    1. During my time working in Central Command's headquarters in Tampa, FL four Top Secret laptops were stolen
    2. These laptops were used to process highly sensitive information and were supposed to be accounted for at a minimum every 12 hours
    3. Laptops were known to be missing in the morning at shift change

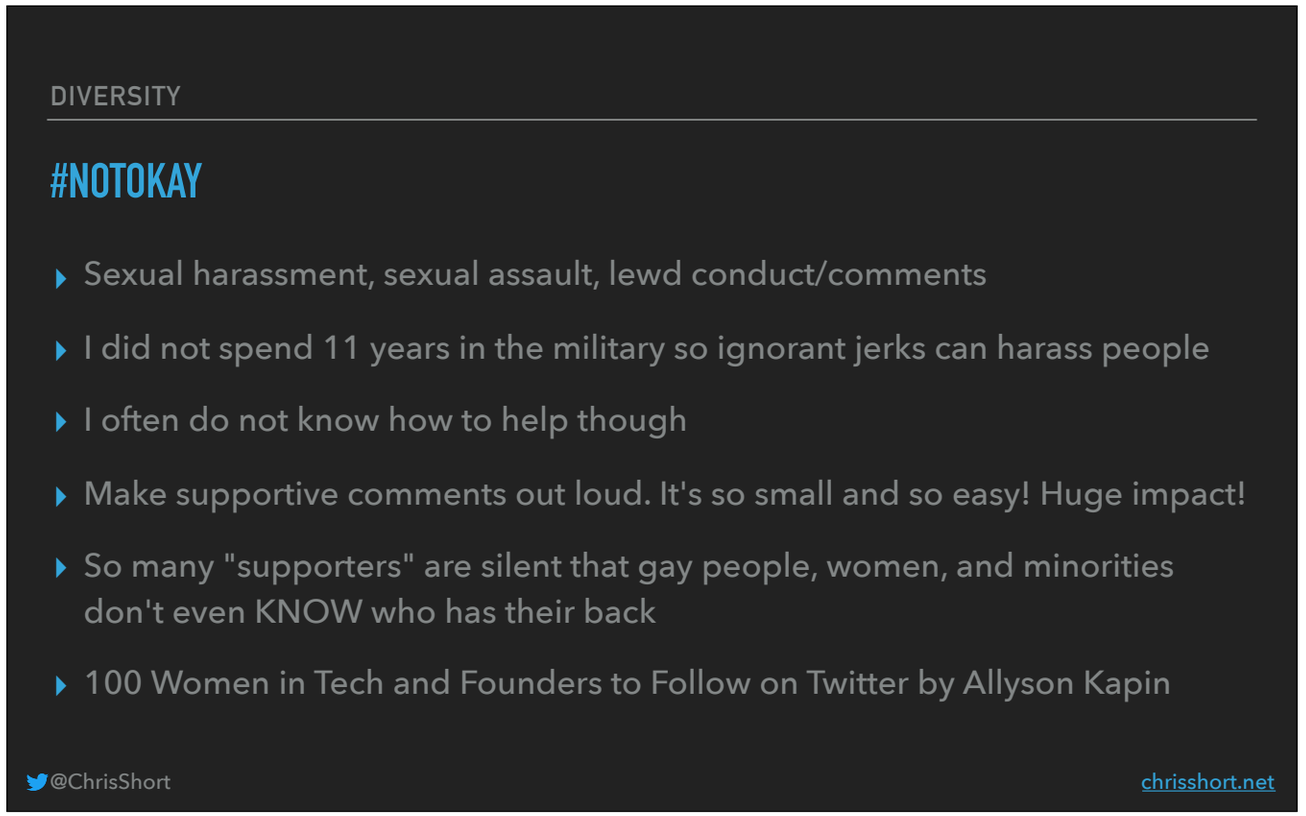
1. Most things in government are built by the lowest bidder
2. You have to practice failing because as a result of this purchasing practice failure is likely to happen
3. The most important part of failing is how you respond to it
4. Practice allows you to respond faster
5. Practicing failure shows you how things operate in those failure conditions
6. Building "muscle memory" allows you to iterate through one scenario while discussing other scenarios more calmly
7. I'm lucky that the Air Force taught me how to do this while covered in sweat and dirt so at a desk it's easy
8. "Plan Z" was a phrase I developed
    1. If everything failed then what were we going to do?
    2. Plan Z is the hail mary pass at the end of the ball game when you are down by four
    3. If all else fails this is what you are going to do to make sure you get back to semi-normal operations
    4. Always have a Plan Z

