



# Trace3

**Engineering DevOps Right  
the First Time**





# Marc Hornbeek

Principal Consultant – DevOps,



39 years experience architecting, designing, developing and managing high-performance solutions for IT and engineering infrastructures deployed in commercial and government applications globally.

- Served as CEO, Board Member, founder, corporate executive, CTO, VP, General Manager, Principal Consultant, Senior Solutions Architect and Professional Engineer.
- Held key roles at Bell-Northern Research, Tekelec, ECI Telecom, GSI Lumonics, Vpacket, EdenTree Technologies, Spirent Communications and Trace3.
- Innovation lead over many successful automation, Lab-as-a-Service and DevOps projects for systems manufacturers and operators.
- Regular speaker, blogger, author and educator on topics including DevOps, Lab-as-a-Service and continuous test automation.
- Author for the DevOps Institute, DevOps Continuous Delivery Architect course and the DevOps Test Engineer course.
- 41-year senior member of the IEEE, 2016 outstanding engineer
- IEEE Region 6 - Western United States.



*“DevOps-the-gray”*



Skills: Consulting – DevOps, LaaS, QA, Test Automation, Engineering Leadership

<https://www.linkedin.com/in/marchornbeek>

<http://devops.com/author/marc-hornbeek/>

Skype: mhexcalibur

Twitter: mhexcalibur

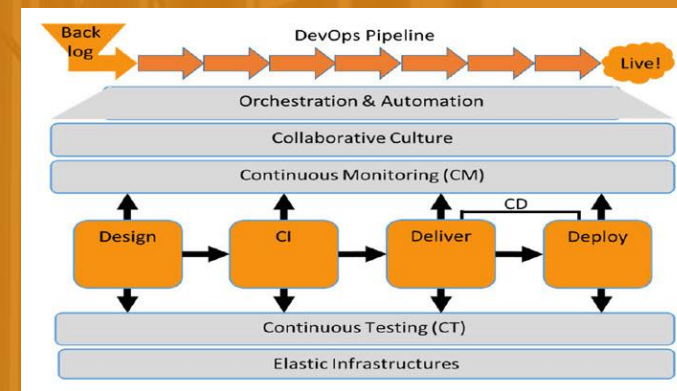
# References

<http://computer.ieee-bv.org/wp-content/uploads/2015/10/2015-09-09-DevOps.pdf>



[http://computer.ieee-bv.org/wp-content/uploads/2016/03/2016-03-09\\_Test\\_Automation.pdf](http://computer.ieee-bv.org/wp-content/uploads/2016/03/2016-03-09_Test_Automation.pdf)

[https://trace3.com/wp-content/uploads/2016/09/Engineering\\_Devops\\_to\\_Meet\\_Business\\_Goals.pdf](https://trace3.com/wp-content/uploads/2016/09/Engineering_Devops_to_Meet_Business_Goals.pdf)



Companies with high-performing IT organizations are twice as likely to exceed their profitability, market share and productivity goals.



**Agility**

200x  
More frequent deployments

2,555x  
Shorter lead times

**Efficiency**

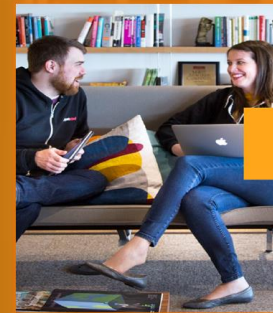
22%

Less time on unplanned work and rework

**Satisfaction**

2.2x

Employees more likely to recommend their organizations as a great place to work.



**Stability**

24x

Faster recovery time for failures

**Security**

50%

Less time remediating security issues

**Quality**

3x

3x lower change failure rate

Only a minority of enterprises are achieving high-performance DevOps implementations. Many are struggling to realize DevOps at all, at the level of business units and enterprise.





## *Dev Perspective*

- ✓ Velocity
- ✓ Productivity
- ✓ Continuous Delivery

# *Ops perspective*

- ⊗ Application quality problems
- ⊗ Security events
- ⊗ Pipeline failures
- ⊗ Interruptive Reverts
- ⊗ Process delays
- ⊗ Schedule delays
- ⊗ Cost overruns
- ⊗ Audit failures



