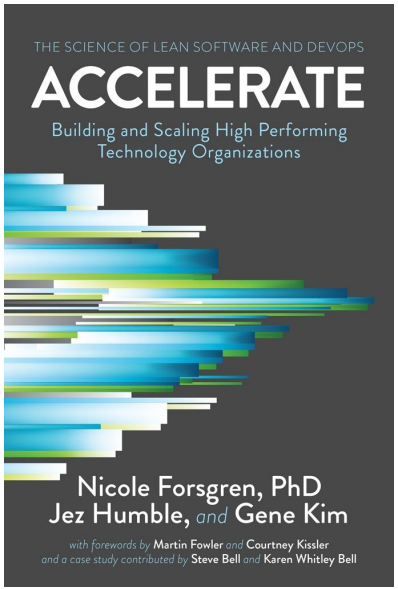
# Was bringt DevOps?

## Accelerate und “State of DevOps”




von Marc Nickert

# Version Control

- for all artifact:
  - Configuration
  - Test data
  - Infrastructure
  - Build artifacts (just not in Git)

# Continuous Integration

- Build automation
- Automatic Unit Testing

# Deployment Automation

- Configuration should be versioned like Code
- Any Configuration change should be tested like code
- If manual approval is needed than the approval should be the only manual step

# Trunk-Based Development

- Merge more often into Main
- No branche should last longer than a day

# Test Automation

- Continuous testing.
- Makes refactoring and high quality possible

# Test Data Management

- For every possible test, test data will be created and Version Controlled
- no need to test on Production

# Shifting Left on Security

- Security should not be done by an Expert Team after development
- DevOps Teams are responsible for Security
- Security is a Consulting Role

# Continuous Delivery/Empowered Teams

- Build quality in.
- Work in small batches.
- Computers perform repetitiv tasks; people solve problems.
- Relentlessly pursue continuous improvement.
- Everyone is responsible.

# Loosely Coupled Architecture

- Microservices
- Serverless
- Event Driven

# Monitoring/Proactive Notification

# Software Delivery Performance

- Lead Time
- Deployment Frequency
- Mean Time to Restore (MTTR)
- Change Fail Percentage

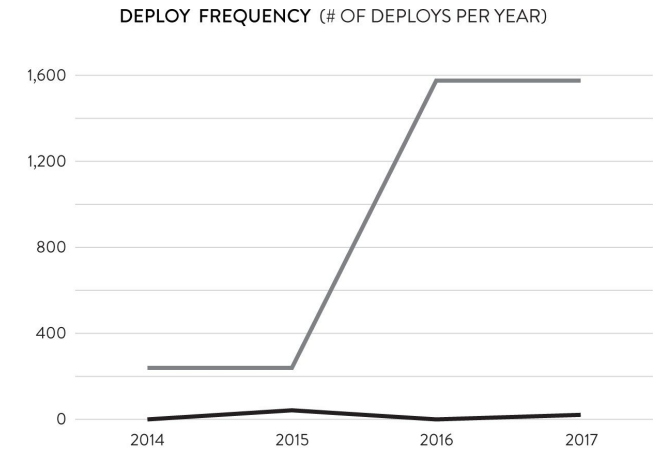
# Frequency of Delivery (Deployment to Production)

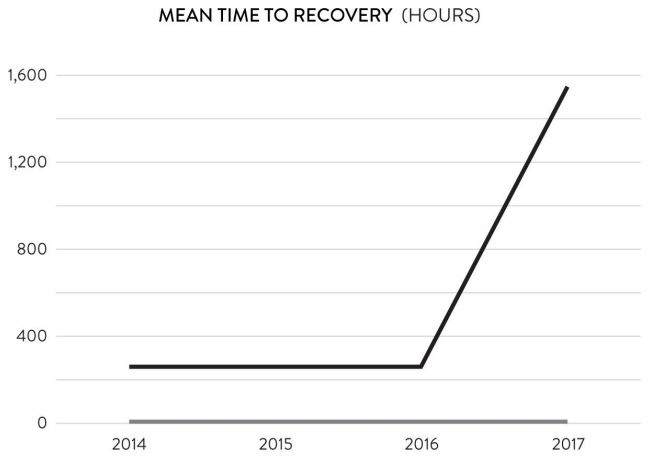
Options:

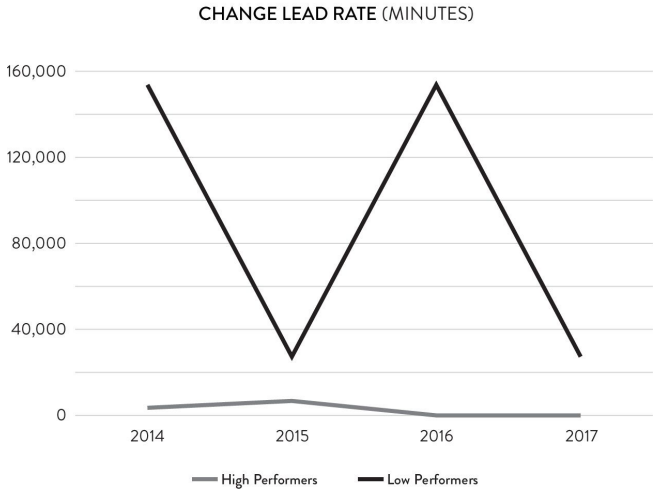
- on demand (multiple deploys per day)
- between once per hour and one per day
- between once per day and once per week
- between once per week and once per month
- between once per month and once every six months
- fewer than once every six months

