

When I joined Dell Technologies in 2016, it was with a goal to improve my sales and marketing skills, becoming more comfortable with public speaking, and honing my ability to craft a message to convey complex concepts.

I was asked shortly after joining, if I could create a lecture based on integrating DevOps into more mature enterprise organizations, often with strong ITIL practices.

I created this talk track and deck, 100% from scratch, and have presented it over a hundred times. The session was one of the highest rated in multiple Dell Technology world events, in addition to countless customers and meetups, as well college courses I've taught.

This shared presentation is part of a larger experiment in visual resume and brand building, the full body of work can be found @: <http://bit.ly/MattSchneider-VisualCV>

The following slide has a video *walkthrough*, it's not the full professional presentation, but gives you a flavor of the talk track and process I used to create a custom session.

DevOps

ITIL

Principles

Practices





INSURANCE



HEALTH CARE



LIFE SCIENCES



AIRLINE



LOGISTICS



RETAIL



CONSUMER GOODS



REAL ESTATE



FINANCIAL SERVICES



INFORMATION TECHNOLOGY



CONSULTING



Matt Schneider  
Chief Architect | Office of Strategy & Technology  
Director | Enterprise Architecture

@md\_schneider  
 /in/mdschneider

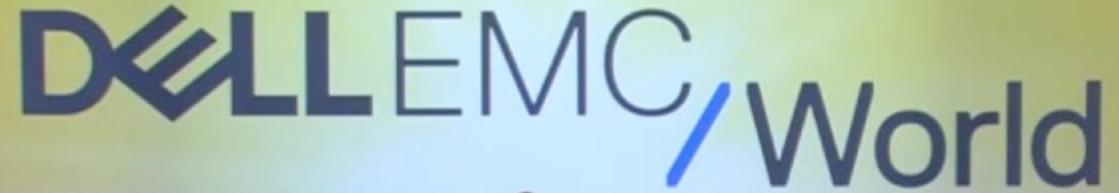
Forward



DELL EMC  
ELECT

vmware®

vEXPERT



When we stand at the center of the world's technology, we stand at the center of human progress... and there is no place I'd rather be."

-Michael Dell  
Chairman & CEO Dell Technologies



# DEVOPS & OR

HOW OUR  
**CAB**  
LEARNED  
TO STOP  
WORRYING  
**AND**  
TRUST  
**TDD**

THEY

This is Ops



Ops,  
management  
here, we  
need a new  
release.





Maybe in a  
few weeks,  
fill out an  
RFC.

# Developers



We have to fill out what?!?

# Change Advisory Board



I have no clue how to code, but I  
approve this release.





Have you tested this?

# We're calling executives.



Fine, let  
me check  
the  
schedule.



Maybe in  
a month?



I made  
some  
changes  
since it took  
a month.

# The release is packaged.

Release

Rollback

So is the rollback plan.



I'll deploy it myself, this code is  
very complicated.



# War Room

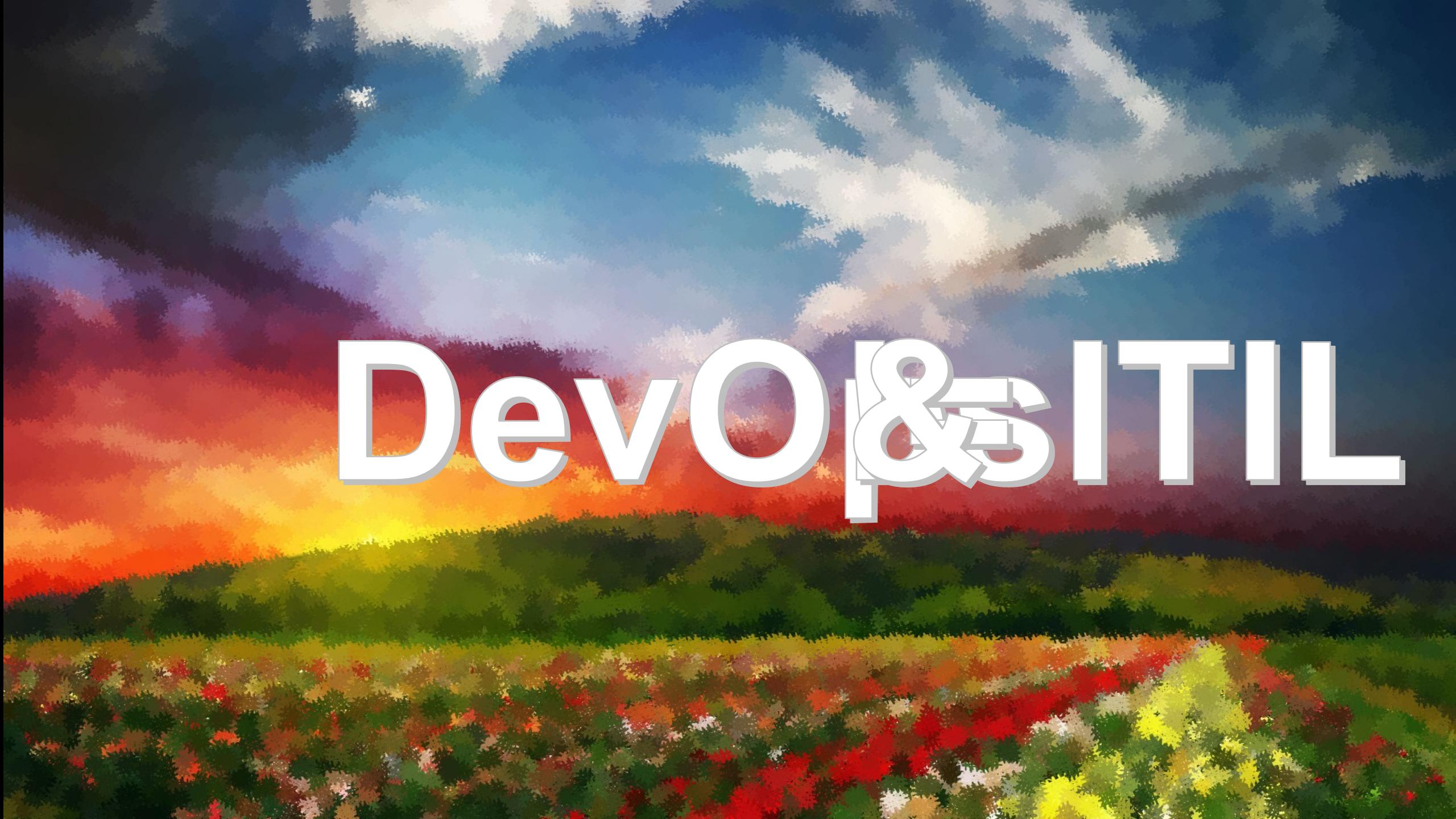


# Post Mortem



# IT Process Consultant





DevOps & ITIL





Skeptic



Teacher



Swayed



WAVES



Prepared

A photograph of two LEGO minifigures. A large, bright yellow minifigure stands in the foreground, facing slightly to the right. Its torso is yellow with a red rectangular sticker. A smaller, dark purple minifigure wearing a black cap and coat is perched on the yellow figure's shoulder, looking towards the camera. They are outdoors with blurred green and yellow foliage in the background.

If I have seen further  
than others, it is by  
standing upon the  
shoulders of giants.

-Isaac Newton



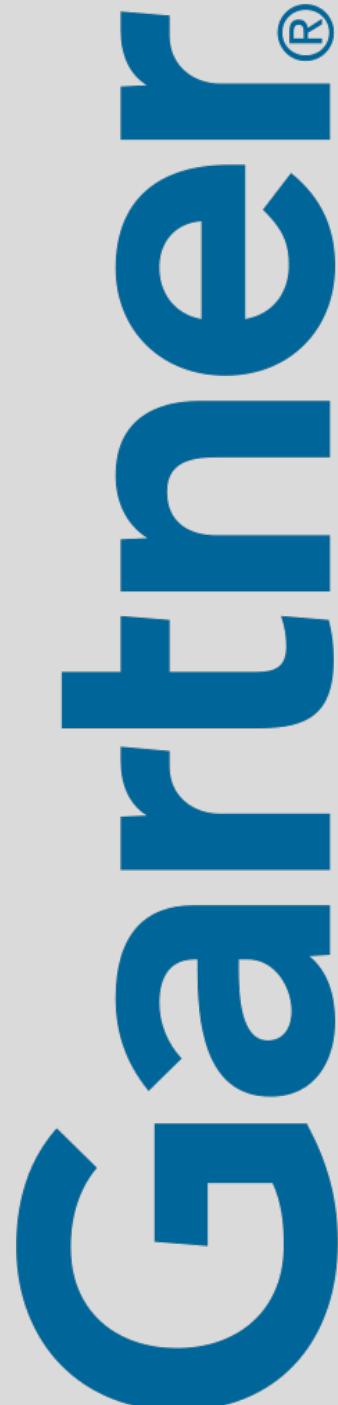
# DevOps



# WIKIPEDIA

The Free Encyclopedia

**DevOps** is a term used to refer to a set of practices that emphasize the collaboration and communication of both software developers and information technology (IT) professionals while automating the process of software delivery and infrastructure changes. It aims at establishing a culture and environment where building, testing, and releasing software can happen rapidly, frequently, and more reliably.



**DevOps** represents a change in IT culture, focusing on rapid IT service delivery through the adoption of agile, lean practices in the context of a system-oriented approach. DevOps emphasizes people (and culture), and seeks to improve collaboration between operations and development teams. DevOps implementations utilize technology — especially automation tools that can leverage an increasingly programmable and dynamic infrastructure from a life cycle perspective.

**Agile software development** describes a set of principles for software development under which requirements and solutions evolve through the collaborative effort of self-organizing cross-functional teams. Based off Agile Architecture, it advocates adaptive planning, evolutionary development, early delivery, and continuous improvement, and it encourages rapid and flexible response to change.



**WIKIPEDIA**  
The Free Encyclopedia

# **Manifesto for Agile Software Development**

We are uncovering better ways of developing software  
by doing it and helping others do it.