# Cultural Divide



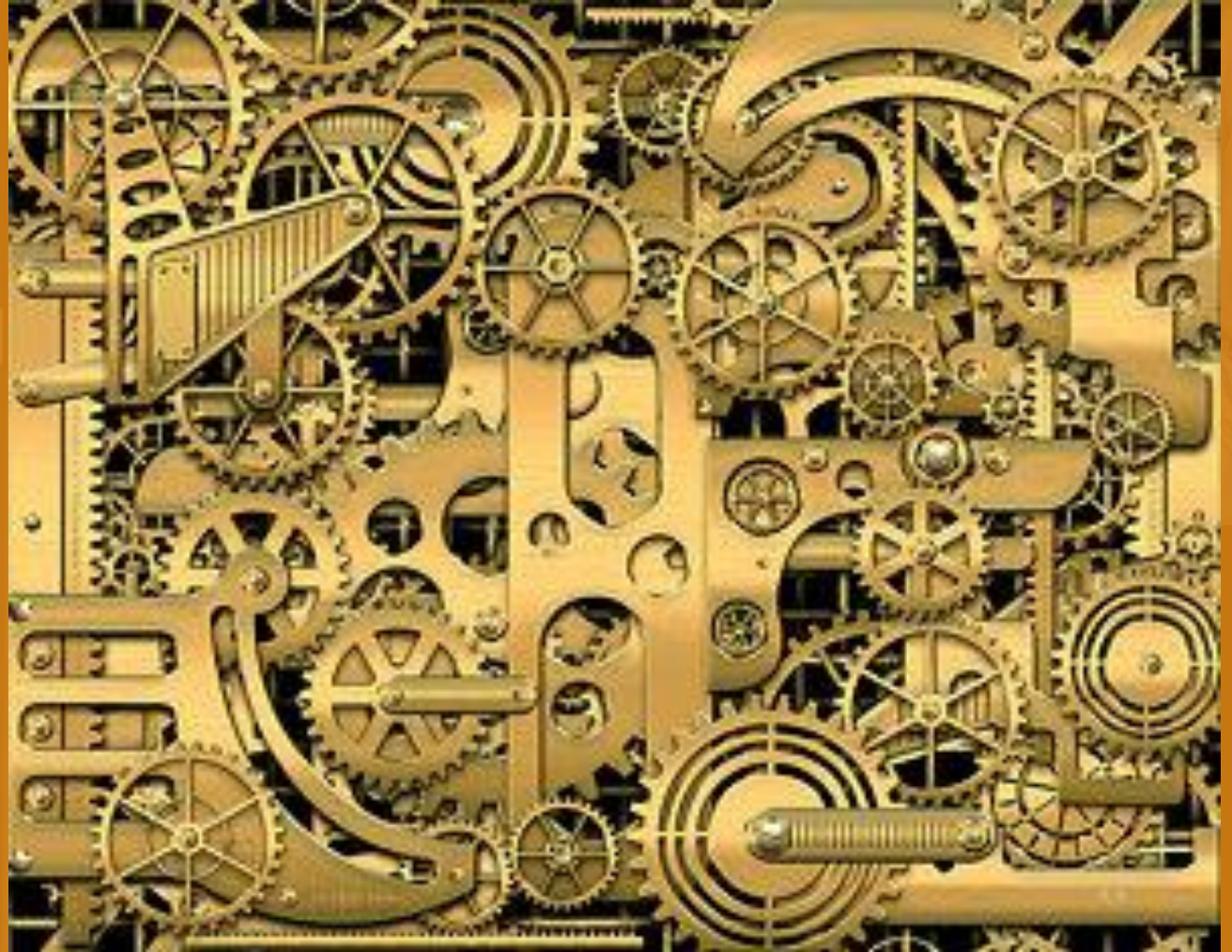
Dev “Agility!”

Ops “Stability!”



# Technology Complexity

- Applications & data
- Process artifacts
- Tools & infrastructure
- Controlled workflows
- Product and process MTTR



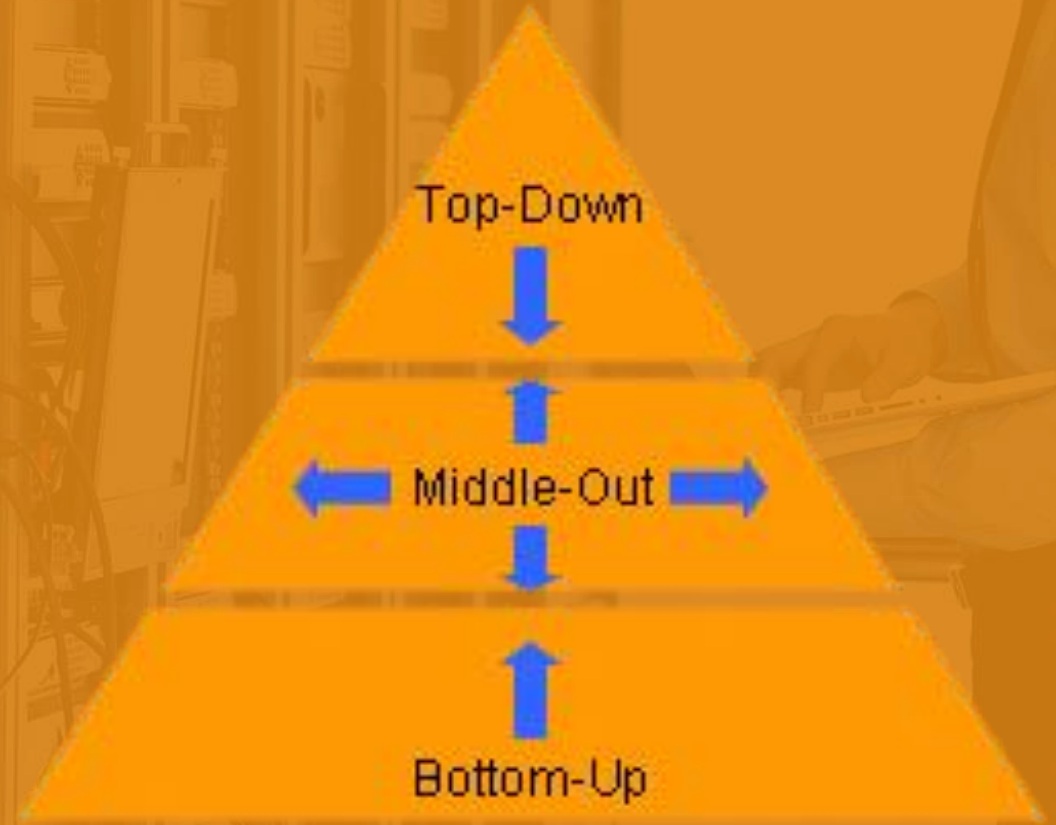
# Organic Bottom-up Approach

- Evolution process
- Local initiatives
- Competing solutions
- Lots of false starts
- Winners and losers



# Top-down/middle-out approach

- Strategic
- Leader sponsored experiments
- Accelerated convergence
- Everybody is a winner !



