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

<table>
<thead>
<tr>
<th></th>
<th>2017</th>
<th>High Performers</th>
<th>Medium Performers</th>
<th>Low Performers</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Deployment Frequency</strong></td>
<td>on demand (multiple deploys per day)</td>
<td>Between once per month and once every six months</td>
<td>Between once per month and once every six months (average lower)</td>
<td></td>
</tr>
<tr>
<td><strong>Lead Time for Changes</strong></td>
<td>Less than one hour</td>
<td>Between one week and one month</td>
<td>Between one week and one month (average lower)</td>
<td></td>
</tr>
<tr>
<td><strong>MTTR</strong></td>
<td>Less than one hour</td>
<td>Less than one day</td>
<td>Between one day and one week</td>
<td></td>
</tr>
<tr>
<td><strong>Change Failure Rate</strong></td>
<td>0-15%</td>
<td>0-15%</td>
<td>31-45%</td>
<td></td>
</tr>
</tbody>
</table>
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?

- Continuous Delivery
  - Less Deployment Pain
    - High Performers
      - New Work: 49%
      - Unplanned Work or Rework: 21%
      - Other Work: 30%
    - Low Performers
      - New Work: 38%
      - Unplanned Work or Rework: 27%
      - Other Work: 35%
- Lean Practices
  - Less Burnout
Plausible?
### Manual Work

<table>
<thead>
<tr>
<th>Manual Work</th>
<th>High Performers</th>
<th>Median Performers</th>
<th>Low Performers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration management</td>
<td>28%</td>
<td>47%*</td>
<td>46%*</td>
</tr>
<tr>
<td>Testing</td>
<td>35%</td>
<td>51%*</td>
<td>49%*</td>
</tr>
<tr>
<td>Deployments</td>
<td>26%</td>
<td>47%</td>
<td>43%</td>
</tr>
<tr>
<td>Change approval process</td>
<td>48%</td>
<td>67%</td>
<td>59%</td>
</tr>
</tbody>
</table>

* 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
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.

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.
## Table 1  How organisations process information

<table>
<thead>
<tr>
<th>Pathological</th>
<th>Bureaucratic</th>
<th>Generative</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Power oriented</strong></td>
<td><strong>Rule oriented</strong></td>
<td><strong>Performance oriented</strong></td>
</tr>
<tr>
<td>Low cooperation</td>
<td>Modest cooperation</td>
<td>High cooperation</td>
</tr>
<tr>
<td>Messengers shot</td>
<td>Messengers neglected</td>
<td>Messengers trained</td>
</tr>
<tr>
<td>Responsibilities shirked</td>
<td>Narrow responsibilities</td>
<td>Risks are shared</td>
</tr>
<tr>
<td>Bridging discouraged</td>
<td>Bridging tolerated</td>
<td>Bridging encouraged</td>
</tr>
<tr>
<td>Failure→</td>
<td>Failure→</td>
<td>Failure→</td>
</tr>
<tr>
<td>scapegoating</td>
<td>justice</td>
<td>inquiry</td>
</tr>
<tr>
<td>Novelty crushed</td>
<td>Novelty→ problems</td>
<td>Novelty implemented</td>
</tr>
</tbody>
</table>
Self check

https://www.surveymonkey.com/r/M7RMCBK
Diskussion & Ausblick