Through this work we have come to value:

**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.

# Manifesto for Agile Software Development

## Principles

Individuals and Interactions

Working Software

Customer Collaboration

Responding to Change

## Practices

processes and tools

comprehensive documentation

contract negotiation

following a plan

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

## Principles

Individuals and Interactions

Working Software

Customer Collaboration

Responding to Change

# Agile

## Practices

processes and tools

comprehensive documentation

# KANBAN

contract negotiation

following a plan

Principles

Practice

DevOps

ITIL

# DevOps

# ITIL

## Principles

Culture

Automation

Lean

Measurement

Sharing

## Practices

organizational structure

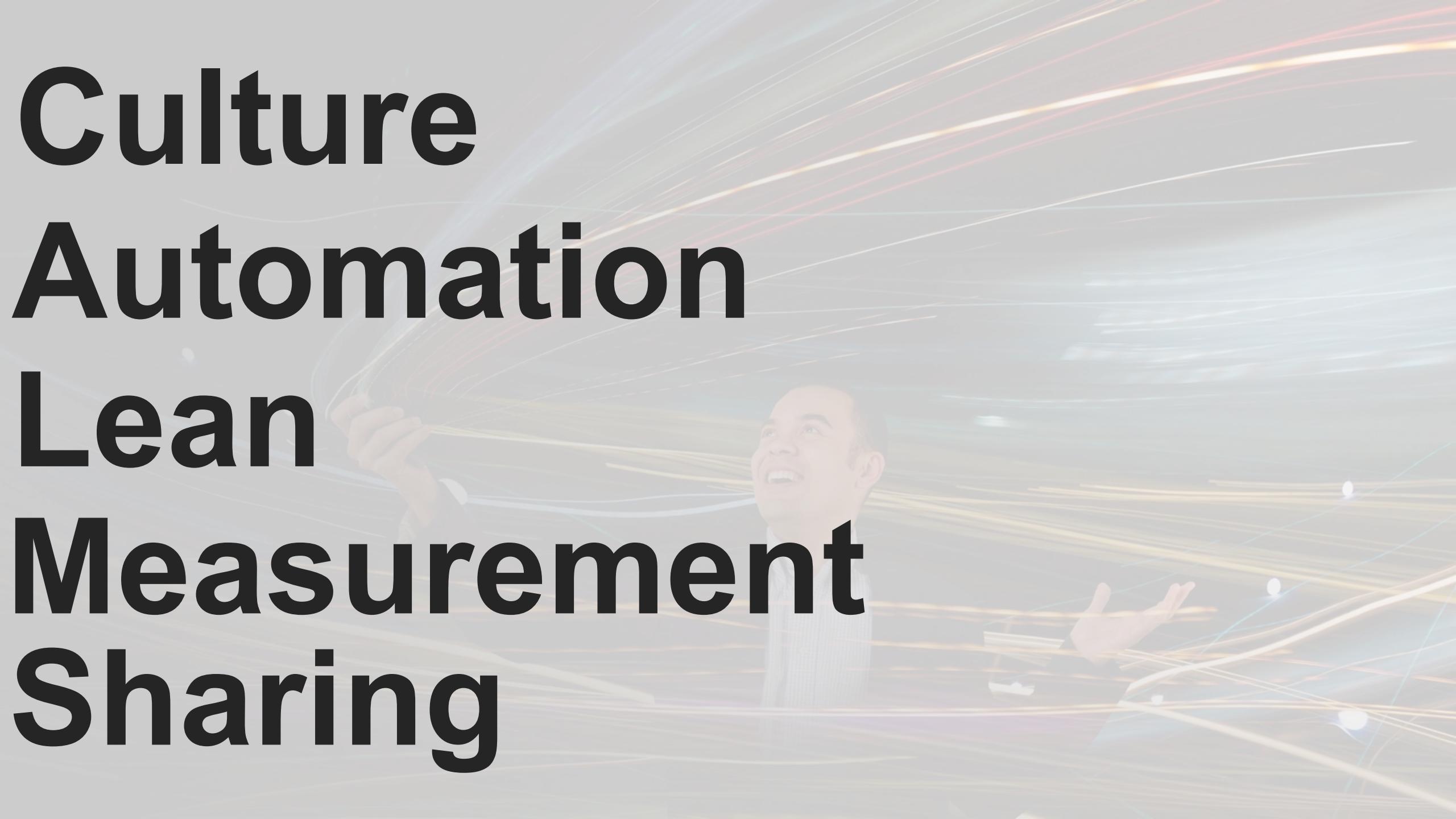
process and procedures

policies and approvals

SLAs and SLOs

detailed documentation

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

The background features a man with his arms raised in a joyful pose, surrounded by a burst of colorful light rays in shades of yellow, orange, and red. This central image is overlaid on a grid of thin, light-colored lines that radiate from the center, creating a sense of motion and connectivity.

**Culture  
Automation  
Lean  
Measurement  
Sharing**



# Culture

## **Generative** *Performance-Oriented*

High Cooperation

Messengers Trained

Risks are Shared

Bridging Encouraged

Failure Leads to Inquiry

Novelty Implemented

## **Bureaucratic** *Rule-oriented*

Modest Cooperation

Messengers Neglected

Narrow Responsibilities

Bridging Tolerated

Failure Leads to Justice

Novelty Leads to Problems

## **Pathological** *Power-oriented*

Low Cooperation

Messengers Shot

Responsibility Shirked

Bridging Discouraged

Failure Leads to Scapegoating

Novelty Crushed

Typology of Organizational Culture – Ron Westrum, 1994

- <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1765804/pdf/v013p0ii22.pdf>

# ITIL Management Practices

## Strategy

Service Portfolio  
Financial  
Demand  
Business Relationship

## Design

Service Catalog  
Service Level  
Capacity  
Service Continuity  
Security  
Supplier

## Transition

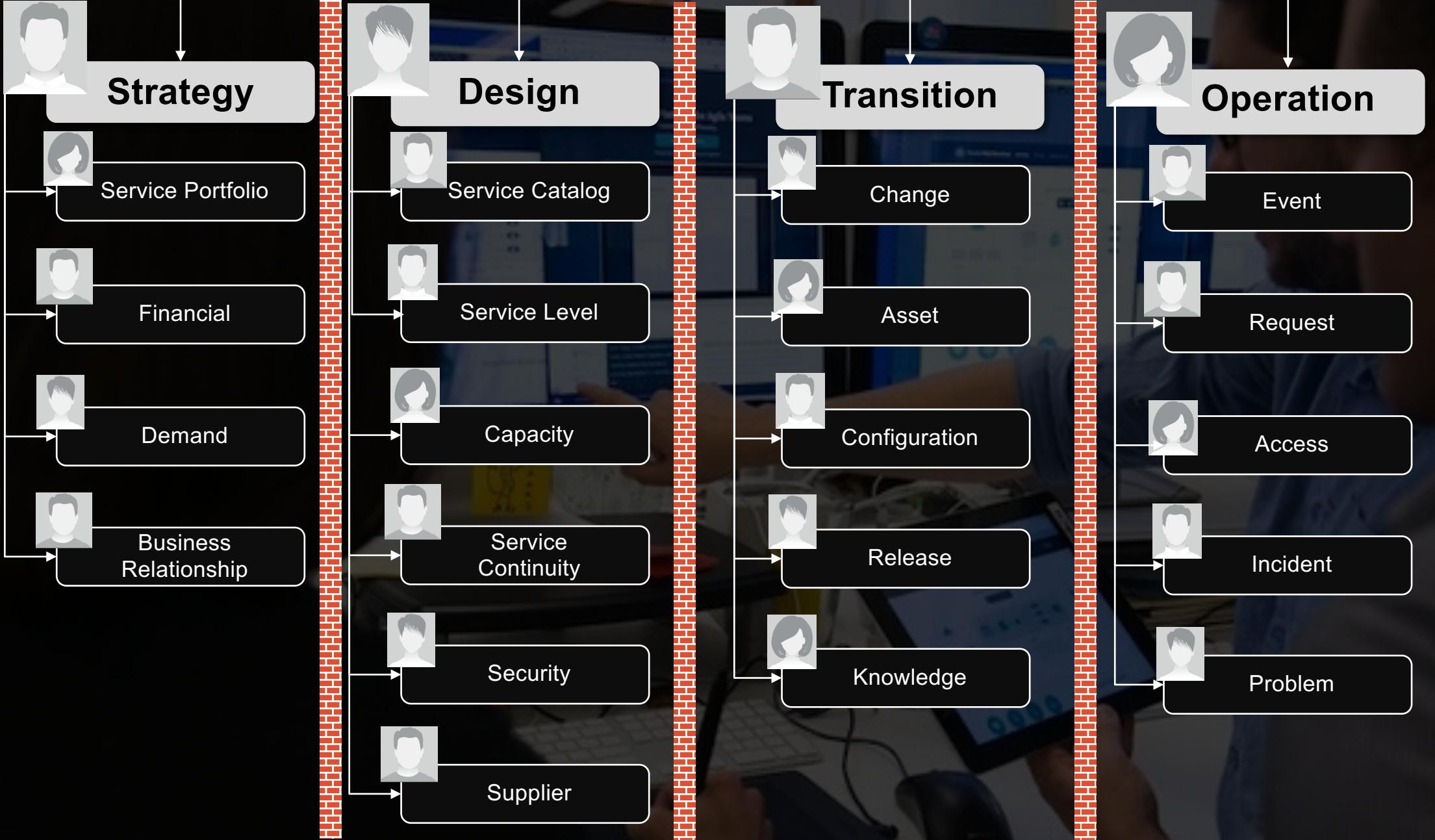
Change  
Asset  
Configuration  
Release  
Knowledge

## Operation

Event  
Request  
Access  
Incident  
Problem

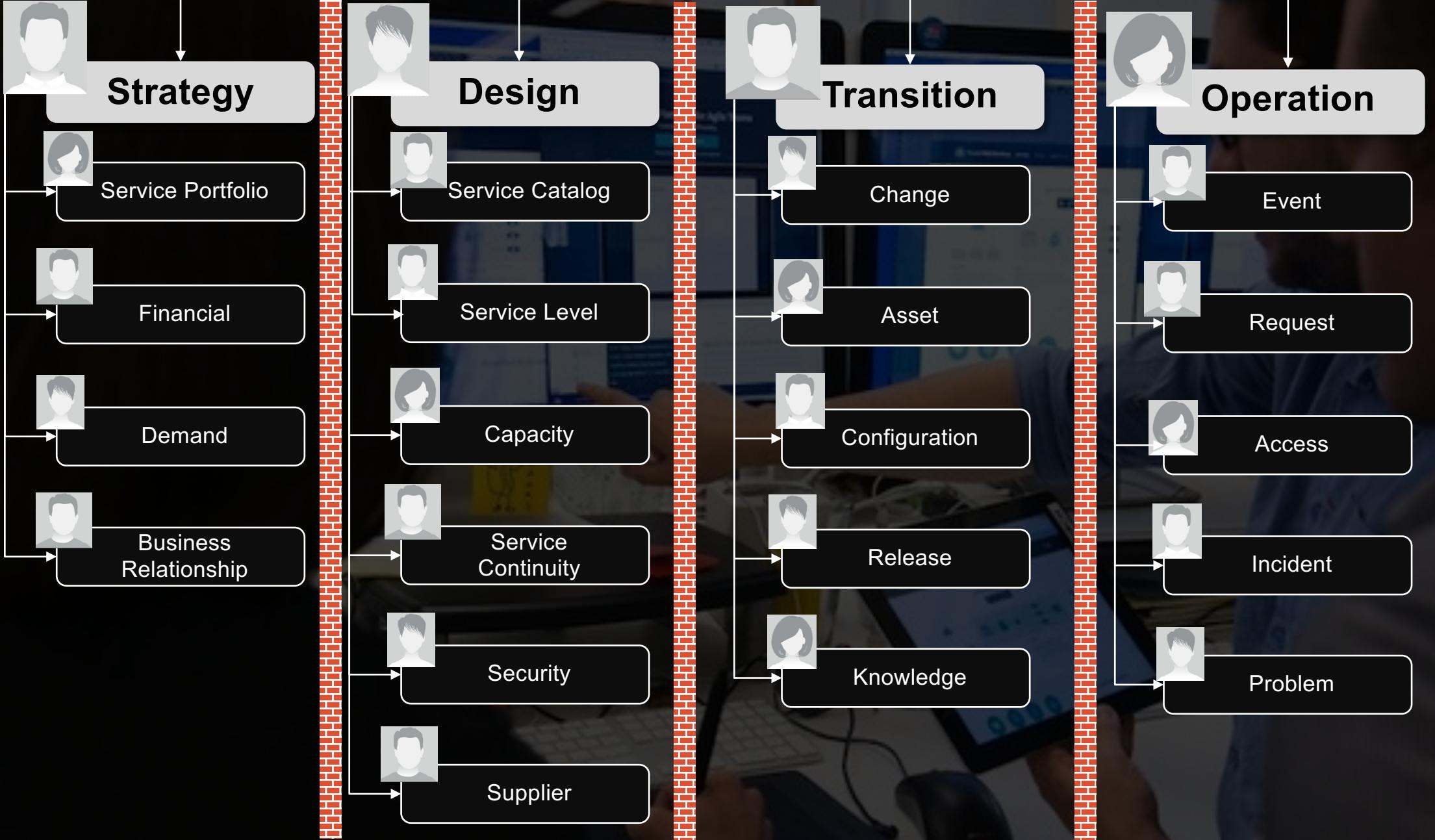
## Continual Improvement

Identify  
Define  
Gather  
Process  
Analyze  
Present  
Implement





Any organization that designs a system will produce a design whose **structure is a copy of the organization's communication structure**