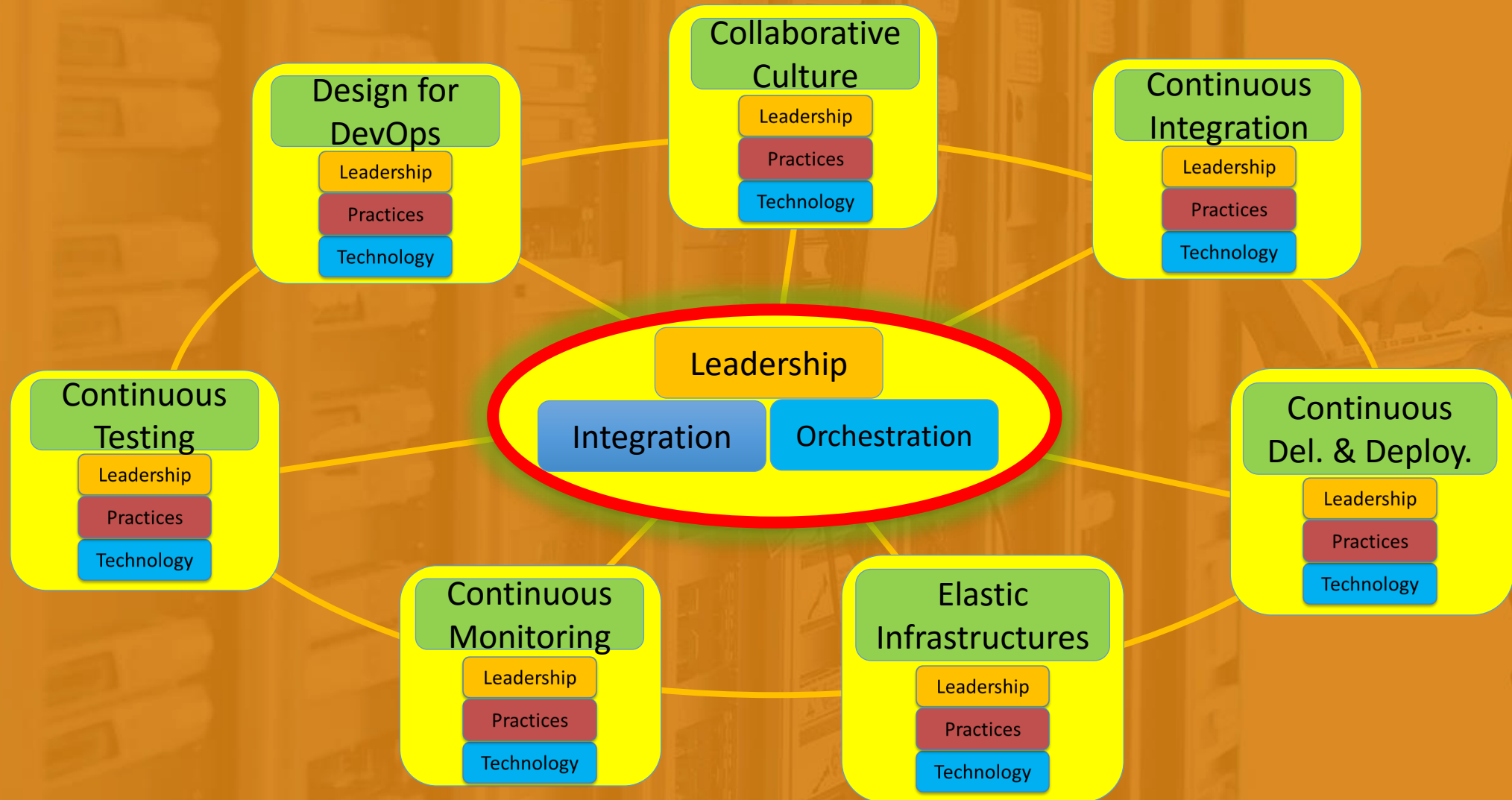
# 7 Pillars of DevOps

- Collaborative culture
- Design for DevOps
- Continuous Integration (CI)
- Continuous Testing (CT)
- Continuous Monitoring (CM)
- Elastic Infrastructures
- Continuous Delivery and Deployment (CD)



<http://trace3.com/blog/7-pillars-of-devops-essential-foundations-for-enterprise-success-2/#.WDI5kMfrtqM>

# DevOps Transformation requires a Symphony



# Top-down/middle-out preferred over silo'd/organic

## Silo'd, organic

- Organizational goals / friction
- False starts, high % of failures
- Dissatisfied staff
- Silos of success every 6 months but some counter-productive
- 5+ years to enterprise-wide success



## Top-down/middle-out

- Business level goals
- Transformed leadership
- Satisfied staff
- Progressive business success every 4-6 months
- 2 years to enterprise-wide success