Let's talk about diversity

Diverse groups are great for DevOps teams because they provide unique perspectives



1. In the military, diversity was everywhere
2. It didn't matter what color you were
    1. Even green folks were welcomed
    2. Just barf somewhere else please
3. I worked with women and men
4. I worked with a lot of different ethnic groups
5. I worked with a lot foreign nationals
6. I worked with Christians, Jews, Muslims; you name it
7. I worked with rich kids and poor kids
8. No one cared what you were in the middle of a mission as long as you were reliable
9. This is how it should be
10. You should be judged on merit, performance, and potential



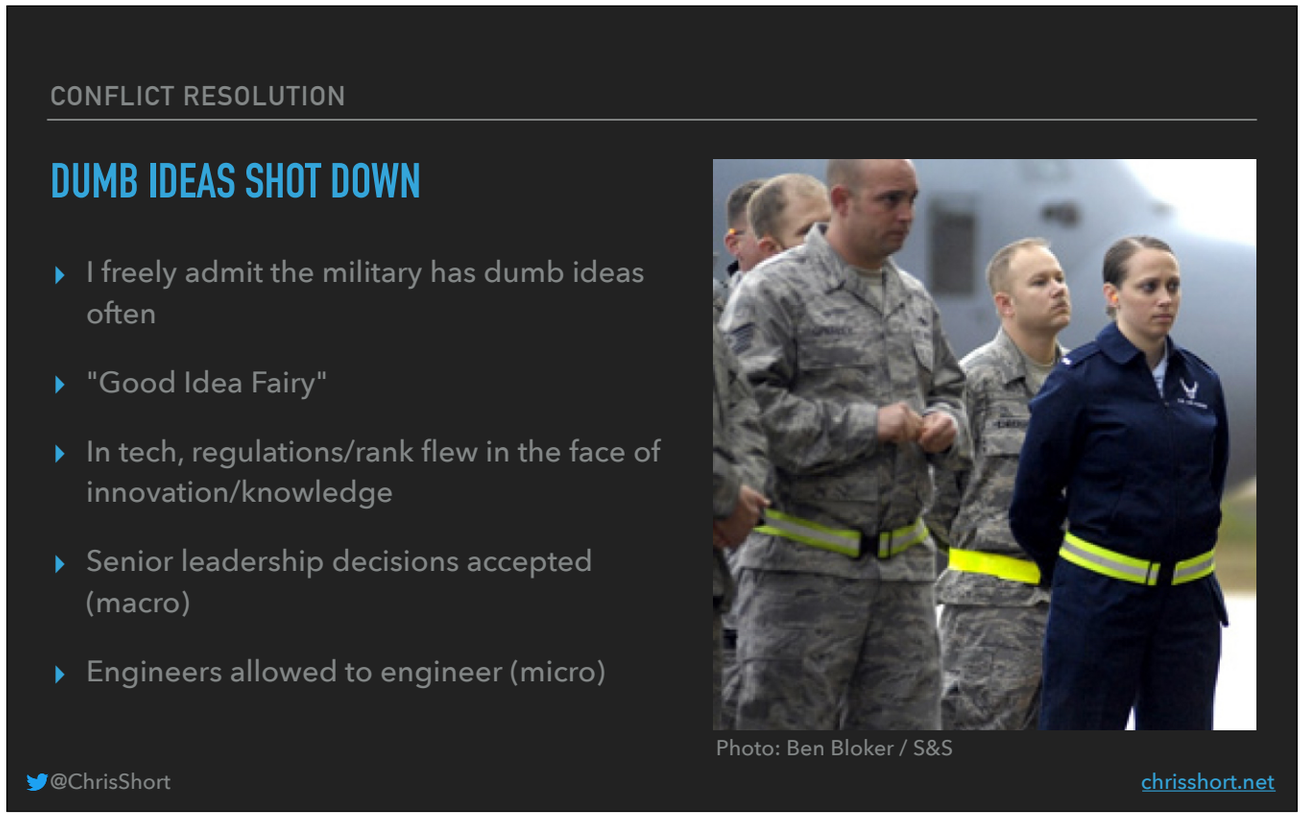
1. What is not okay is sexual misconduct and treating people poorly because of who they are
2. I did not spend 11 years fighting so you can be a jerk to your fellow humans
3. If you were being a real jerk you were dealt with
    1. Sometimes it could be handled by your peers
    2. Others it had to be handled by the military justice system
    3. The hard part was figuring out the gray area
4. On the outside, in civilian life, it's amazing to me this is such a big issue today
5. It is time we all step up and help make everyone feel welcome
    1. I have my struggles that I am dealing with
    2. I'm instantly suspicious of people I don't know
        1. What do they want from me?
        2. Are they a threat?
    3. I am alarmed by the presence of traditional Arab attire
    4. I am trying really hard to break myself of these things
6. I often don't know how to help
7. Here is what I am doing to try to solve this problem?
    1. I am saying something when I hear or see someone being a jerk
        1. I can talk to the jerk and get them to be better in most cases
        2. At the very least I let them know their conduct is not appreciated
    2. Make supportive statements out loud