## Strategy

Service Portfolio  
Financial  
Demand  
Business Relationship

## Design

Service Catalog  
Service Level  
Capacity  
Service Continuity  
Security  
Supplier

## Transition

Change  
Asset  
Configuration  
Release  
Knowledge

## Operation

Event  
Request  
Access  
Incident  
Problem

## Continual Improvement

Identify  
Define  
Gather  
Process  
Analyze  
Present  
Implement

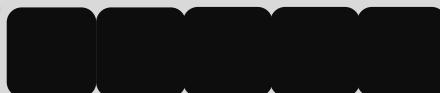
# Skills on the Product Lifecycle



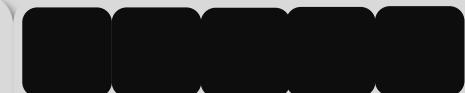
Strategy



Design

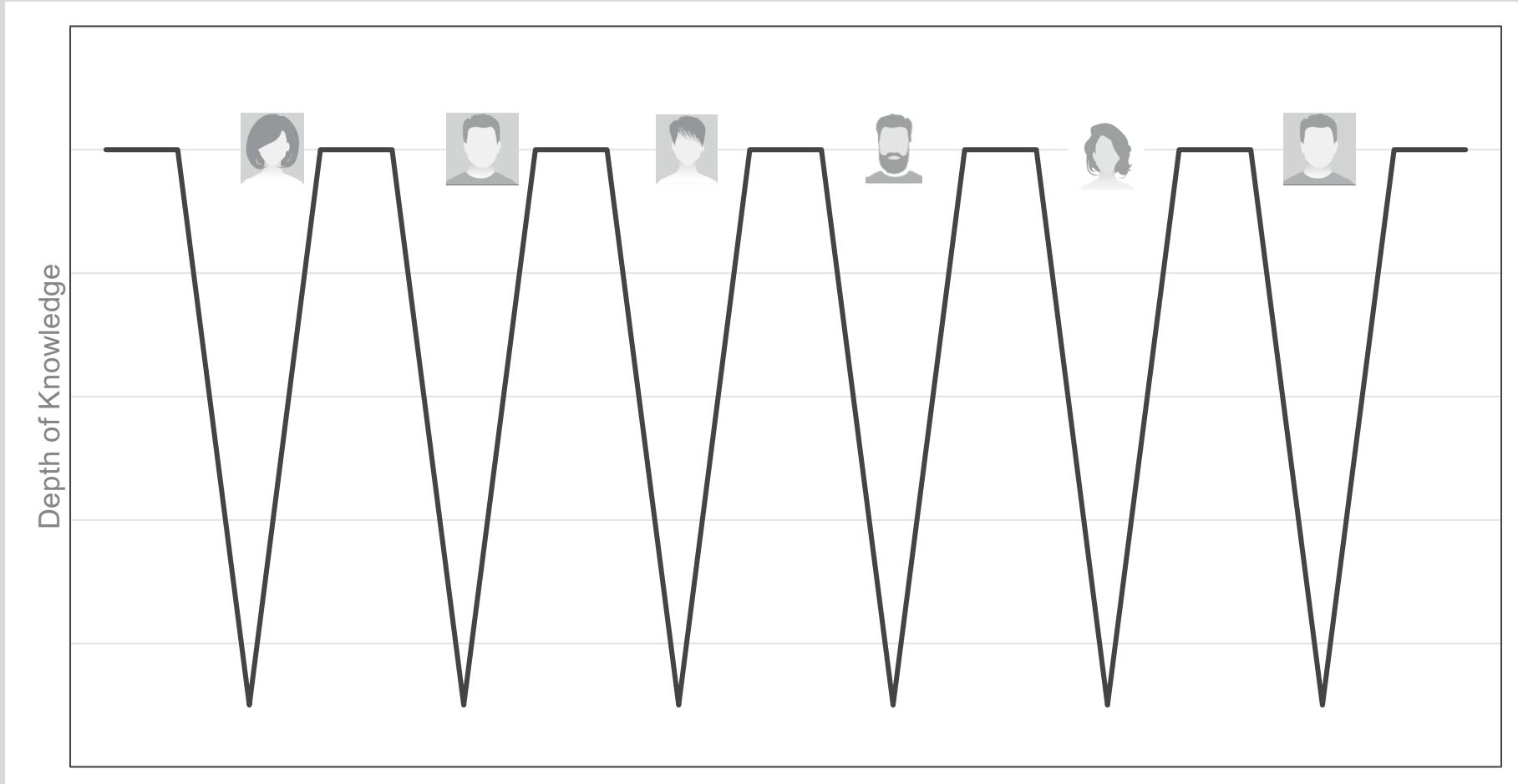


Transition



Operation

# Skills on the Product Lifecycle



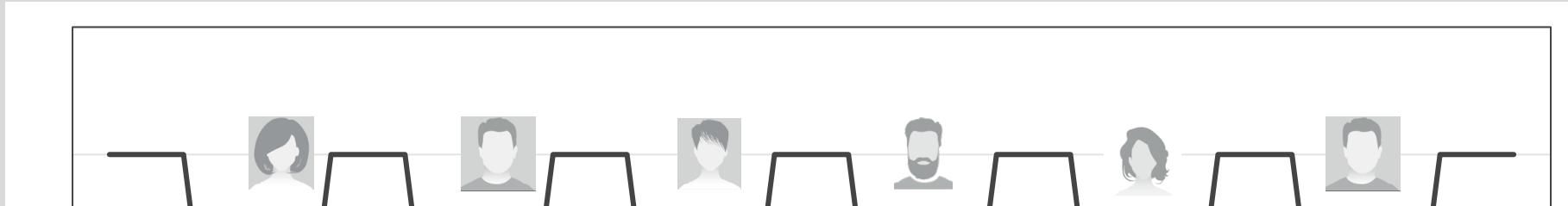
Strategy

Design

Transition

Operation

# Skills on the Product Lifecycle



Knowledge

Depth

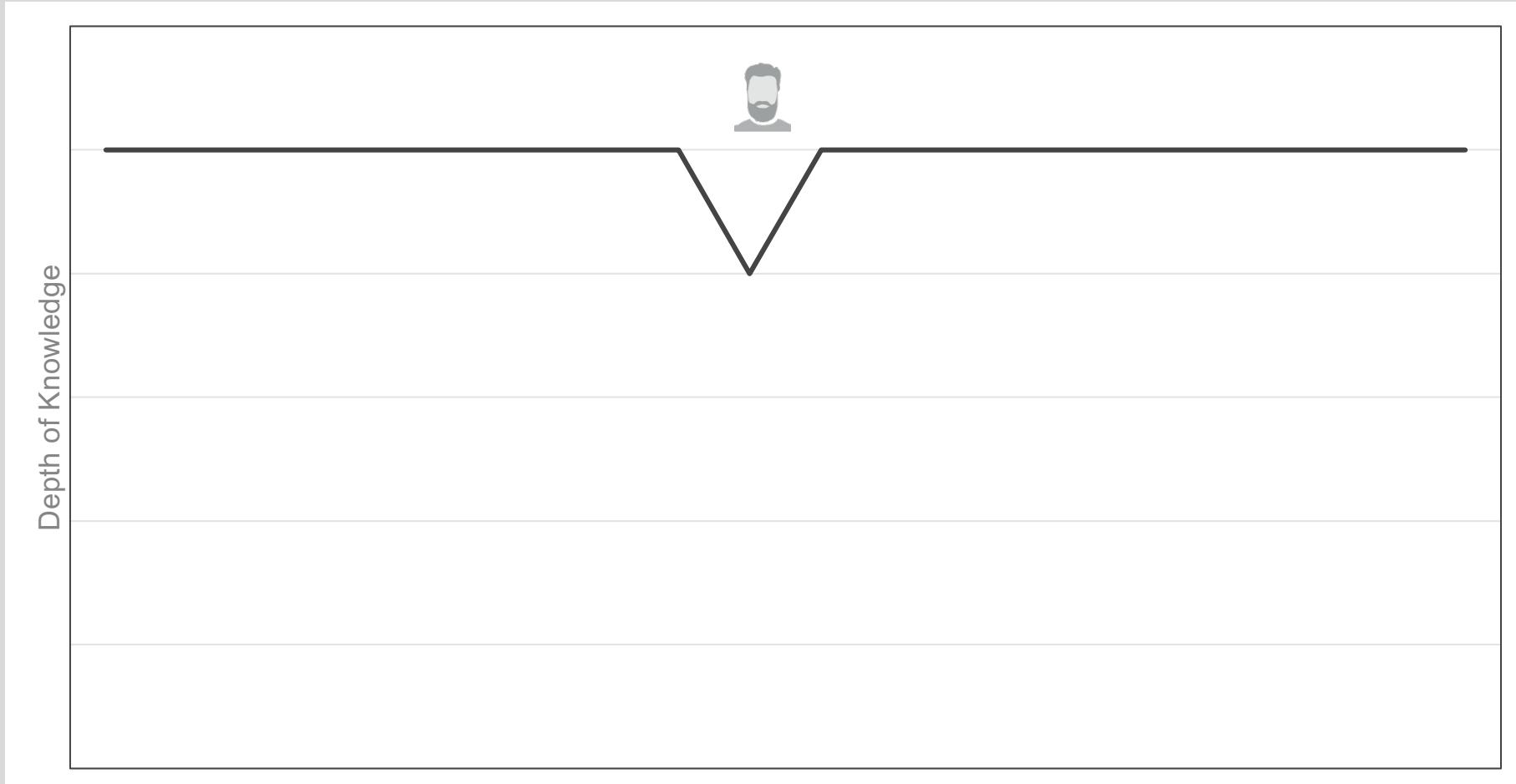
Strategy

Design

Transition

Operation

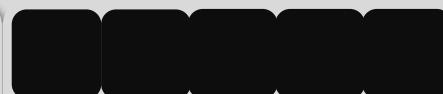
# Skills on the Product Lifecycle



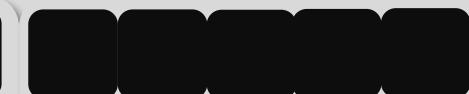
Strategy



Design

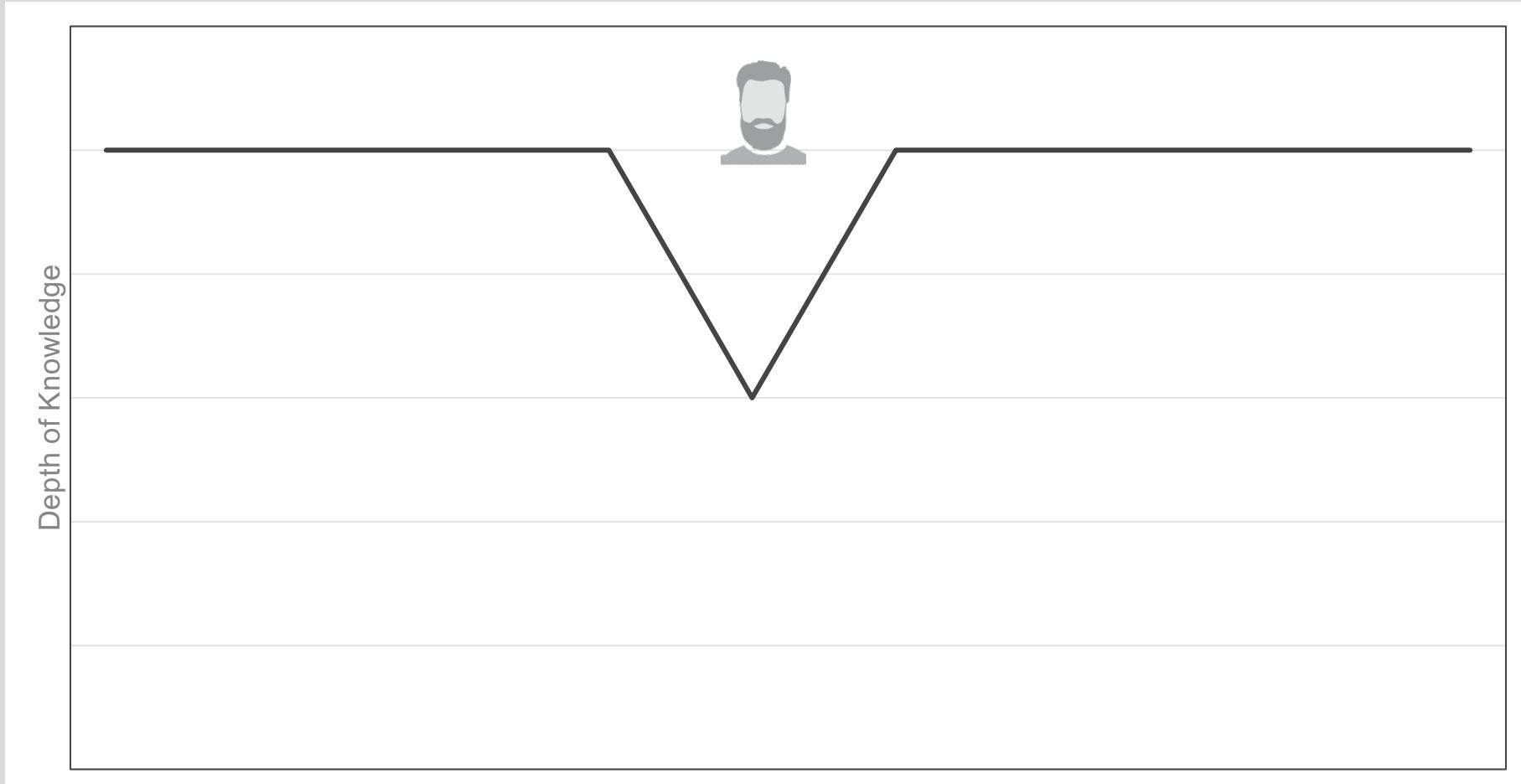


Transition



Operation

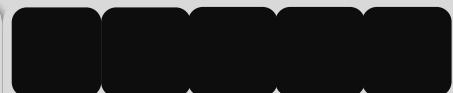
# Skills on the Product Lifecycle



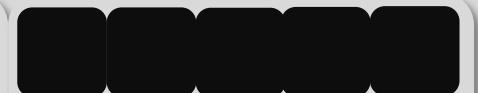
Strategy



Design

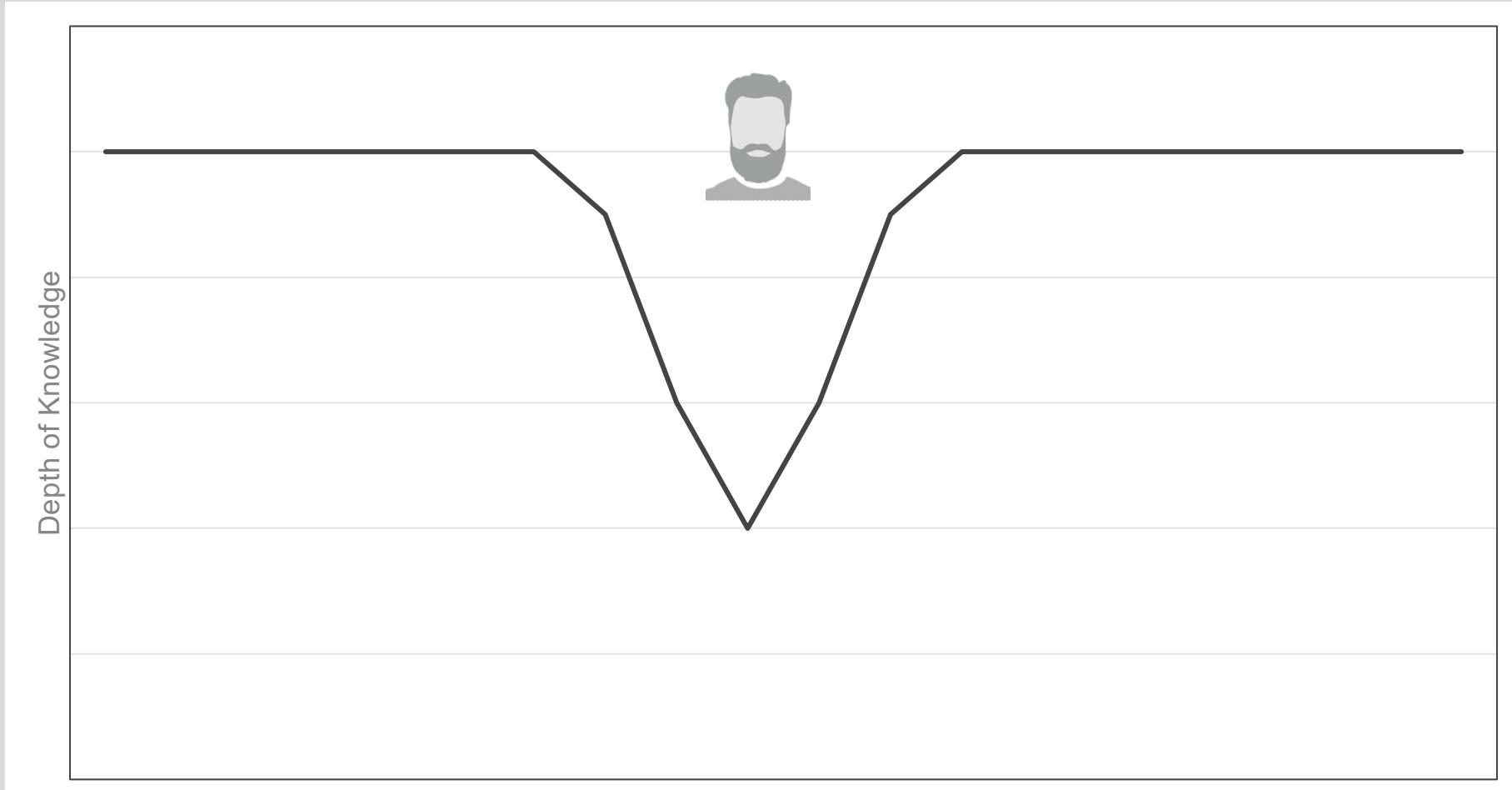


Transition



Operation

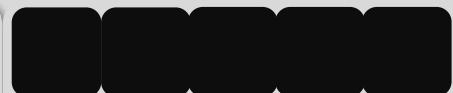
# Skills on the Product Lifecycle



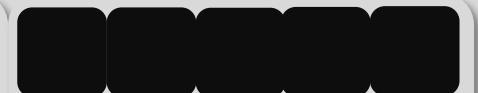
Strategy



Design

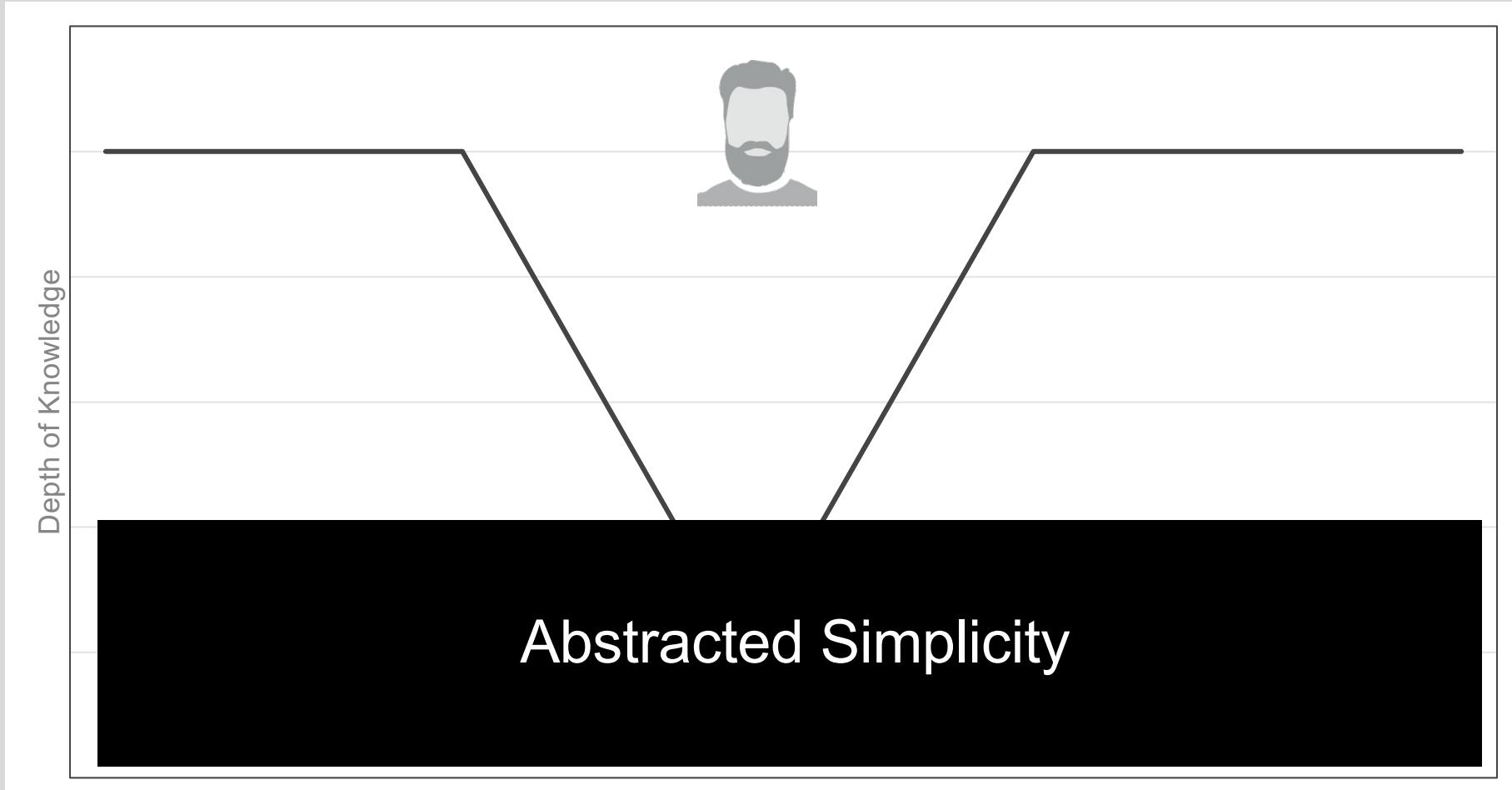


Transition



Operation

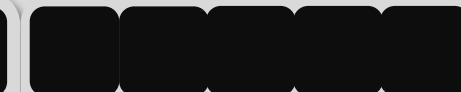
# Skills on the Product Lifecycle



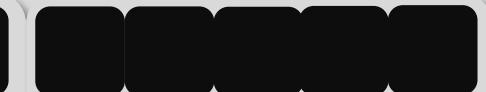
Strategy



Design



