

What legacy government organizations need to advance the state of DevOps



November 2017

Robin Yeman
Lockheed Martin Corporation
Robin.Yeman@lmco.com



© 2017 LOCKHEED MARTIN CORPORATION

Dr. Suzette S. Johnson
Northrop Grumman Corporation
Suzette.Johnson@ngc.com



Approved for Public Release #17-2336. Distribution Unlimited.
Date Approved: 11/8/17

© 2017 Northrop Grumman Systems Corporation

Abstract

Two large scale legacy government organizations have had similar journeys in transforming their companies to Agile and DevOps. The journey continues. In order to continue the transformation we need to evolve and build the next generation engineer.

Building cross functional teams with broad skills to deliver value require building new engineers. Currently universities are educating stove piped engineers to solve the last generation of problems. The future of our organizations requires a wider curriculum. If we cannot hire out of universities, we need to build them internally.

Introduction

Robin and Suzette in London!



Who we work for....

Northrop Grumman Overview



- Leading global security company
- \$24.5 billion sales in 2016
- Leading capabilities in:
 - Autonomous Systems
 - Strike Operations
 - Aircraft and Spacecraft Design, Integration and Manufacturing
 - Intelligence, Surveillance, and Reconnaissance
 - Cyber and Intelligence Mission Solutions
 - Missile Defense
 - Space Exploration
 - Advanced Technologies
 - Health IT