# Leadership & Culture Transformation

**Managing leaders**

**Organization silos**

**Functional goals, rewards**

**Local processes**

**Local infra/tools**

**Local training**

**Local level KPIs**

**Facilitating leaders**

**Collaborative org structure**

**Business goals, rewards**

**Local services**

**Infra/tools as services**

**Cross training**

**Business level SLAs**



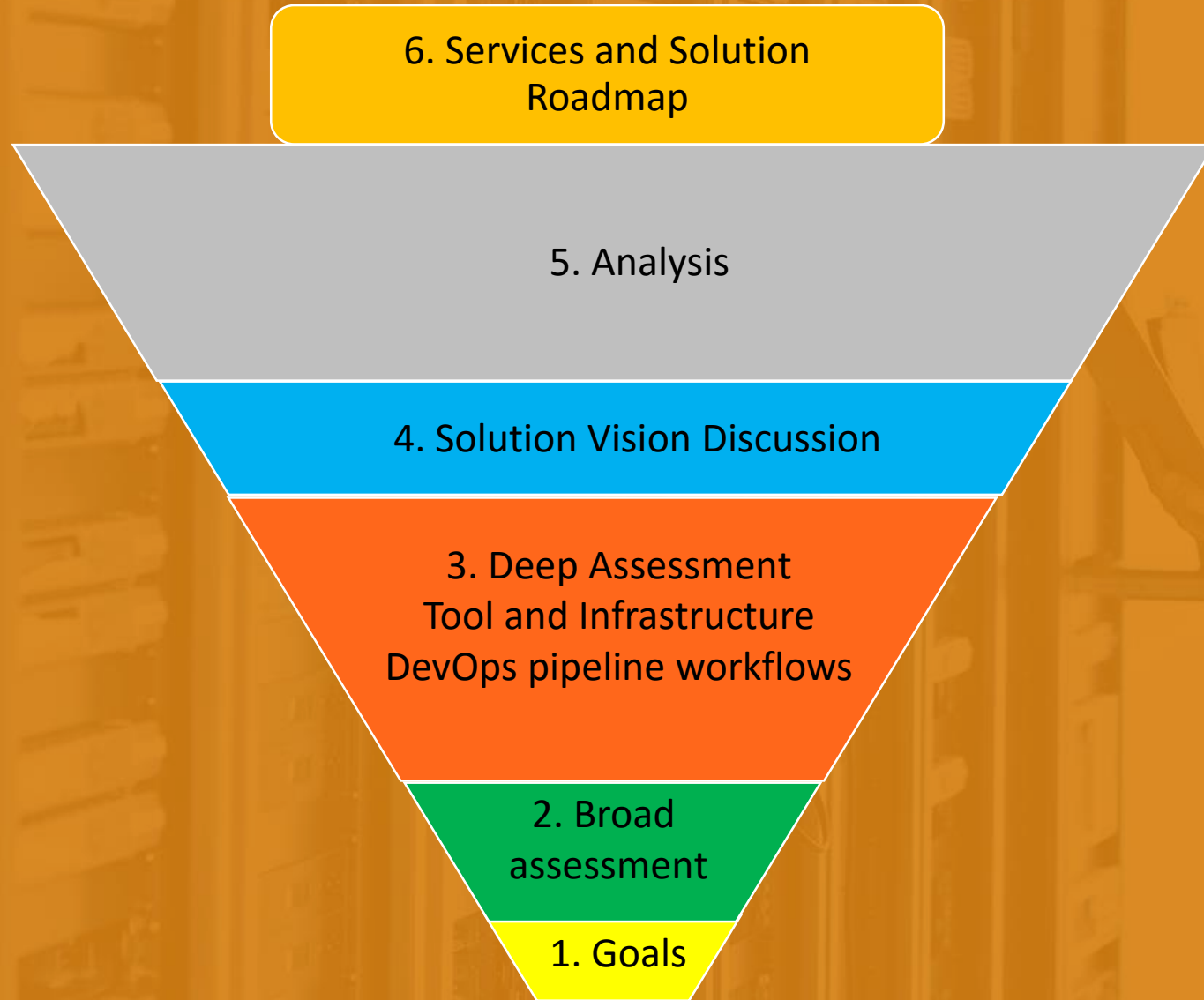
**Courage**

**Passion**

**Investment**

**Measurement**

# Strategic Engineering Approach





# Journey of Engineering Engagement & Evolution



Services and Solution recommendation

Analysis with customer interactions

Solution Vision Discussion

Deep Assessment  
Tool and Infrastructure  
DevOps pipeline workflows

Broad assessment

Goals

Business Milestone

Business Milestone

Business Milestone

Business Milestone

Business Milestone

Services and Solution recommendation  
Analysis with customer interactions  
Solution Vision Discussion  
Deep Assessment  
Tool and Infrastructure  
DevOps pipeline workflows  
Broad assessment  
Goals

# Leadership Alignment

Business improvement goals (Example):

- 1) Reduce time-to-market from 30 days to 7 days.
- 2) Make workflows visible to all stakeholders.
- 3) Change the culture have good communication flow, cooperation and trust.
- 4) Reduce the frequently of failures from weekly to monthly.
- 5) Reduce unplanned work from 30% to 20%.