Transition



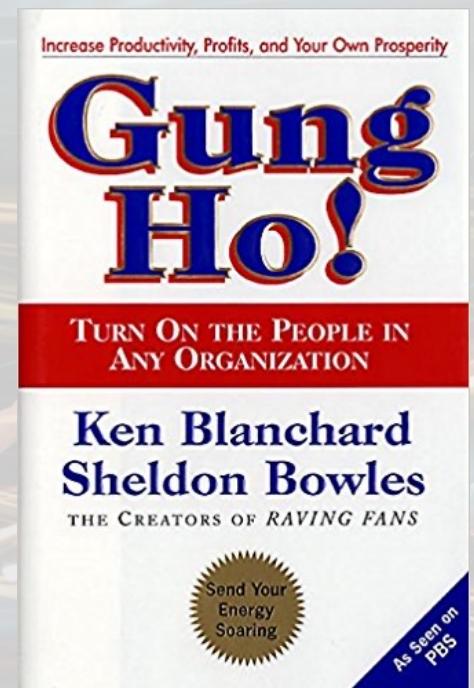
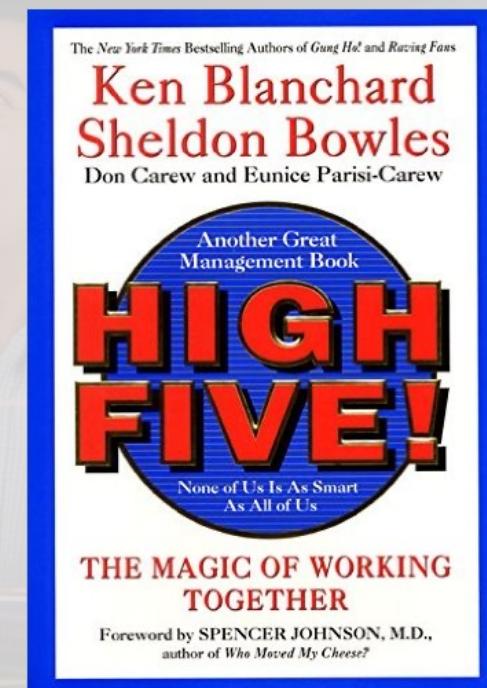
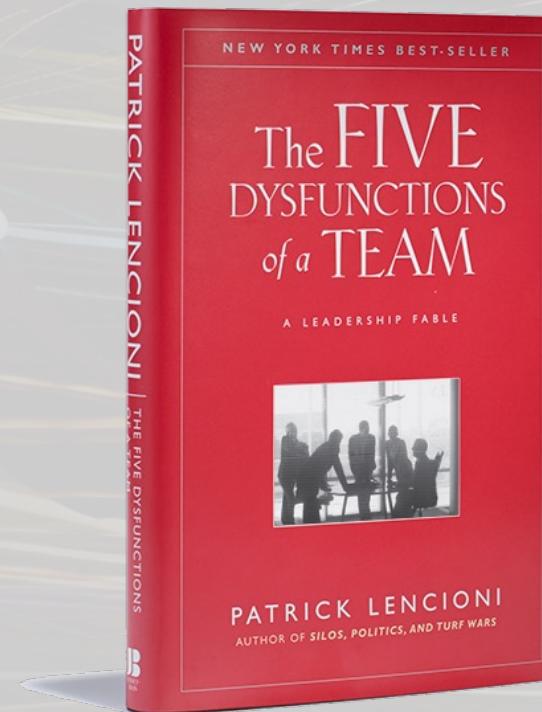
Operation

# Culture

A generative culture can exist with ITIL, the framework doesn't say to be bureaucratic. Identify where process pain-point are due to culture and aren't truly ITIL driven. Use the Principles of Agile and DevOps to guide how you implement Practices Speak openly about where you and your team can improve the culture. . Learn your surrounding skill sets and reach out to those who practice them.

Read:

Start a book club?

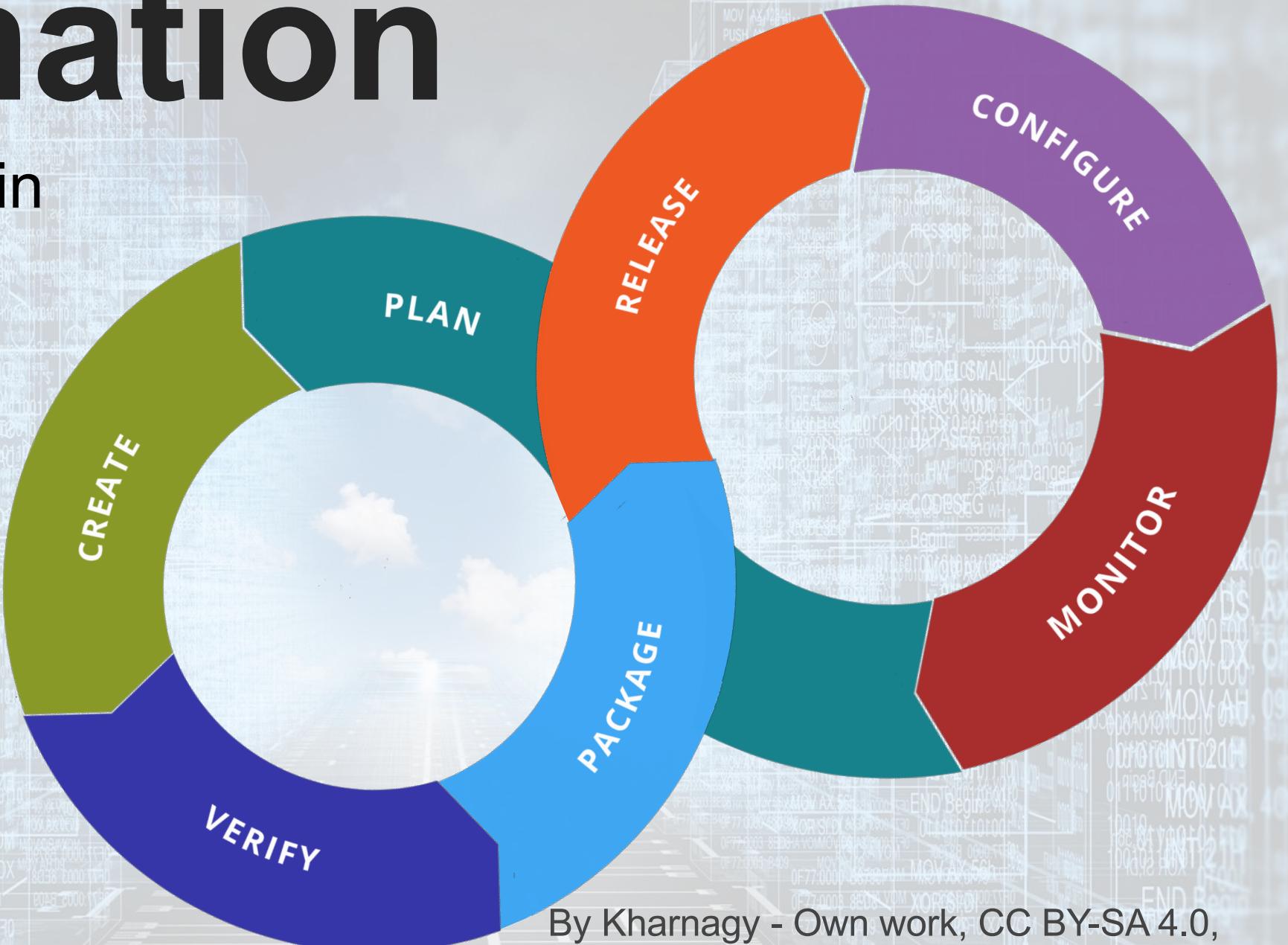


# Automation

Automate all the things.

# Automation

## DevOps Tool Chain



By Kharnagy - Own work, CC BY-SA 4.0,  
<https://commons.wikimedia.org/w/index.php?curid=51215412>

# Automation

Automate deployment to production

RELEASE



*all  
changes,*

Wikipedia – Change Management (ITSM)  
• [https://en.wikipedia.org/wiki/Change\\_management\\_\(ITSM\)](https://en.wikipedia.org/wiki/Change_management_(ITSM))

*ensure that standardized methods  
are used*

*minimize the impact of change-  
related incidents upon service quality,  
improve the day-to-day operations of  
the organization.*

Wikipedia – Change Management (ITSM)  
• [https://en.wikipedia.org/wiki/Change\\_management\\_\(ITSM\)](https://en.wikipedia.org/wiki/Change_management_(ITSM))



# Types of Changes

Normal

Standard

Emergency

Best Practices Insights - Focus On: ITIL Service Transition  
Published January 2016 by bmc – <https://www.bmc.com>