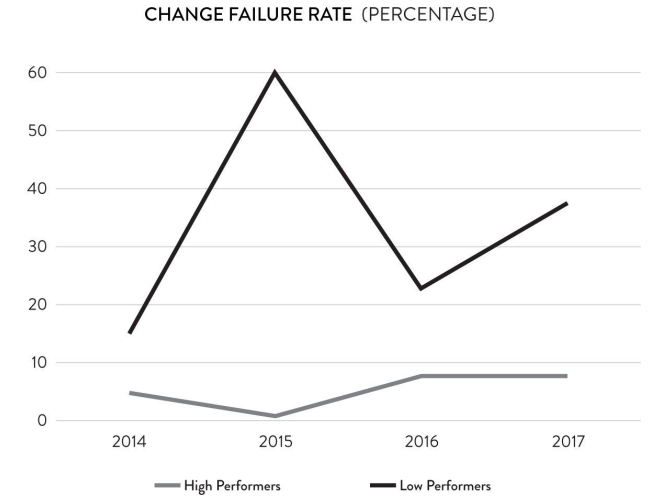
# Clustering

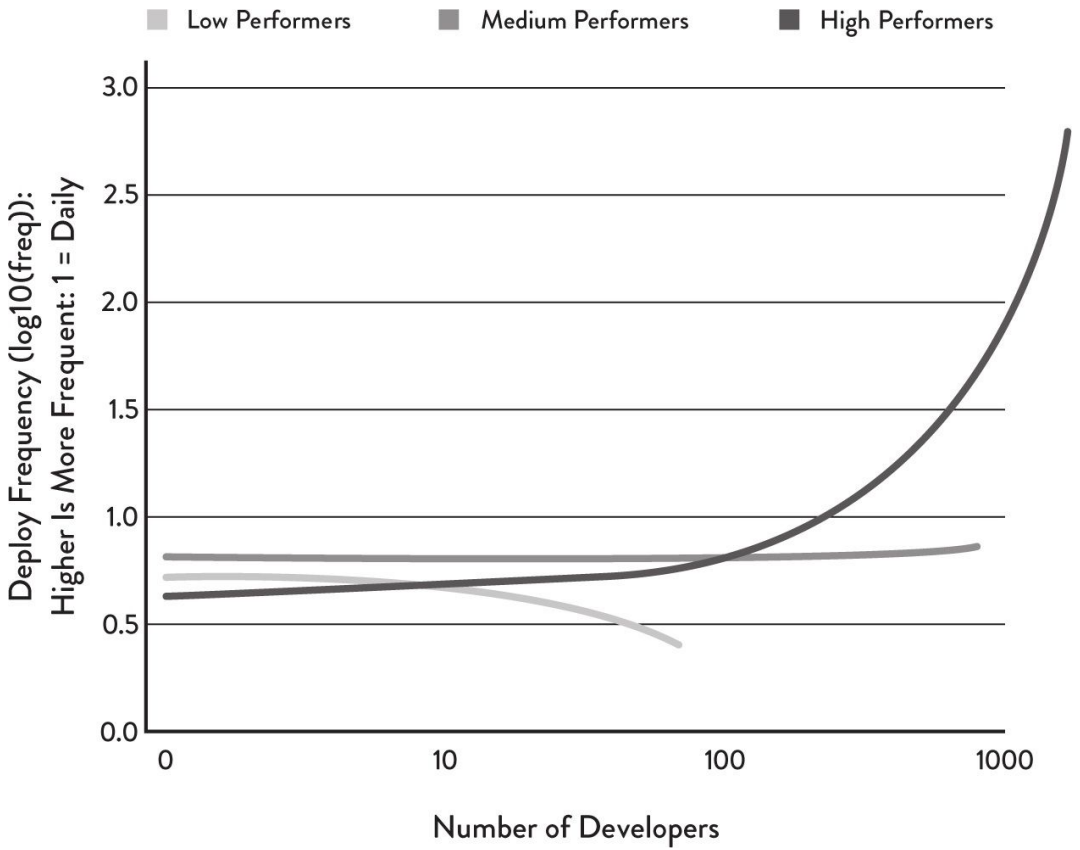
| 2017 | High Performers | Medium Performers | Low Performers |
|---|---|---|---|
| Deployment Frequency | on demand (multiple deploys per day) | Between once per month and once every six months | Between once per month and once every six months (average lower) |
| Lead Time for Changes | Less than one hour | Between one week and one month | Between one week and one month (average lower) |
| MTTR | Less than one hour | Less than one day | Between one day and one week |
| Change Failure Rate | 0-15% | 0-15% | 31-45% |