FIGURE IT OUT

- Diversity in ideas is a great thing
- Taught color blind folks to make network cables (terminating CAT5e/CAT6)
- If someone is willing to learn you must be willing to teach regardless of race, color, religion, national origin, age, sex

Photo: wocintechchat.com #WOCinTech

1. I cannot solve this problem of including people by myself
2. I need your help too
3. Let me tell you about my story of when I first got out of the Air Force
    1. I was medical separated at the end of 2010
    2. Got a job in early 2011 at a company using a lot of open source software
    3. I was using Linux before I joined and on and off during my time in the Air Force
    4. But in 11 years a ton changed in open source software and Linux
    5. I was totally terrified after the end of my first week on the job
    6. IT WAS HARD
    7. By the end of my second week I went to my VP and told him things were not working
        1. Documentation was bad
        2. Coworkers were cold
    8. My VP teamed me up with two people that are still good friends today and we slayed problems together
    9. All it took was a nudge and I was off and running
    10. The key was identifying problems and applying different ideas to solve them
4. Diverse ideas and prospectives are a great thing
    1. We all need to embrace diversity in a meaningful way
    2. As a veteran I had a very unique perspective and ability to make a decision and go
    3. Someone else would approach life differently than I did because of their experiences
    4. Isn't this part of what DevOps is about?



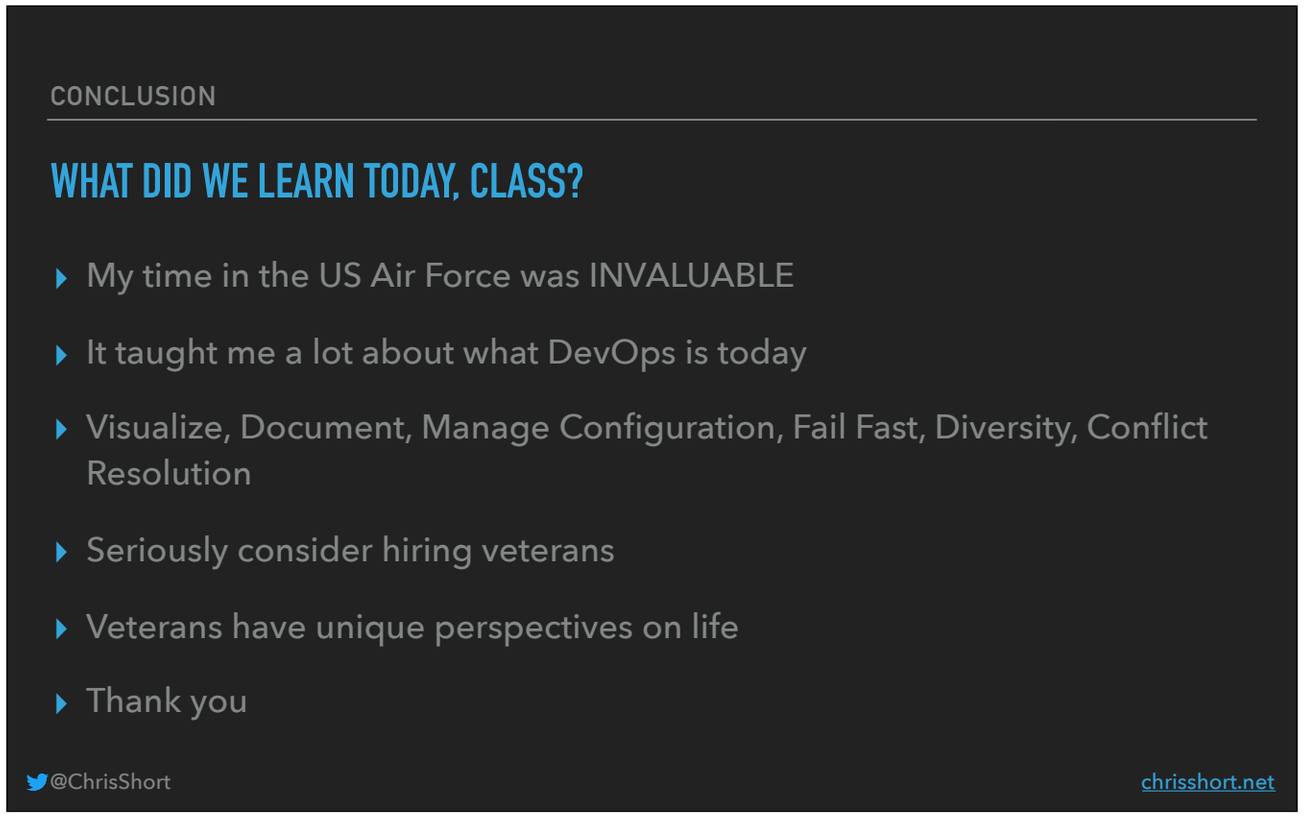
1. Let's discuss conflict resolution and what the military taught me about it
2. But first let me give you a breaking news flash:
    1. The military had some really dumb ideas at the macro level
    2. Camouflage with reflective belts was for "safety" reasons
        1. We ended up wearing reflective belts in places where being seen could be very bad
        2. The reflective belt became the butt of countless jokes in all branches
3. This idea was brought to us by the "Good Idea Fairy"
    1. What is the good idea fairy?
    2. The urban dictionary definition is spot on:
        1. good idea fairy: Military term; An evil mythical creature that whispers advice and ideas into the ears of military leadership, causing hundreds unnecessary changes and countless wasted man-hours every year. The Good Idea Fairy should be shot on sight if she is seen in your area.
    3. Do not be the good idea fairy in your teams
    4. Obviously, if the CEO is the good idea fairy then you have to accept it to a degree
        1. But, even in the upper echelons of Air Force leadership there were feedback loops
        2. Demonstrate tangible issues with "good ideas" to the good idea fairy
        3. Don't come to people with problems
        4. Come with solutions
        5. The Good Idea Fairy will change with enough pressure
4. Inside the tech fields in the Air Force the regimented regulations and rank structure flew in the face of innovation and knowledge
    1. Leaders were often forced to recognize they were not subject matter experts