# Normal

# Standard

changes should follow the change management process from the beginning of the process and include all activities. You should assume that the impact of these changes to the business is significant until you determine that they are low impact. If the changes are low impact or insignificant, you can follow the standard change procedures. Then prioritize, authorize, and schedule the changes. Change management coordinates the change implementation, while release and deployment execute the change.

# Emergency

**Standard** changes are often preauthorized. They represent low-risk, common activities with a known outcome. For example, these can be services entered into the service catalog. The user community can select from the request fulfillment service to order cloud services, new users, modifications to applications, and access rights, for example. Standard changes can also be routine, low-impact IT changes.

# **Normal** **Emergency**

Best Practices Insights - Focus On: ITIL Service Transition  
Published January 2016 by  **bmc** – <https://www.bmc.com>



# Automation

Automate deployment to production

RELEASE

# Automation

Automate linking code to user stories (*traceability*)

Automate code coverage & security at check-in

Automate the build including build failure alerts

CREATE

# Automation

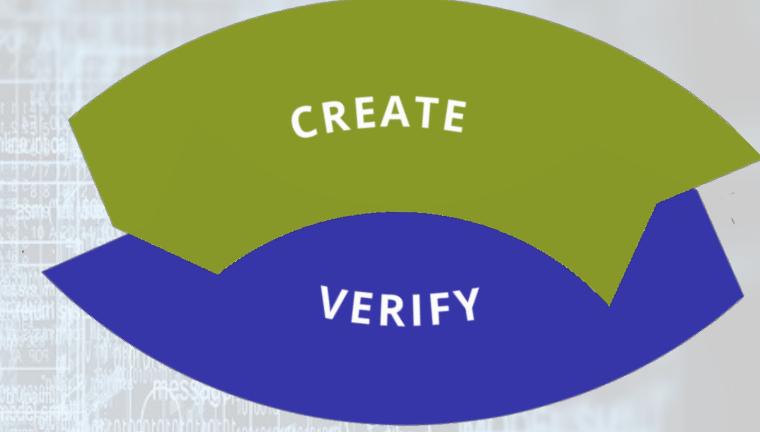
Automate linking code to user stories (*traceability*)

Automate code coverage & security at check-in

Automate the build including build failure alerts

Automate deploying to production-like test beds

Automate testing (*acceptance, regression, performance, etc*)



# Automation

Automate linking code to user stories (*traceability*)