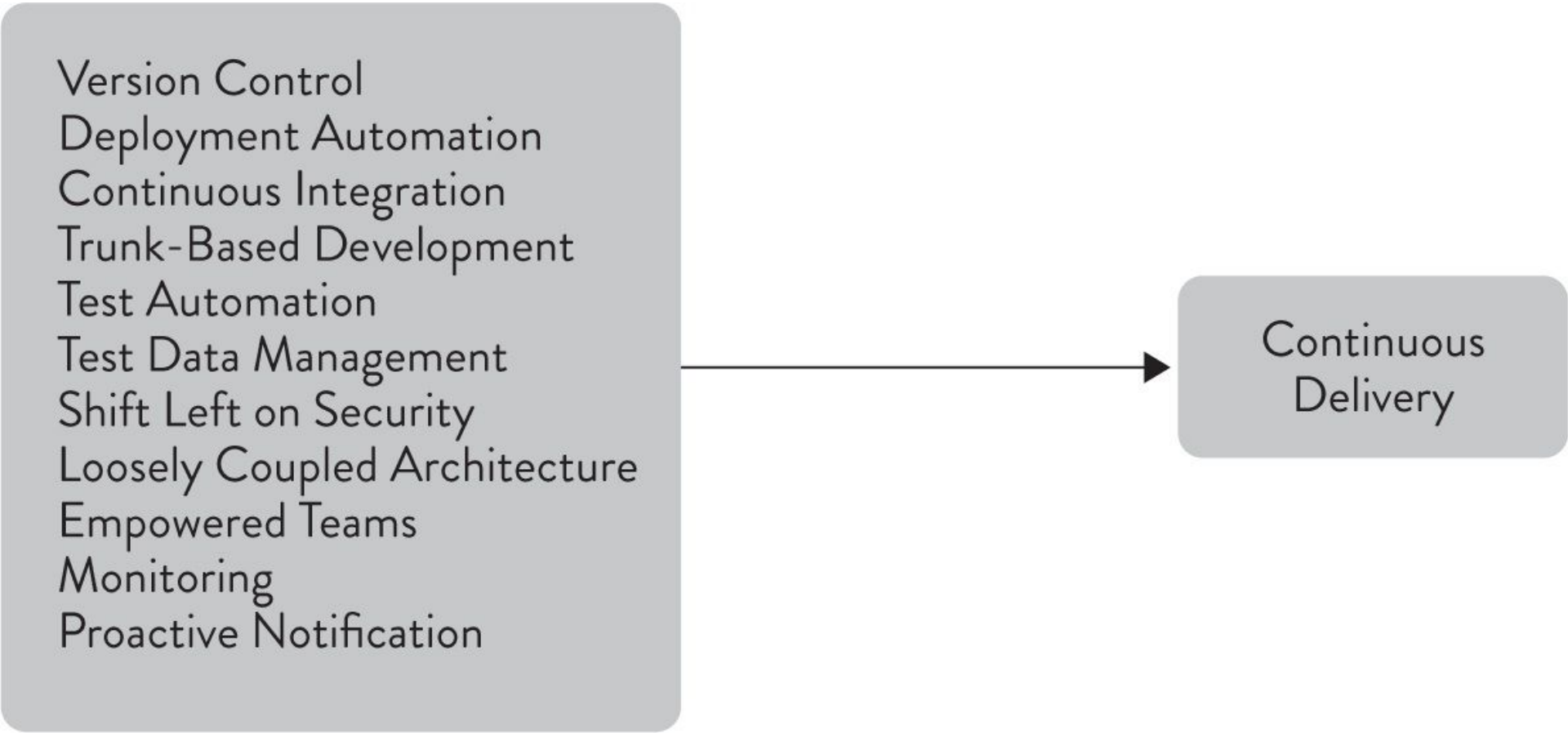

DEPLOY FREQUENCY (# OF DEPLOYS PER YEAR)
1,600
1,200
800
400
0
2014
2015
2016
2017


MEAN TIME TO RECOVERY (HOURS)
1,600
1,200
800
400
0
2014
2015
2016
2017

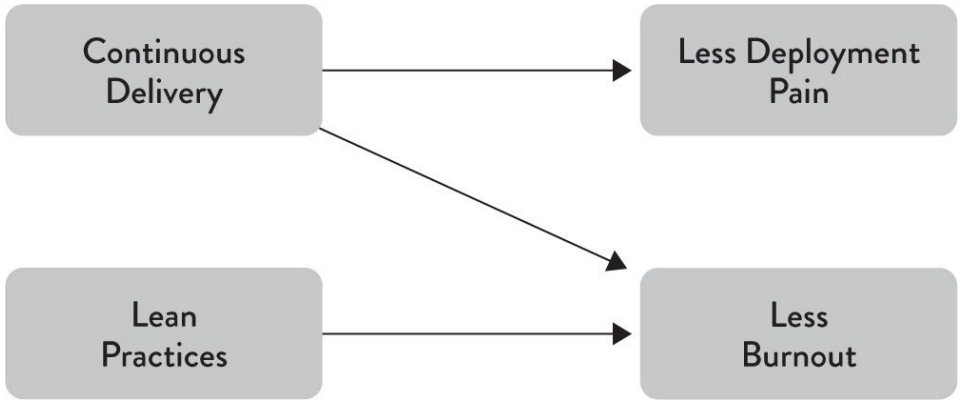

CHANGE LEAD RATE (MINUTES)
160,000
120,000
80,000
40,000
0
2014
2015
2016
2017
High Performers
Low Performers

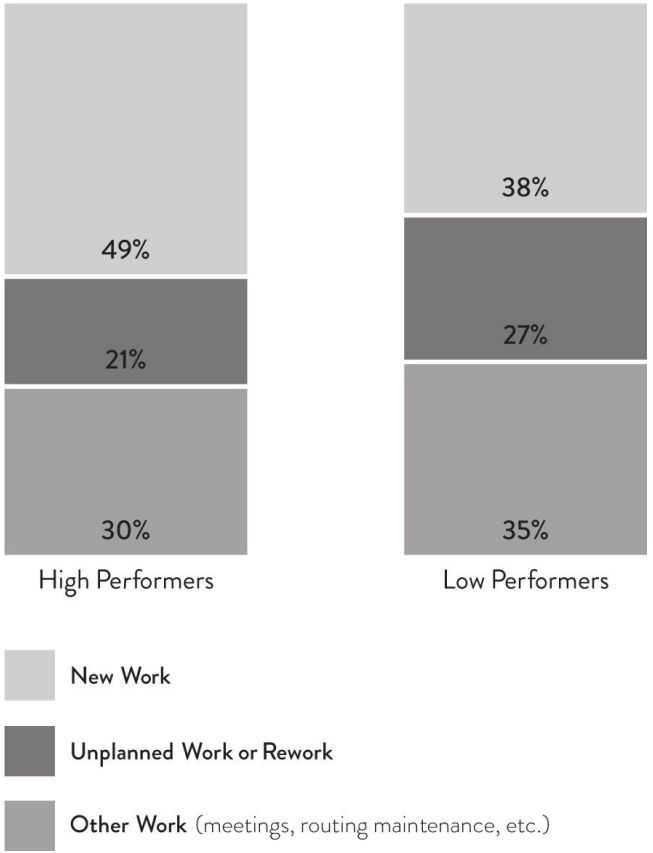

CHANGE FAILURE RATE (PERCENTAGE)
60
50
40
30
20
10
0
2014
2015
2016
2017
High Performers
Low Performers