CONFLICT RESOLUTION

AGGRESSIVE NEGOTIATIONS

- Heated arguments about ideas and philosophies
- Almost religious discussions
- In a room full of leader mentalities only the best ideas rose to the top
- Don’t let the perfect be the enemy of the good
- If you pulled rank, you sucked

1. Let's discuss Conflict Resolution in terms of "Aggressive Negotiations"
2. Often times peers have competing ideas
    1. Some times these competing ideas led to arguments
    2. Some times arguments amongst service members were of near biblical proportions
3. We were all had leadership training that demonstrated the best ideas were easy to explain and implement
    1. Once an elegant solution was provided it frequently rose to the top and was implemented
    2. Notice I said elegant, not perfect
4. You cannot let the perfect be the enemy of the good
    1. In a tent in the middle of the desert it's almost like you're in outer space
        1. All you have is what you brought with you
        2. Getting someone else to bring you something you are missing was unlikely
    2. You can't wait for the perfect tool, solution, or feature to show up
        1. Sometimes it is better to embrace a good idea when yours is a better idea but maybe slower
        2. Work to make the solution as good as possible and then champion fixes after implementation
    3. You have to release an operational product or function not necessarily a perfect one
        1. This sounds a lot like "Ship It" doesn't it
        2. Balance is very important in life, negotiations, AND in DevOps
5. Last but certainly not least if you think your idea is the best because of your position or rank you are going to have a bad time
    1. In the military these people were often worked out of feedback loops
    2. The "my idea is superior because of rank" people often looked like fools to their peers and subordinates



1. Even after being injured I would not change anything about my life
2. Joining the Air Force at the time I did taught me a ton of skills that are relevant to DevOps today
    1. Visualizing tasks
    2. Documenting processes
    3. Managing configuration
    4. Failing Fast
    5. The military taught me a lifetime's worth of lessons in these core competencies
3. I also learned that diversity is an asset to good teams
    1. Perspective goes a long way
    2. Innovative ideas come from a diverse group
4. Conflict resolution was something I was not good with until I was in the Air Force
5. I would be remiss to not take this opportunity to mention a cause I'm passionate about:
    1. Veterans have great skills that are going to help your teams in the long run
    2. I implore you to seek out veterans when looking for candidates
    3. We might have to go to the VA hospital every once in a while for a check-up or procedure
    4. We are hard working
    5. We are objective and goal driven
    6. We can think on our feet
    7. We are definitely not robots that have to be programmed
6. Thank you to the DevOpsDays Detroit volunteers and sponsors for this opportunity