Automate code coverage & security at check-in

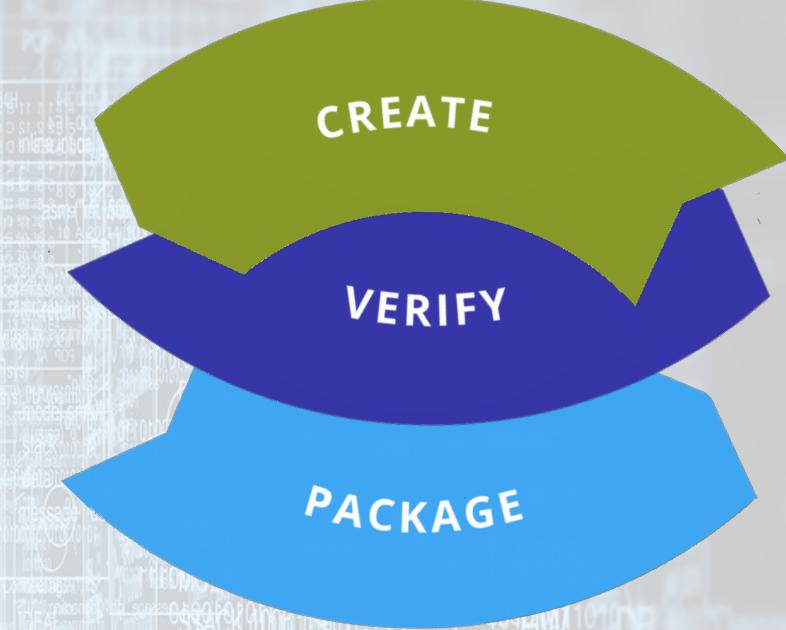
Automate the build including build failure alerts

Automate deploying to production-like test beds

Automate testing (*acceptance, regression, performance, etc*)

Automate populating request for change (*normal*)

Automate staging for deployment & rollback



# Automation

Automate linking code to user stories (*traceability*)

Automate code coverage & security at check-in

Automate the build including build failure alerts

Automate deploying to production-like test beds

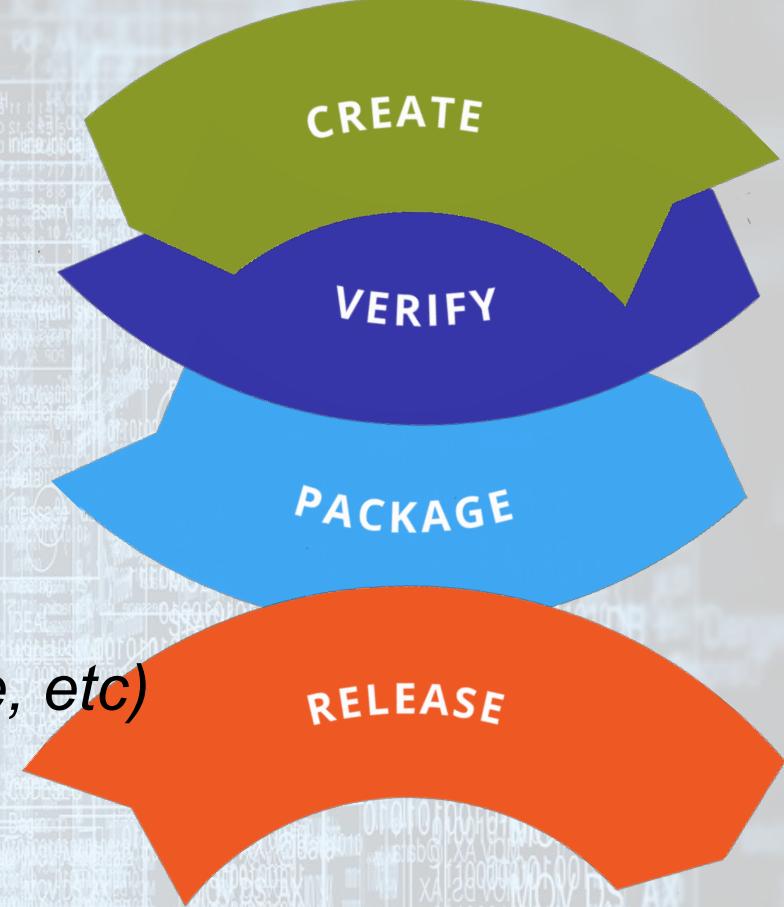
Automate testing (*acceptance, regression, performance, etc*)

Automate populating request for change (*normal*)

Automate staging for deployment & rollback

Automate logging change record

Automate deployment (*after approval*)



# Automation

Automate linking code to user stories (*traceability*)

Automate code coverage & security at check-in

Automate the build including build failure alerts

Automate deploying to production-like test beds

Automate testing (*acceptance, regression, performance, etc*)

Automate populating request for change (*normal*)

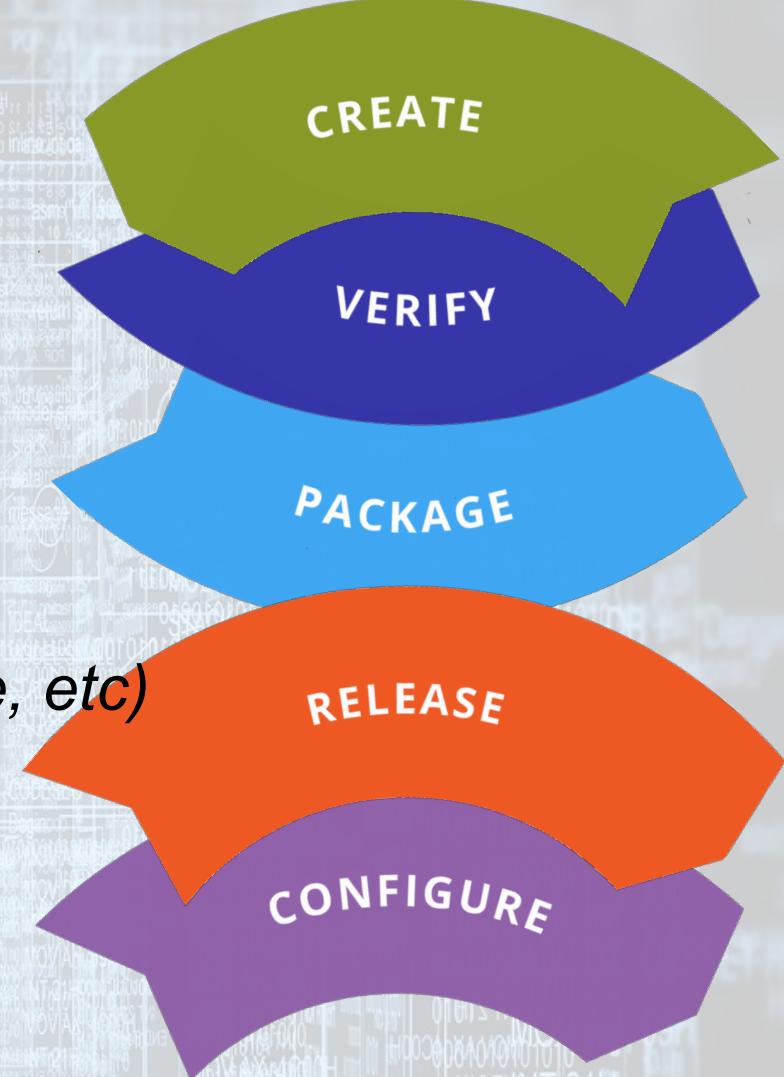
Automate staging for deployment & rollback

Automate logging change record

Automate deployment (*after approval*)

Automate updating CMDB

Automate desired/future state of environment



# Automation

Automate linking code to user stories (*traceability*)

Automate code coverage & security at check-in

Automate the build including build failure alerts

Automate deploying to production-like test beds

Automate testing (*acceptance, regression, performance, etc*)

Automate populating request for change (*normal*)

Automate staging for deployment & rollback

Automate logging change record

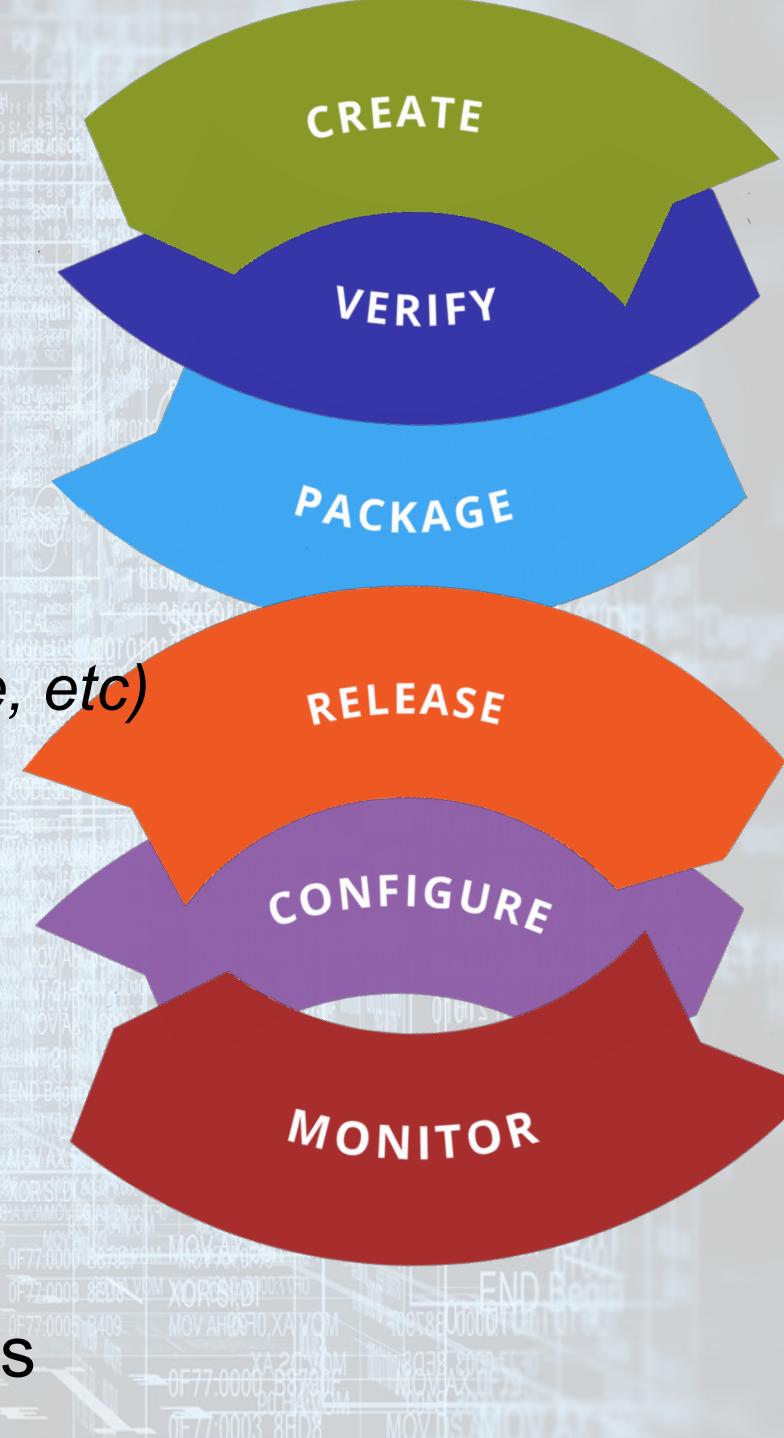
Automate deployment (*after approval*)

