Why DevOps?
IS IT SAFE AND POSSIBLE TO RELEASE WORK AT ANY TIME?
Development | Functional Testing | Performance Testing | Operations

[Diagram of a construction crane with labels for Development, Functional Testing, Performance Testing, and Operations.]
Ronald Westrum shows that a Generative Culture is predictive of higher safety outcomes. Generative Cultures have the following traits:

- High co-operation
- Messengers trained
- Risks are shared
- Bridging encouraged
- Failure leads to inquiry
- Novelty implemented

High performers

- Deployment frequency = On demand (Multiple times per day)
- Lead time for changes = Less than 1 hour
- Mean Time to Recover = Less than 1 hour
- Change Failure Rate = 0% - 15%

DevOps Research and Assessment (DORA) has found that for high performing IT organizations:

- correlate to Generative Cultures
- are twice as likely to exceed their profitability, market share and productivity goals
- achieved higher levels of throughput AND stability
- spend more time on new work / less on rework

Also: IT performance is predictive of the performance of the organization as a whole.

DevOps—Cost of Quality

- Agile testing is orders-of-magnitude more efficient
- Based on millions of automated tests run in seconds
- One-touch auto-delivery to billions of global end-users

<table>
<thead>
<tr>
<th>Activity</th>
<th>Def</th>
<th>CoQ</th>
<th>DevOps Economics</th>
<th>Hours</th>
<th>ROI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development Operations</td>
<td>100</td>
<td>0.01</td>
<td>100 Defects x 70% Efficiency x 0.001</td>
<td>0.070</td>
<td>72,900%</td>
</tr>
<tr>
<td>Continuous Delivery</td>
<td>30</td>
<td>0.01</td>
<td>30 Defects x 70% Efficiency x 0.01</td>
<td>0.210</td>
<td>24,300%</td>
</tr>
<tr>
<td>Continuous Integration</td>
<td>9</td>
<td>0.1</td>
<td>9 Defects x 70% Efficiency x 0.1</td>
<td>0.630</td>
<td>8,100%</td>
</tr>
<tr>
<td>Software Inspections</td>
<td>3</td>
<td>1</td>
<td>2.7 Defects x 70% Efficiency x 1</td>
<td>1.890</td>
<td>2,700%</td>
</tr>
<tr>
<td>&quot;Traditional&quot; Testing</td>
<td>0.81</td>
<td>10</td>
<td>0.81 Defects x 70% Efficiency x 10</td>
<td>5.670</td>
<td>900%</td>
</tr>
<tr>
<td>Manual Debugging</td>
<td>0.243</td>
<td>100</td>
<td>0.243 Defects x 70% Efficiency x 100</td>
<td>17.010</td>
<td>300%</td>
</tr>
<tr>
<td>Operations &amp; Maintenance</td>
<td>0.073</td>
<td>1,000</td>
<td>0.0729 Defects x 70% Efficiency x 1,000</td>
<td>51.030</td>
<td>n/a</td>
</tr>
</tbody>
</table>

*4,500 x Faster than Code Inspections*

How do we start?
What will we see?
Branches live for less than a day before being merged
System health is monitored proactively
The majority of primary business value has test coverage and tests are run when code is committed
Work-in-progress limits are used to monitor and improve flow and throughput
Work is decomposed into small batches of less than a week’s effort
Test data is adequately available
The team has visibility into how their work is being received and is free to improve things based on that awareness.