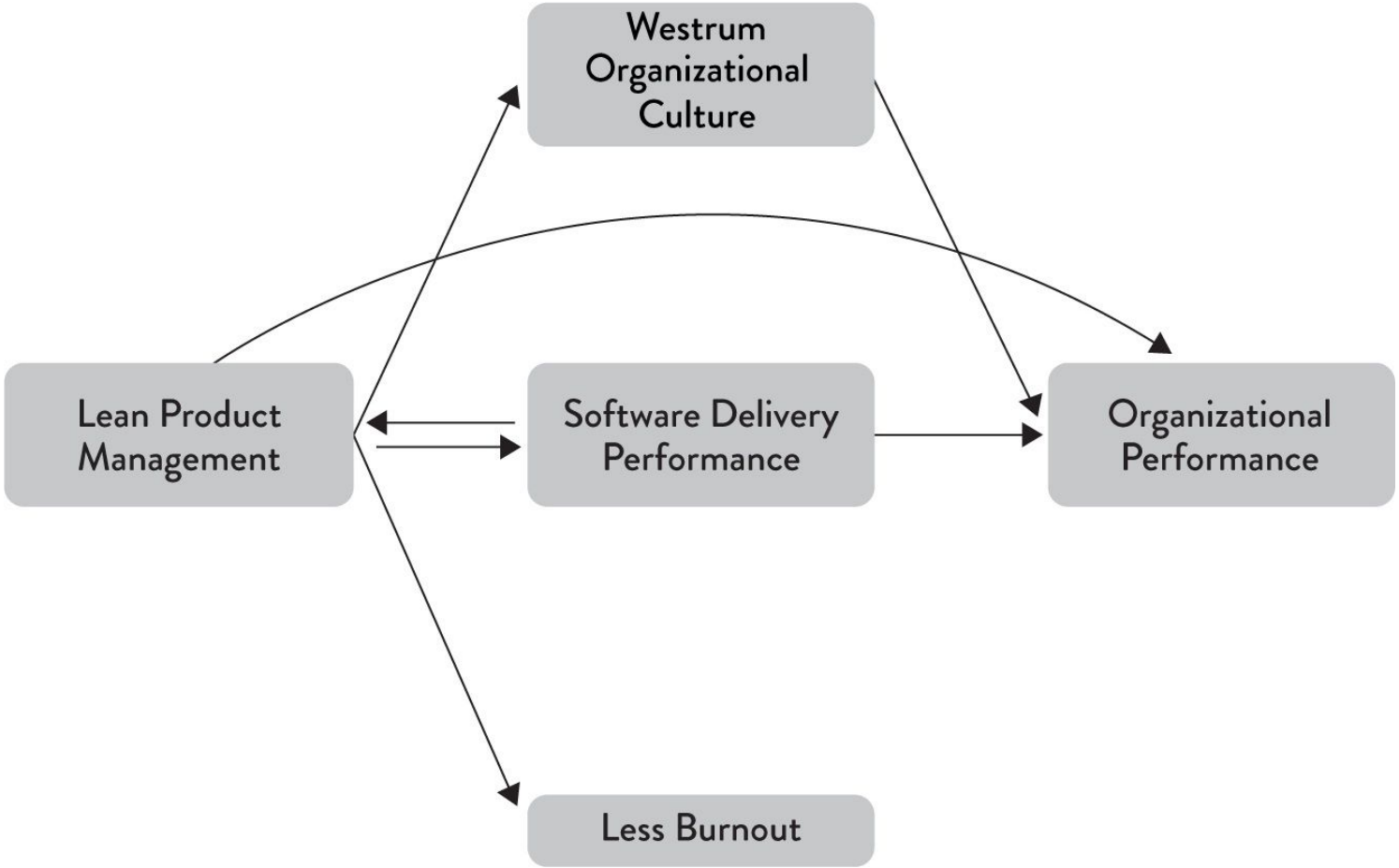

Low Performers
Medium Performers
High Performers
Deploy Frequency (log10(freq)):
Higher Is More Frequent: 1 = Daily
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
10
100
1000
Number of Developers