Automate updating CMDB

Automate desired/future state of environment

Automate application monitoring & logging

Automate linkage of Incident/Problem to User Stories



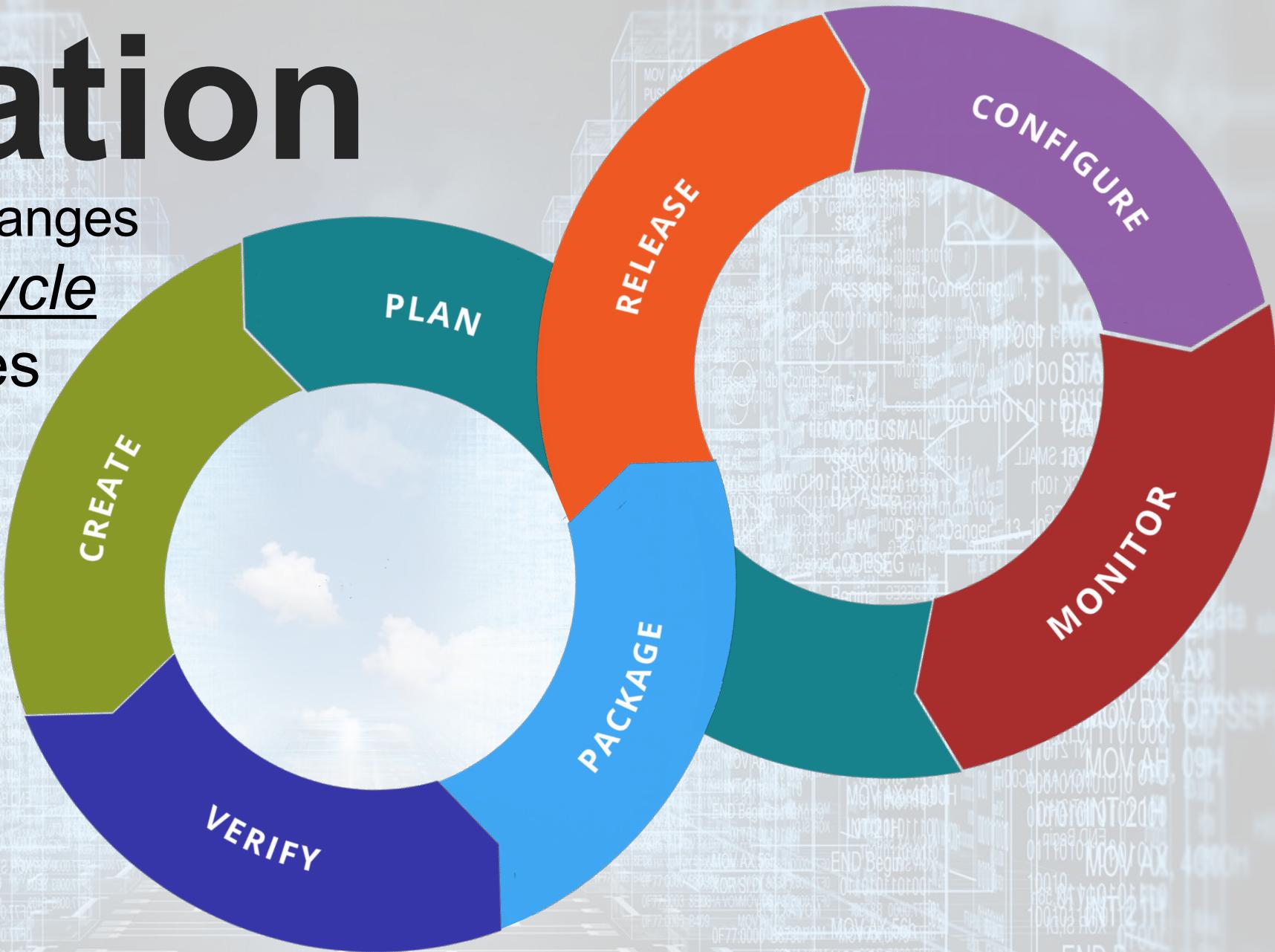
# Automation

Implement standard changes

Automate the Lifecycle

Everyone Automates

Everyone Benefits



By Kharnagy - Own work, CC BY-SA 4.0,  
<https://commons.wikimedia.org/w/index.php?curid=51215412>

# Automation

It is easier to automate a defined process – ITIL helps with that definition.

Leverage the Standard Change as encouragement to automate.

Automation belongs in all the practices, not just transition:

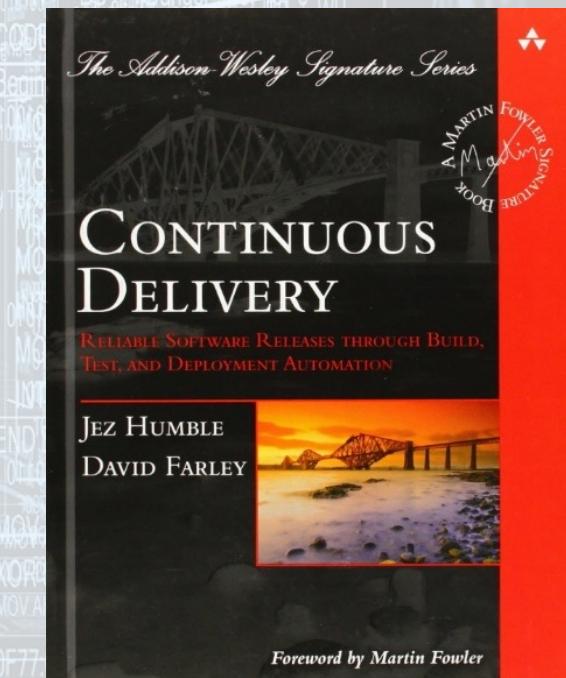
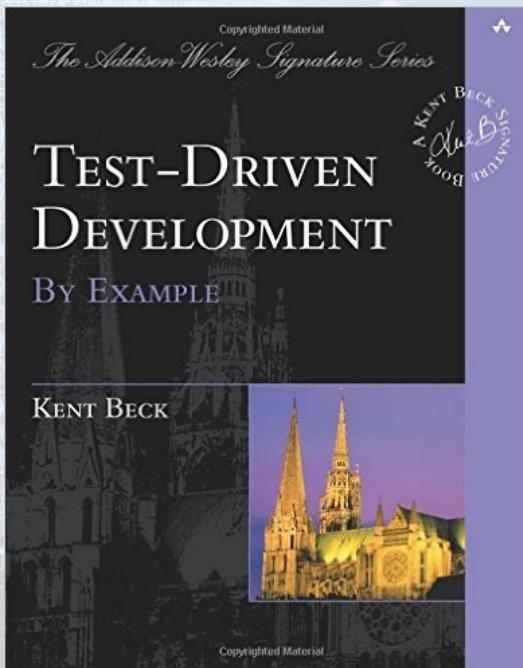
Service Catalog – Automated Provisioning

Self Service Incident Resolution

Configuration Management Tracking

Access Management Request Approval

Read:





# Lean

# Lean

**Lean software development (LSD)** is a translation of lean manufacturing and lean IT principles and practices to the software development domain. Adapted from the Toyota Production System, a pro-lean subculture is emerging from within the Agile community.

-Wikipedia

## Strategy

Service Portfolio  
Financial  
Demand  
Business Relationship

## Design

Service Catalog  
Service Level  
Capacity  
Service Continuity  
Security  
Supplier

## Transition

Change  
Asset  
Configuration  
Release  
Knowledge

## Operation

Event  
Request  
Access  
Incident  
Problem

## Continual Improvement

Identify  
Define  
Gather  
Process  
Analyze  
Present  
Implement

# Process Management

## Strategy

- Service Portfolio
- Financial
- Demand
- Business Relationship

## Design

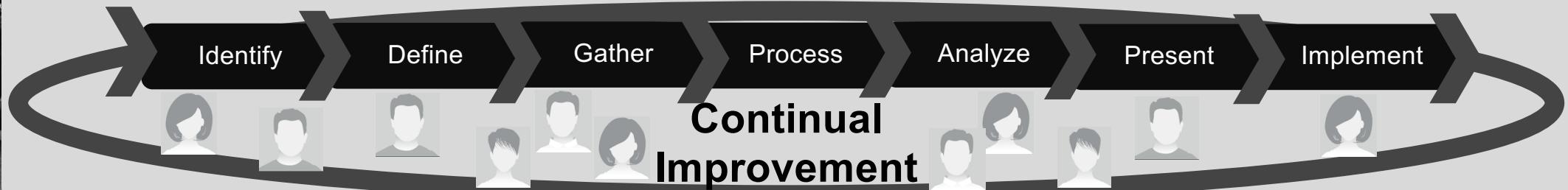
- Service Catalog
- Service Level
- Capacity
- Service Continuity
- Security
- Supplier

## Transition

- Change
- Asset
- Configuration
- Release
- Knowledge

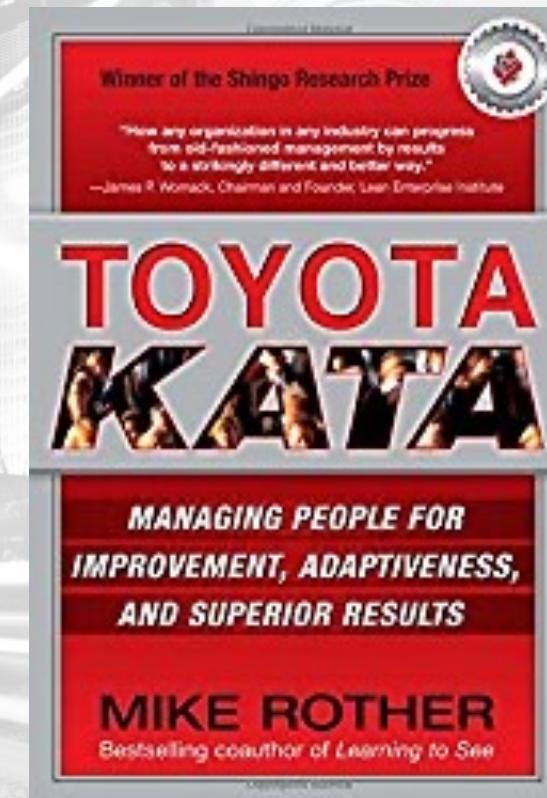
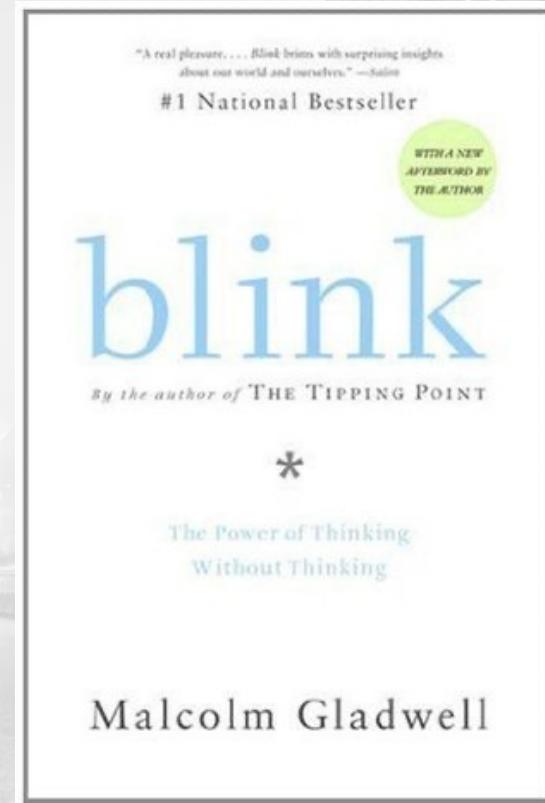
## Operation

- Event
- Request
- Access
- Incident
- Problem



# Lean

ITIL v3 incorporates continual improvement, which is the goal of lean.  
ITIL functions should be under continual improvement.  
If there is unnecessary waste in your ITIL processes, seek to remove it.