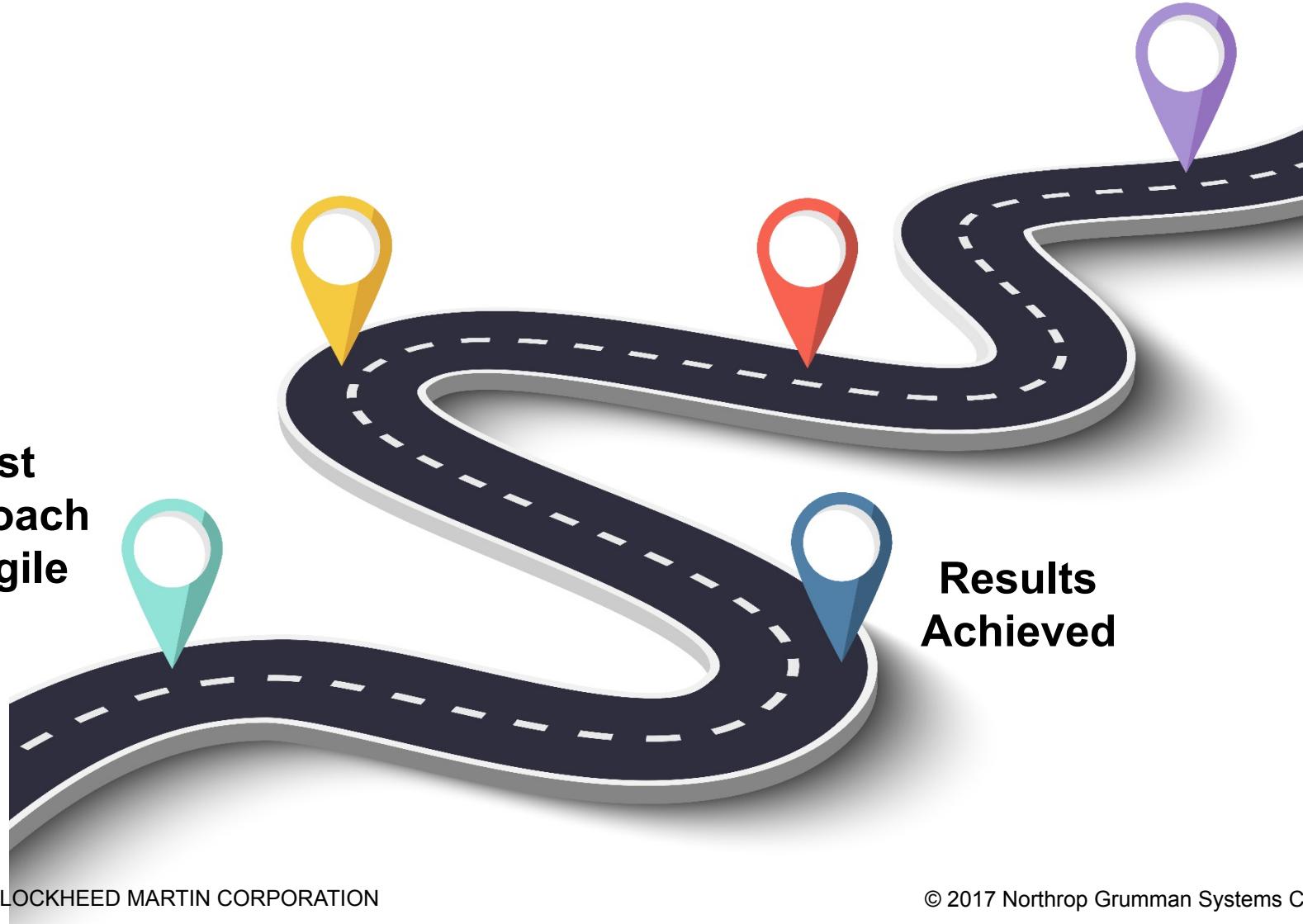
Focus on Performance

Lockheed Martin



WE'RE ENGINEERING A
**BETTER
TOMORROW**

Journey to Agile and DevOps



Suzette's First Adventure

NORTHROP GRUMMAN

Problem:

- 2005 a large data collection system to support the warfighter
- Needed regular delivery of information to the field
- 10 Teams of (~10 people; 125 person team as part of a larger systems integration environment.)

Actions:

- Read everything I could about Agile & attend training (Ken Schwaber / Mike Cohn)
- Trained the team after just 'reading a book' didn't seem to be enough
- Focused on scaling agile, systems engineering, quality, security, transparency
- Made some mistakes = quality a big issue early on

Outcome:

- Great award fees from customer
- Delivered on budget and met schedules on a three month cycle
- Focused on automated testing and improving lead time (<12,000 tests per day)
- Defined the art of the possible and started meeting with great minds every other Thursday during lunch

Lessons Learned:

- Identify small steps of improvement with goals and metrics
- Be clear about the goal and know where you are headed
- Whole team collaborative planning
- Goals around automated testing, intentional architecture, and you are never done

Robin's First Adventure



Problem:

- Classified Asset Management System – 2 years (\$80M)
- Organization required Agile – 2002 // New Approach
- 8 Teams of (8-10)

Actions:

- Read everything I could about Agile & Attend Training (Ken Schwaber / Mike Cohn)
- Understand Contract / Vision / Customer
- Engaged the teams define rhythms and product backlog
- Made some mistakes

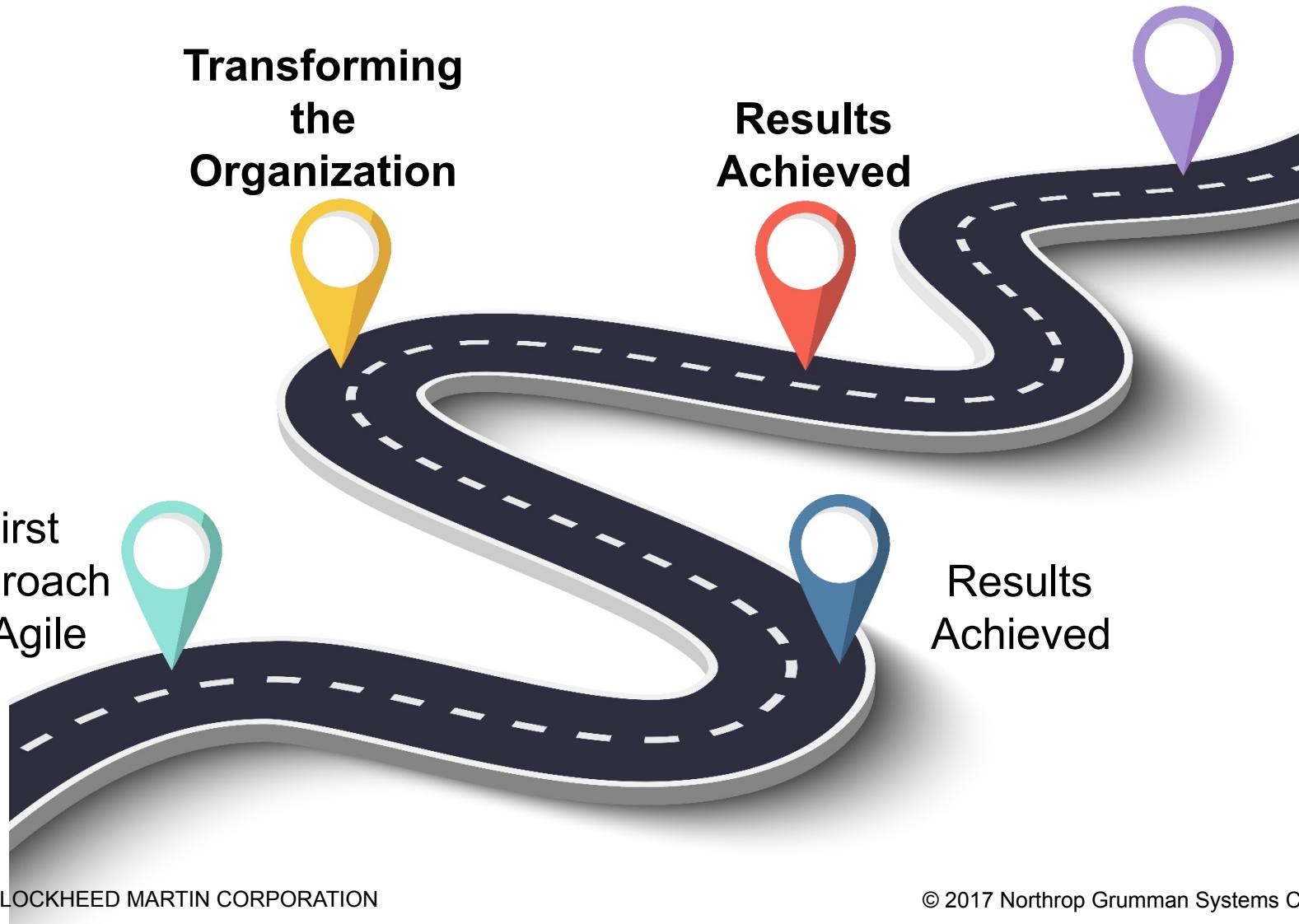
Outcome:

- Multiple 100% Award fees from customer
- Delivered on budget, slightly over schedule
- Made a lot of long term friends
- Defined the art of the possible

Lessons Learned:

- Communicate Success and Failure early & often (Own results)
- Always Begin and End with the team
- Include everyone in the teams (Not just Software / Systems / Test)
- Measure results and review regularly

Journey to Agile and DevOps



Transforming the Organization

Internal Perspective

Problem:

- Growing Agile/DevOps practices at scale in unique domains and across the value stream
- Creating a common mindset and address misconceptions
- Existing functional silos and internal processes (HW/SW/IT/PMO/Subcontracts)
- Building the skills and culture
- Dealing with existing processes and compliance concerns

Actions:

- Agile Center of Excellence, Internal Coaching, Build the infrastructure
- Define the delivery pipeline, tools, automation, with training and coaching
- Transformation strategy and continuous improvement
- Don't lose site of the end goal

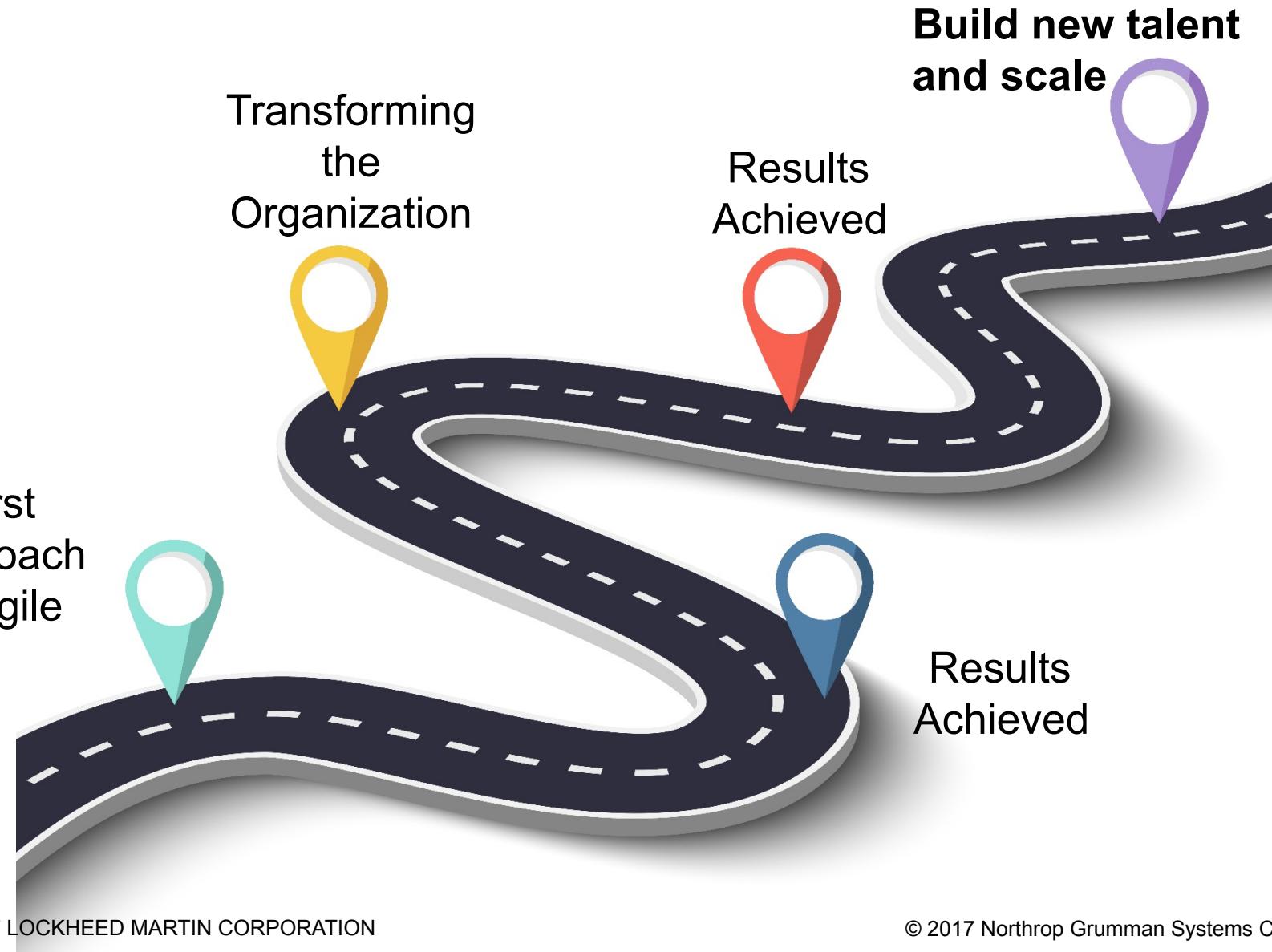
Outcome:

- Faster rate of adoption within the company
- Improved quality with significant improvements on cost and schedule performance
- Increased in active communities of practice to share experiences and new ideas
- Increased awareness and conversations on how we can deliver better
- Increased customer engagement
- Increased visibility (good and the bad)

Lessons Learned:

- Extend Agile and DevOps principles into areas beyond software
- Look across the organization and ensure internal procedures (auditable) enable the change
- Build more champions than you think you need; define a path for their growth
- Build Infrastructure / Automate / Increase CM and train/coach the teams
- Train at all levels of the organization for common understanding, benefits, language, etc

Journey to Agile and DevOps



Building New Talent and Scaling

Problem:

- Bringing in new talent to support Agile and DevOps practices at scale
- Lacking diversity of skills (college curriculums still the same)
- Not enough cross functional teams
- Legacy technology
- Rapidly pace

Actions:

- Engage with universities on curriculum
- Mentor new employees in cross functional skills
- Leverage communities of practice (provide guest speakers)
- Provide opportunities for learning (secure coding, innovation challenges, IRAD)
- Subcontract to small businesses with niche skills

Outcome:

- Enthusiastic teams
- Engaged customers
- Innovative products
- Focus on business outcomes and customer satisfaction

Lessons Learned:

- Measure employee satisfaction / Net promoter score
- Attend external meet-ups for new ideas
- Expose as many people you can to new ideas frequently
- Provide psychological safety to enable experimentation
- Its not failure until you quit, its getting to the next known state

Our Career Lessons Along the Journey

Lessons

- *Constantly* be looking to learn new things that solve customer problems
- Find kindred spirits and find ways to work with them (make the sharing of ideas a safe place)
- Build internal champions (at all levels in the organization)
- Have courage (if something in the company doesn't make sense say so)
- Never give up!
- Respect ideas (but this doesn't mean you have to agree with them)
- Create a vision and repeat it (over and over and over....)
- Keep focused on the goal. (Mission success)
- Have fun (and bring food)

Results

- Lockheed Technical Fellow; Northrop Grumman Fellow
- Agile/DevOps Lead roles in the company
- Continued to press forward despite the obstacles
- Agile/DevOps for Government committees/working groups
- Suzette – Doctorate on Leadership behaviors/Agile and Traditional Environ.
- Robin – Doctorate in progress

We welcome more discussions on....

- How can we continue to build the talent needed for continued growth and change in an affordable way?
- How do we instill the desire and enthusiasm for lifelong learning in large organizations across geographic locations?
- How do we continue to grow the adoption of these practices in large organizations in highly regulated environments?
- How can we continue to expand/mature/improve DevOps principles across embedded systems and hardware?

Summary

- Build psychological safety
- Train and coach at all levels
- Modify existing procedures



Work across the Organization



Cultivate Agile Teams and Culture

- Build the delivery pipeline
- Automate
- Think how the pipeline is different as we scale



Build the Infrastructure



Measure Progress and Success

- Build and mentor internal champions
- Provide variety of training to meet demand

- Measure results:
Employee sat, customer sat,
- Flow and cycle time – are you delivering
- Manage cost of change