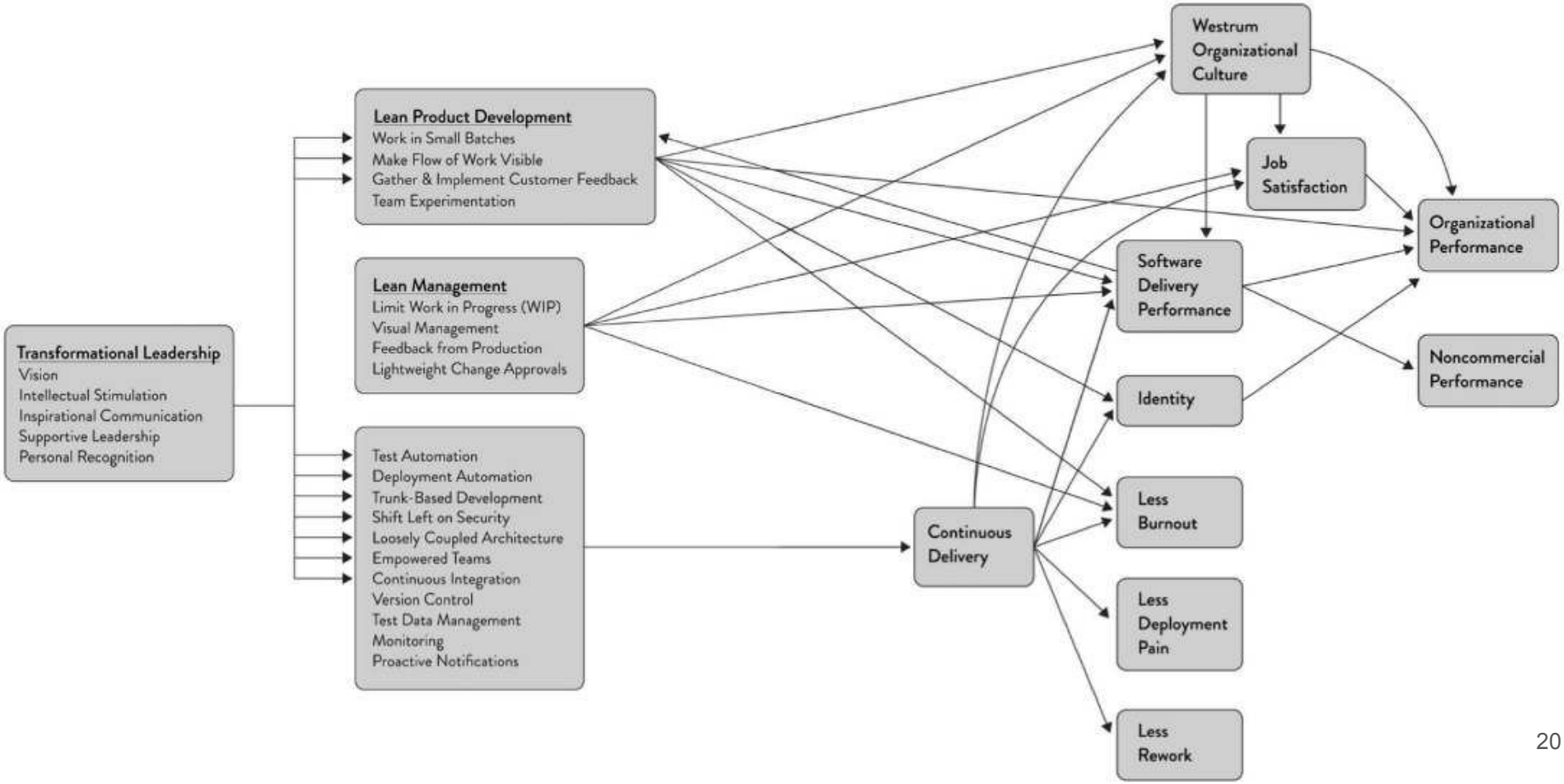

Version Control
Deployment Automation
Continuous Integration
Trunk-Based Development
Test Automation
Test Data Management
Shift Left on Security
Loosely Coupled Architecture
Empowered Teams
Monitoring
Proactive Notification