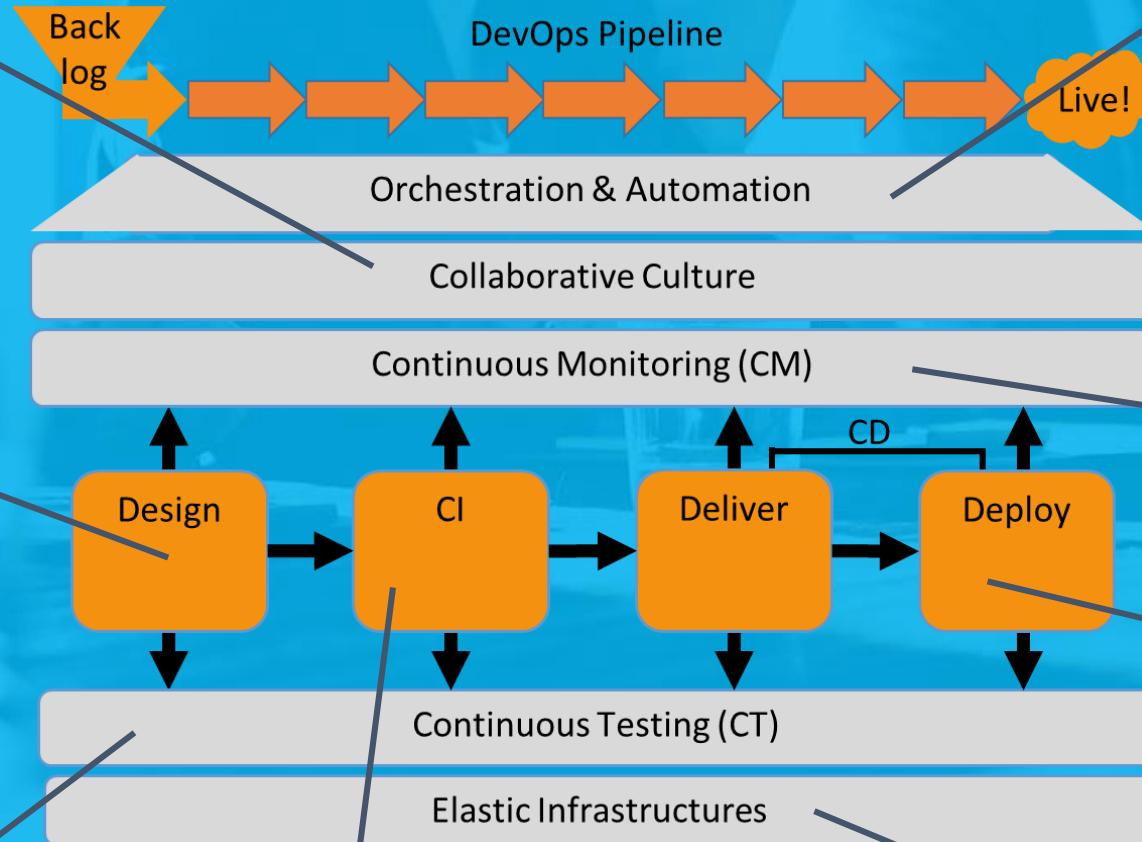


# Gap Assessment

## Where are you now?

Does your culture have DevOps goals, rewards and training for leaders and staff?

Do you have an orchestration tool that forms the backbone of an end-to-end pipeline?



Are your designers designing according to the principles of 12-factor apps.?

Can Dev, QA and Ops see how each feature and release is performing at each stage in the pipeline?

How many of your tests are automated for each stage in the pipeline?

How long does it take to build and package a new version?

How long does it take anyone in Dev, QA or Ops to obtain a test environment?

How long does it take to deploy a change?

# Process Re-engineering

Process engineering optimizations requires balancing three dimensions across multiple dependent pipeline stages.

## Agility

Lack of innovation  
Re-invest resources freed by improved efficiency.

## Efficiency

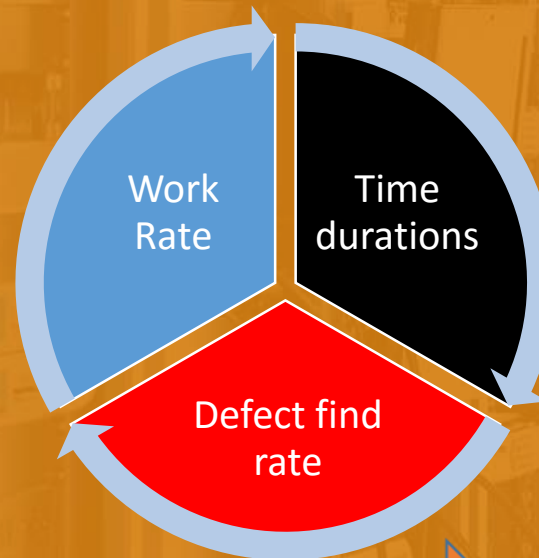
Wasted resources  
Continuous Integration, micro-services & elastic Infrastructures

## Quality

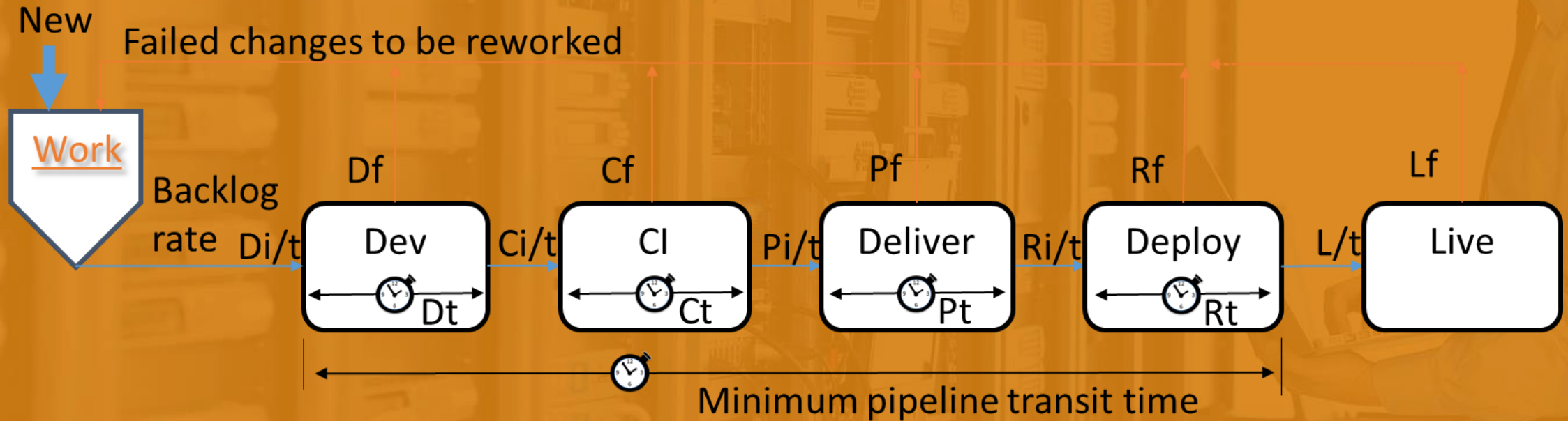
Failure frequency  
Continuous Testing and Monitoring

## Stability

Long problem fix time  
Modular Continuous delivery & Deployments



# Model the DevOps Pipeline



To meet agility, efficiency, quality and stability goals it is necessary to tune and balance process input rates, stage durations, and stage defect find rates.