# Measurement

Configuration  
Management

Capacity  
Management

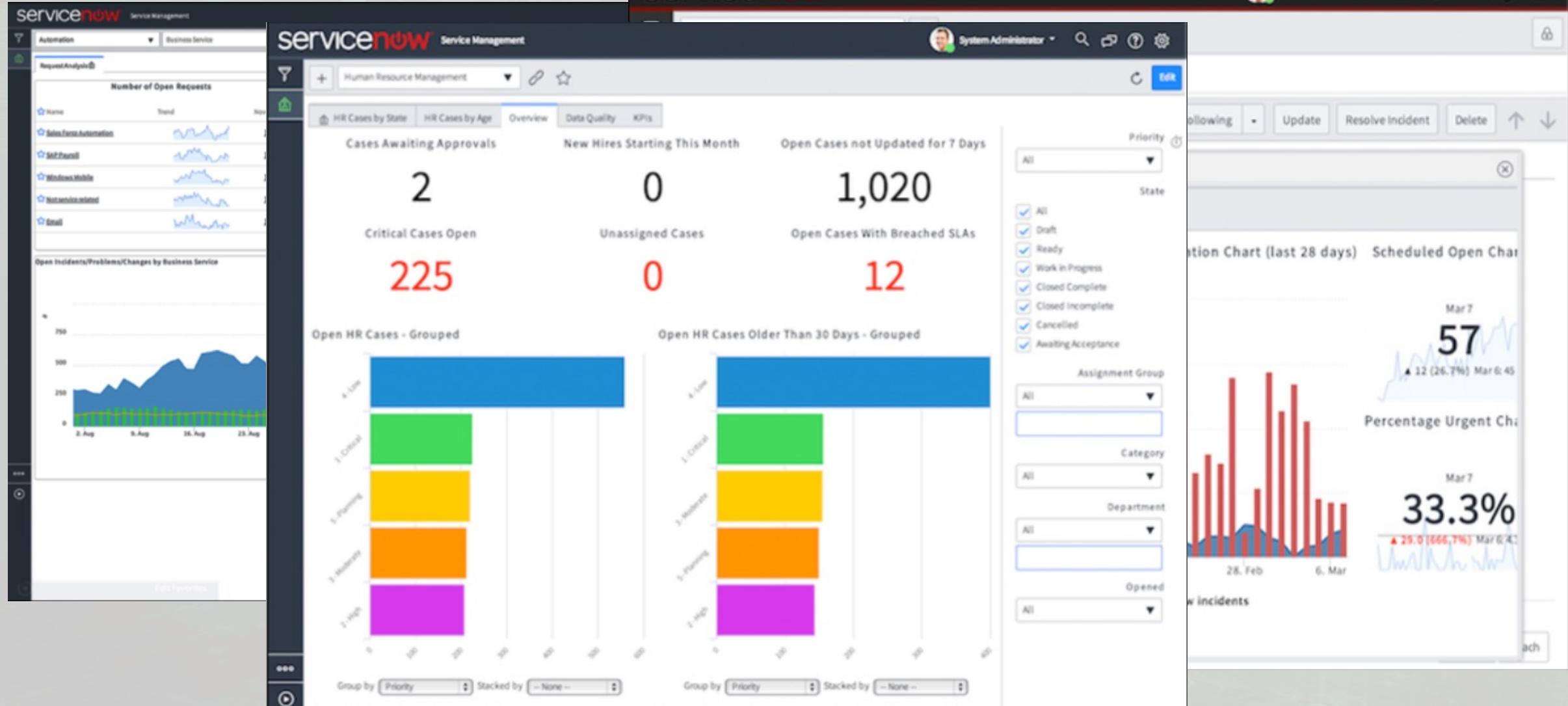
Problem  
Management

Asset  
Management

Incident  
Management

Service  
Continuity  
Management

Demand  
Management



## servicenow® Analytics

- <https://www.servicenow.com/products/performance-analytics.html>

# Measuring DevOps - Automation

Deployment Frequency

How often does your organization deploy code?

Lead time for changes

What is your lead time for changes?  
(code commit to production deploy)

Mean time to recover  
(MTTR)

How long does it generally take to restore service when an unplanned service incident occurs?

Change failure rate

What percentage of the changes either result in degraded service or subsequently require remediation.

Puppet State of the DevOps

- <https://puppet.com/resources/whitepaper/2016-state-of-devops-report>

	High Performers	Medium Performers	Low Performers
Deployment Frequency	On Demand (multiple deploys per day)	1 Week – 1 Month	1 Week – 6 Months
Lead time for changes	Less than one hour	Between one week and one month	1 Week – 6 Months
Mean time to recover (MTTR)	Less than one hour	Less than one day	Less than one day
Change failure rate	0-15%	31-45%	16-30%

## Puppet State of the DevOps

- <https://puppet.com/resources/whitepaper/2016-state-of-devops-report>

# Measuring DevOps - Culture

High Performers

2.2x

more likely to recommend organization

1.8x

more likely to recommend team

eNPS

Puppet State of the DevOps

- <https://puppet.com/resources/whitepaper/2016-state-of-devops-report>

# Measuring DevOps - Culture

2.5x

revenue growth – high engagement over low engagement

3x

stock performance of high trust environments  
over market index – measured 1997-2011

eNPS

The chemistry of enthusiasm

- <http://www.bain.com/publications/articles/the-chemistry-of-enthusiasm.aspx>

## Highly Engaged & Enabled Employees

Employee Performance

50%

Increase in employees above performance expectations.

Employee Retention

-54%

Reduction in employee turnover rates.

Customer Satisfaction

89%

Increase in customer satisfaction rates.

Financial Success

x4.5

Increase in revenue growth.

HayGroup – Giving everyone the chance to shine

- [http://www.haygroup.com/downloads/uae/giving\\_everyone\\_the\\_chance\\_to\\_shine\\_whitepaper\\_low\\_res\\_singles.pdf](http://www.haygroup.com/downloads/uae/giving_everyone_the_chance_to_shine_whitepaper_low_res_singles.pdf)

# Measurement

“You get what you measure”.

Both DevOps and ITIL have a strong measurement base.

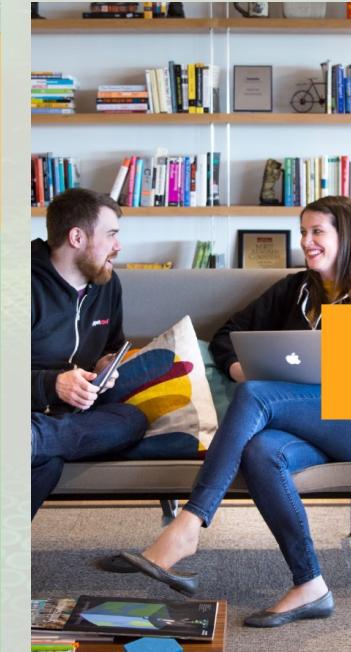
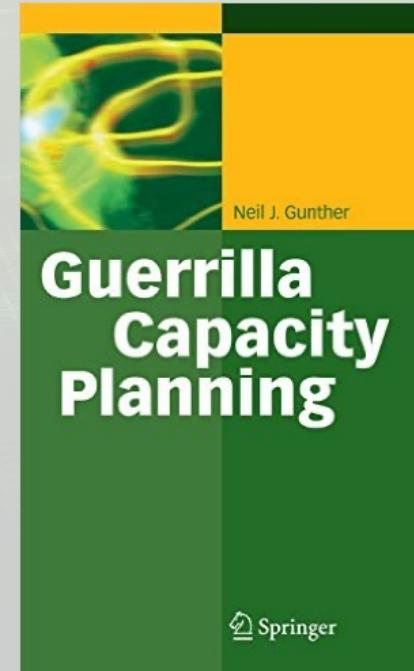
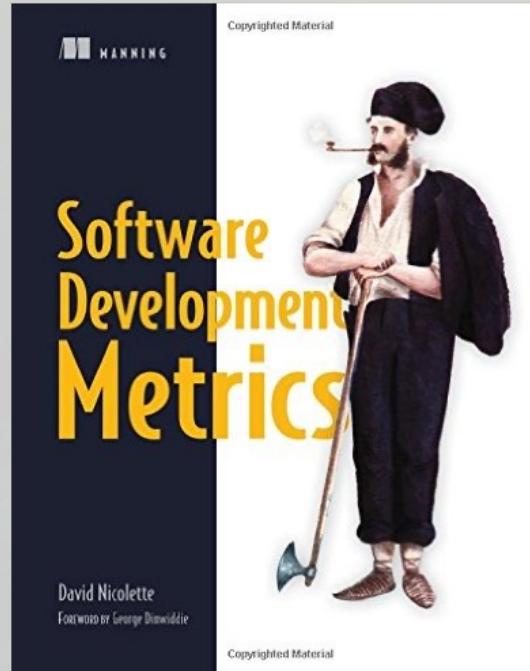
Remember to embody a generative culture when interpreting measurements.

Ensure you can measure your Purpose, Principles and Practices.

Merge your business, development and ITIL measurements into one view.

Find correlations between the measurements to understand team interaction.

Read:



# Sharing ITIL Knowledge Management

*“The process responsible for sharing perspectives, ideas, experience and information, and for ensuring that these are available in the right place and at the right time. The knowledge management process enables informed decisions, and improves efficiency by reducing the need to rediscover knowledge.”*

-Source: ITIL 2011 Glossary of Terms

Culture  
Automation  
Lean  
Measurement  
Sharing

C  
A  
L  
M  
S



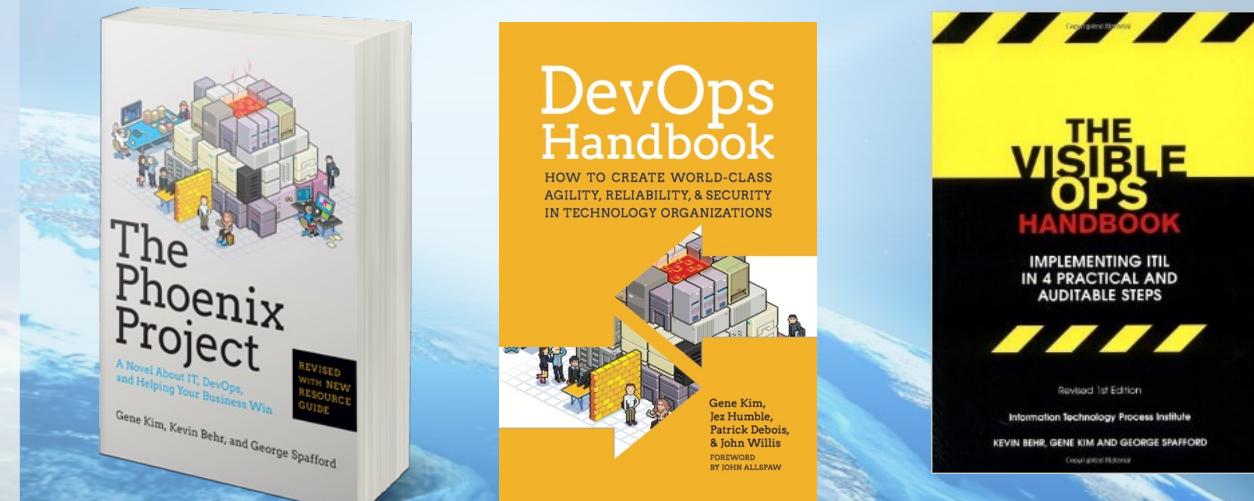
**DevOps**  
**CAMS**  
**ITIL**

A large, semi-transparent red heart is positioned in the center of the word "CAMS". The heart is filled with a detailed circuit board pattern, with various electronic components and labels visible.



“It is my firm belief that ITSM and the DevOps movement are not at odds. Quite to the contrary, they’re a perfect cultural match. As DevOps gains momentum I’m excited by what we can achieve using a winning combination of the two.”

-Gene Kim



From the article:  
Trust me: The DevOps Movement fits perfectly with ITSM

<http://www.theitsmreview.com/2014/03/trust-devops-movement-fits-perfectly-itsm/>



DELL EMC

Pivotal.

RSA

Secureworks®

virtustream®

vmware®

A landscape photograph of a train track that curves away from the viewer, creating a sense of depth and perspective. The sky is filled with warm, orange and yellow hues of a setting sun. The text "THE END" is overlaid in large, white, serif capital letters across the center of the image.

THE END

<https://bit.ly/DevOpsITIL>

Matt Schneider



@md\_schneider



/in/mdschneider