Continuous Delivery

# Was bringt uns das?

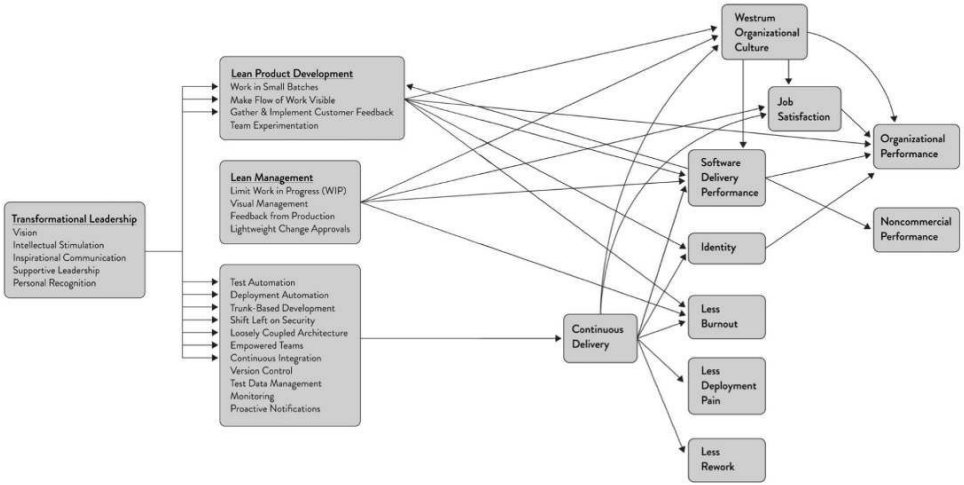


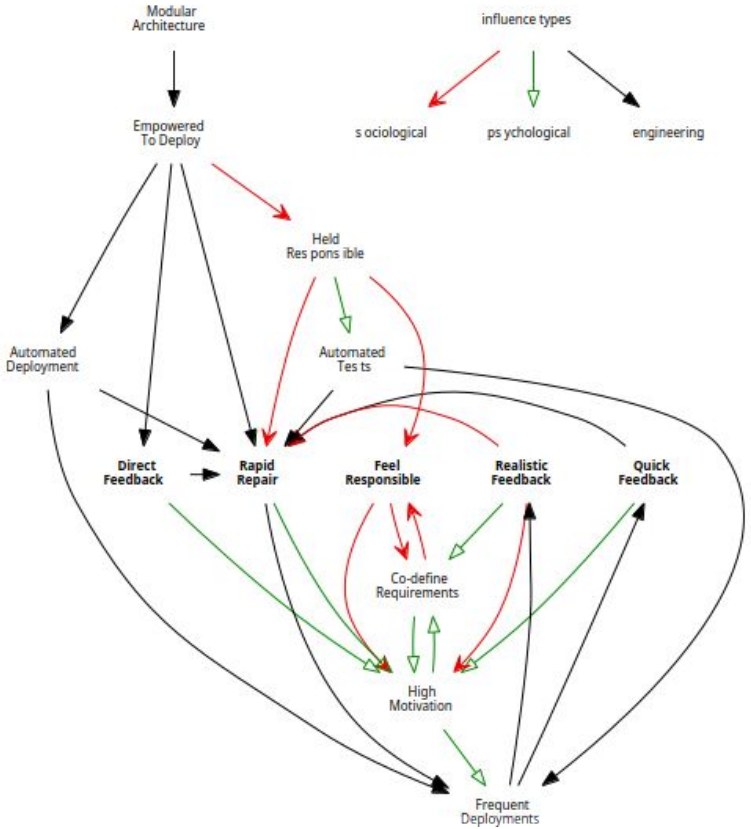




Westrum Organizational Culture
Lean Product Management
Software Delivery Performance
Organizational Performance
Less Burnout