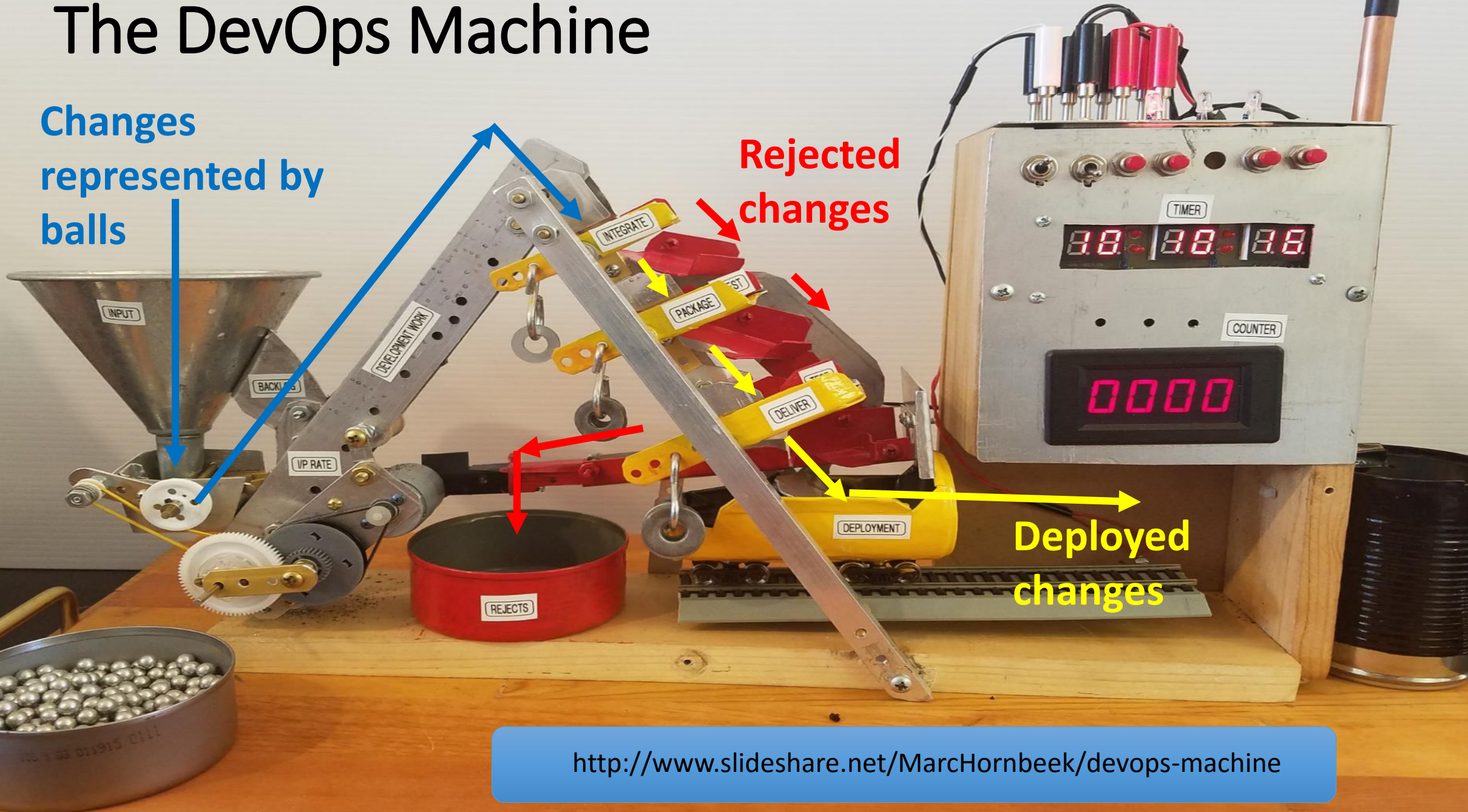
- Adjust input rates to meet required output rates.
- Adjust defect find rates so that bugs are found during earlier stages
- Adjust stage processing time to equalize continuous flow.