Transformational Leadership
Vision
Intellectual Stimulation
Inspirational Communication
Supportive Leadership
Personal Recognition
Lean Product Development
Work in Small Batches
Make Flow of Work Visible
Gather & Implement Customer Feedback
Team Experimentation
Lean Management
Limit Work in Progress (WIP)
Visual Management
Feedback from Production
Lightweight Change Approvals
Test Automation
Deployment Automation
Trunk-Based Development
Shift Left on Security
Loosely Coupled Architecture
Empowered Teams
Continuous Integration
Version Control
Test Data Management
Monitoring
Proactive Notifications
Continuous Delivery
Westrum Organizational Culture
Job Satisfaction
Organizational Performance
Software Delivery Performance
Noncommercial Performance
Identity
Less Burnout
Less Deployment Pain
Less Rework

# Plausible?


Transformational Leadership
Vision
Intellectual Stimulation
Inspirational Communication
Supportive Leadership
Personal Recognition
Lean Product Development
Work in Small Batches
Make Flow of Work Visible
Gather & Implement Customer Feedback
Team Experimentation
Lean Management
Limit Work in Progress (WIP)
Visual Management
Feedback from Production
Lightweight Change Approvals
Test Automation
Deployment Automation
Trunk-Based Development
Shift Left on Security
Loosely Coupled Architecture
Empowered Teams
Continuous Integration
Version Control
Test Data Management
Monitoring
Proactive Notifications
Continuous Delivery
Westrum Organizational Culture
Job Satisfaction
Software Delivery Performance
Identity
Less Burnout
Less Deployment Pain
Less Rework
Organizational Performance
Noncommercial Performance


Modular Architecture
influence types
Empowered To Deploy
s ociological
ps ychological
engineering
Held Res pons ible
Automated Deployment
Automated Tes ts
Direct Feedback
Rapid Repair
Feel Responsible
Realistic Feedback
Quick Feedback
Co-define Requirements
High Motivation
Frequent Deployments

# Manual Work

| Manual Work | High Performers | Median Performers | Low Performers |
|---|---|---|---|
| Configuration managment | 28% | 47%* | 46%* |
| Testing | 35% | 51%* | 49%* |
| Deployments | 26% | 47% | 43% |
| Change approval precess | 48% | 67% | 59% |

*Differences are not statistically significant between medium and low performers for configuration management and testing.

# Bias Tests

- Chi-square test Katigoriellen daten (vermutung Chi-Quadrat-Homogenitätstest)
- T-tests auf scale werten von frühen und späten antworten
- Common method bias
- Common method variance
- Harmans’s single-factor test
- The maker variable test

# Testing for Relationships

- Principal components analysis
- Average variance extracted
- Correlation
- Reliability:
  - Cornbach’s alpha CR 0.70
  - Composite reliability CR 0.70
- Linear Regression
- Partial least square regression

# Clustering

- Hiracial Clustering (no predefined count)
- Analysis of variance

- deploy frequency
- lead time
- mean time to restore
- change fail rate

# Bias in Questions

Addressed:

- Leading questions
- Loaded questions
- Multiple questions in one
- Unclear language

Other Bias:

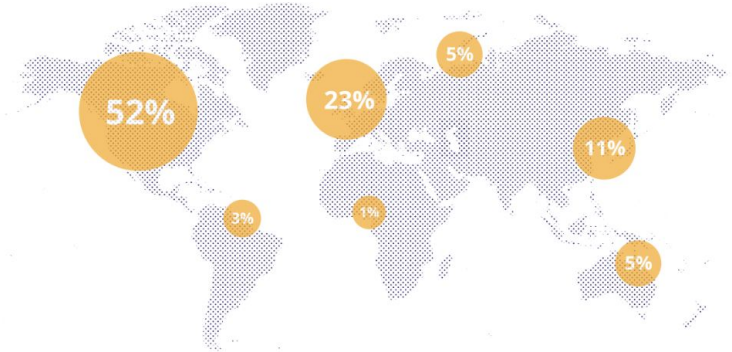
- Soziale erwünschtheit
- Missverständnisse

## Departments


One notable difference this year was an increase in DevOps departments. This year, 19 percent of respondents were part of a DevOps department, up from 16 percent last year.

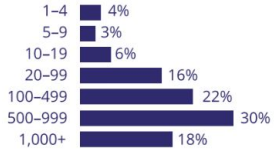
This year, 4,976 respondents completed the 2015 State of DevOps Survey. Compared to last year, we saw similar distributions across geographies, company size, industries and size of infrastructure.

## Geography

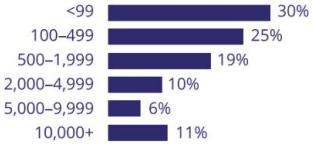



## Industries

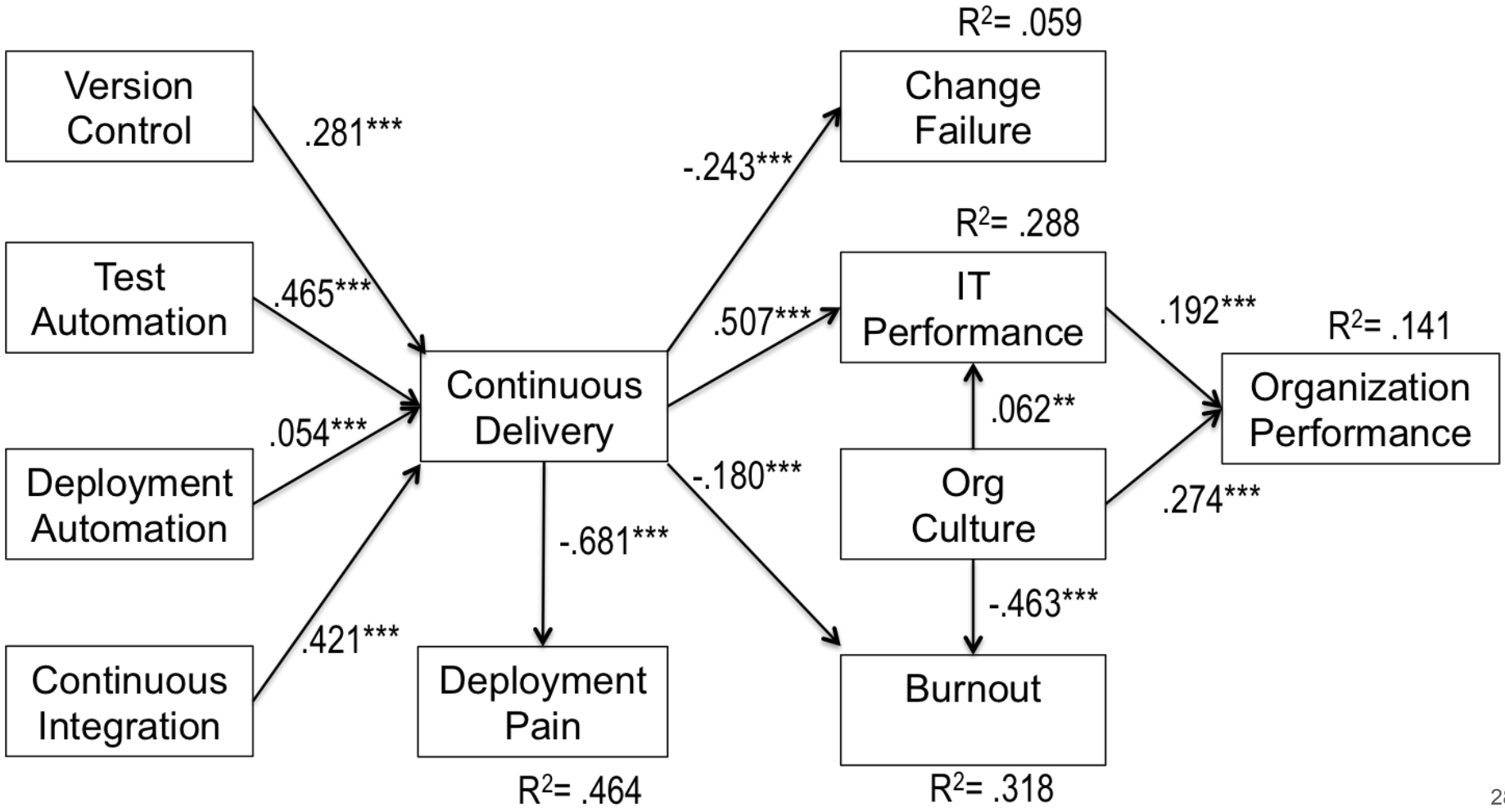

## Size of Infrastructure by Number of Servers




## Company Size




## Gender


Version Control
Test Automation
Deployment Automation
Continuous Integration
Continuous Delivery
Deployment Pain
R2= .464
Change Failure
R2= .059
IT Performance
R2= .288
Org Culture
Burnout
R2= .318
Organization Performance
R2= .141
.281***
.465***
.054***
.421***
-.243***
.507***
-.180***
-.681***
.062**
.192***
.274***
-.463***

# Model by Ron Westrum

**Table 1** How organisations process information

| Pathological | Bureaucratic | Generative |
|---|---|---|
| **Power oriented** | **Rule oriented** | **Performance oriented** |
| Low cooperation | Modest cooperation | High cooperation |
| Messengers shot | Messengers neglected | Messengers trained |
| Responsibilities shirked | Narrow responsibilities | Risks are shared |
| Bridging discouraged | Bridging tolerated | Bridging encouraged |
| Failure→ scapegoating | Failure→ justice | Failure→ inquiry |
| Novelty crushed | Novelty→ problems | Novelty implemented |

# Self check

https://www.surveymonkey.com/r/M7RMCBK

# Diskussion & Ausblick