# The DevOps Machine

Changes represented by balls

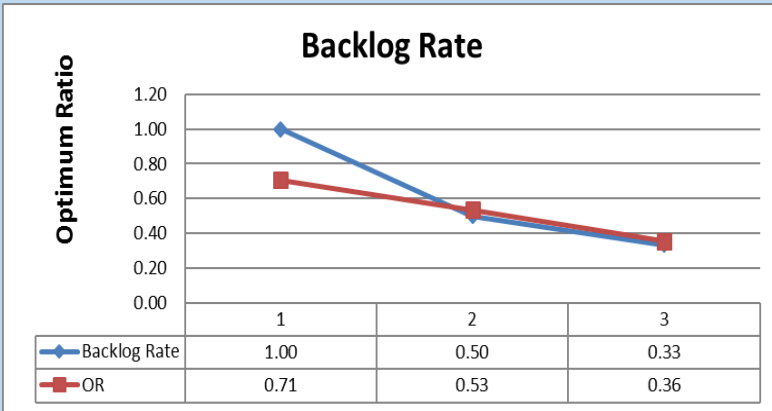
Rejected changes

Deployed changes

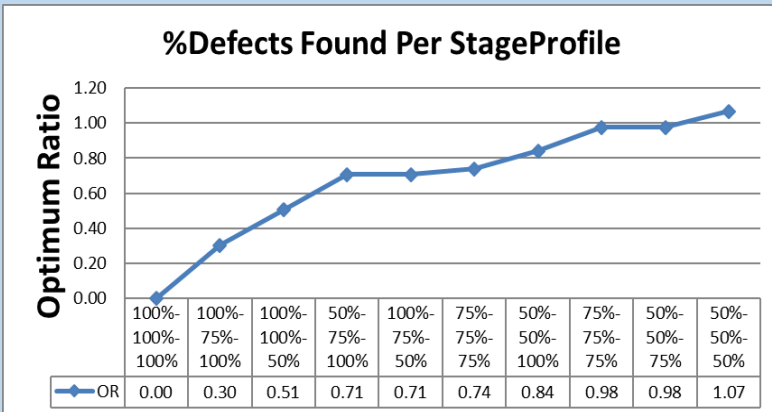


# Results of simulation runs

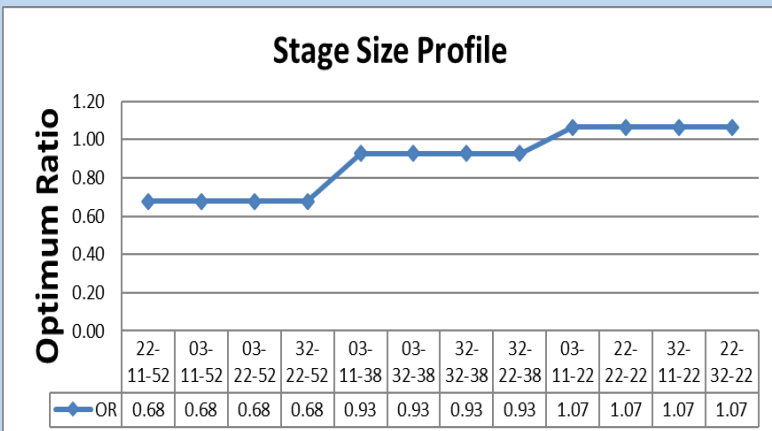
Faster backlog rate is best



Finding more defects early and throughout pipeline is best



Smaller later stages is best

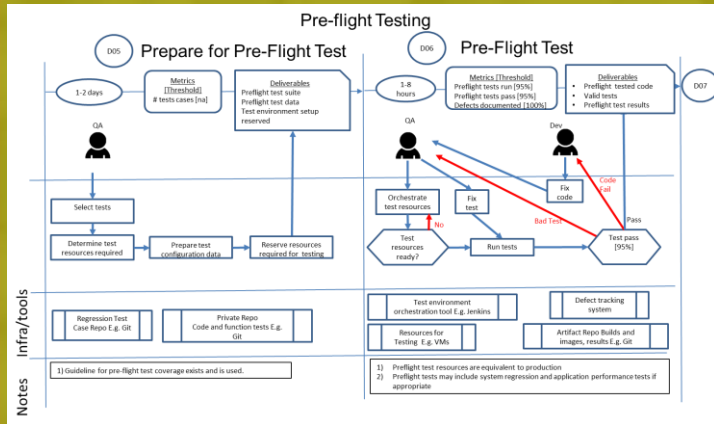


Optimum agility, efficiency, quality and stability were achieved when **input rates** are highest, **stage durations** are short, most **bugs are found during earlier stages** of the pipeline, and the time between stages is equal so there is **continuous flow**.

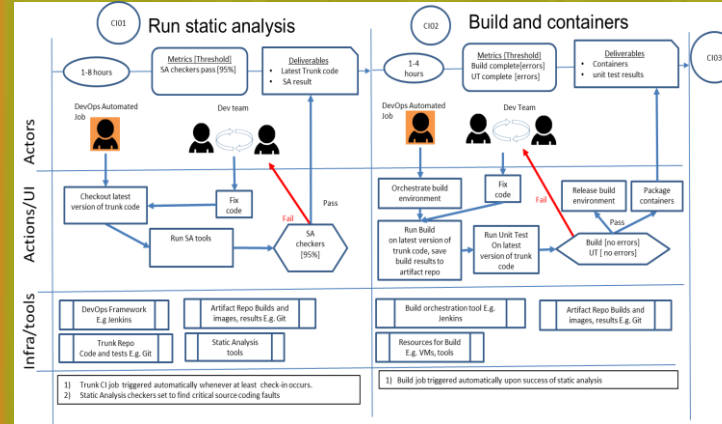
These results are satisfying because they confirm a number of DevOps tenets: “Deliver frequently”, “Fail Fast”, “Fail Early”, and “Fail Often”. The model demonstrated the basis of the business value of DevOps: accelerate of innovation with quality. 😊

# Analyze Pipeline Elements – Time, Tools, Metrics

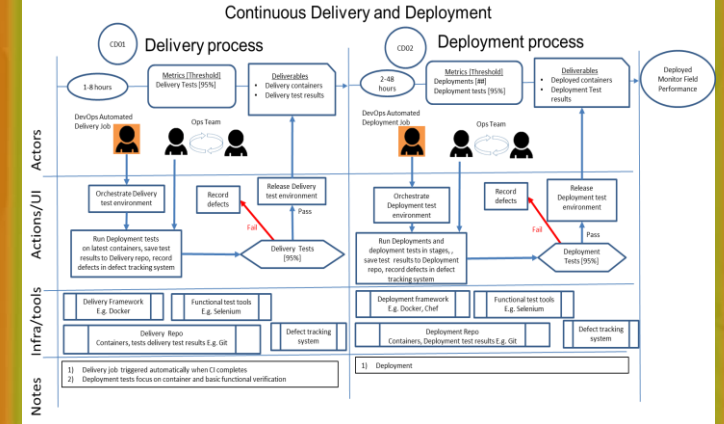
## Dev



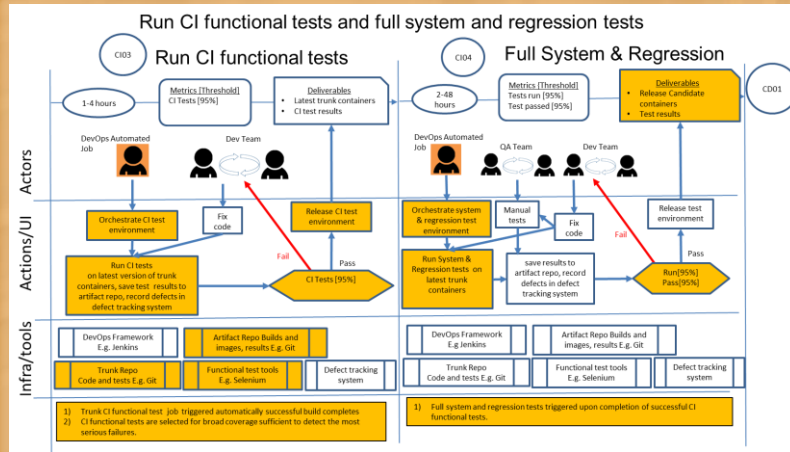
## CI



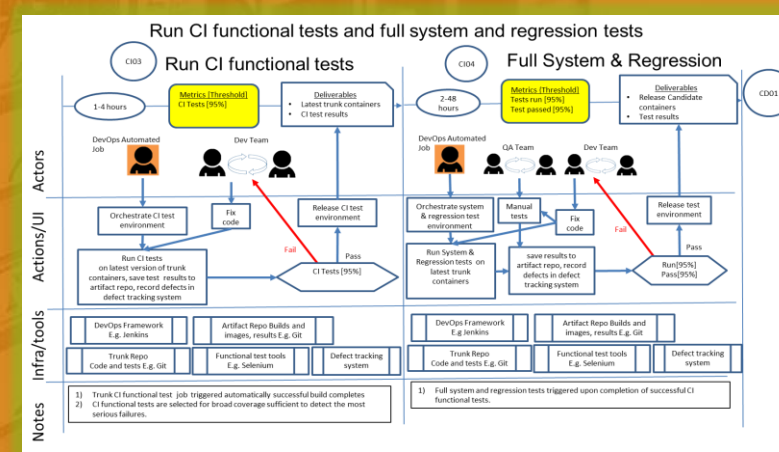
## CD



## CT



## CM





# DevOps Pipeline Engineering

- 1) Using the timing goals, determine a time budget for each stage that will equalize the stage timings and meet the end-to-end timing goal. (Total time/# stages)
- 2) Set a failure budget for each stage such that each stage fails at least 20% more than the subsequent one. ( $\text{Failure\_Rate}^{\text{Stage}(X)} = 1.2 \times \text{Failure\_Rate}^{\text{Stage}(X+1)}$ )
- 3) Set the input rate equal to the goal release rate increased by the sum of the expected cumulative failures through the pipeline.
- 4) Identify strategies to shorten the stage times that will meet the end-to-end time.
- 5) Identify strategies to increase defect find rates higher in earlier stages to match the budgets.
- 6) Rework the work-flows to match the new time and defect detection budgets.
- 7) Identify team, process, tools and infrastructure changes

# Sample tactics - shorten stage times

1. Task automation - eliminate manual tasks
2. Faster stage transitions - automate gate checks
3. Reduce waste by halting processes upon verdict failures
4. Horizontal scaling - process modules in parallel
5. Vertical scaling - process each module faster
6. Dynamic scheduling – only process what is relevant to the changes in the pipeline



# Sample tactics- Shift left defect detection

1. Increase test coverage (E.g.. Static analysis checkers, more test cases, better tests)
2. Test using production configurations earlier.
3. Test-Driven-Development TDD
4. Pre-Flight testing
5. Reduce wasted test time by halting processes upon failed jobs
6. Horizontal test scaling - run tests in parallel
7. Vertical test scaling - run test processes faster
8. Dynamic scheduling – only test what is relevant for changes in the pipeline



# Sample tactics - security

1. Use security static analysis during the Dev and CI stages with security checkers turned on.
2. Do not put credentials in automation scripts!
3. Include dynamic pen and DDS security test cases in functional & regression tests
4. Security test all production variations of OSs and browsers
5. Live audits of production node for latest security patches supported by CMM
6. ACLs for pipeline stage artifacts & multi-tenancy labs

## Security

Unauthorized uses  
**Security integrated  
into end-to-end  
pipeline**



# Sample tactics - Satisfaction

1. Leaders frequently clarify business goals and results with all staff.
2. Redefine or create new roles as needed to fill gaps.
3. Empower staff to stop the pipeline and call for action if they see problems. (The red button on the assembly line.)
4. Training for DevOps practices and skills.
5. Awards and incentives for collaborative successes.

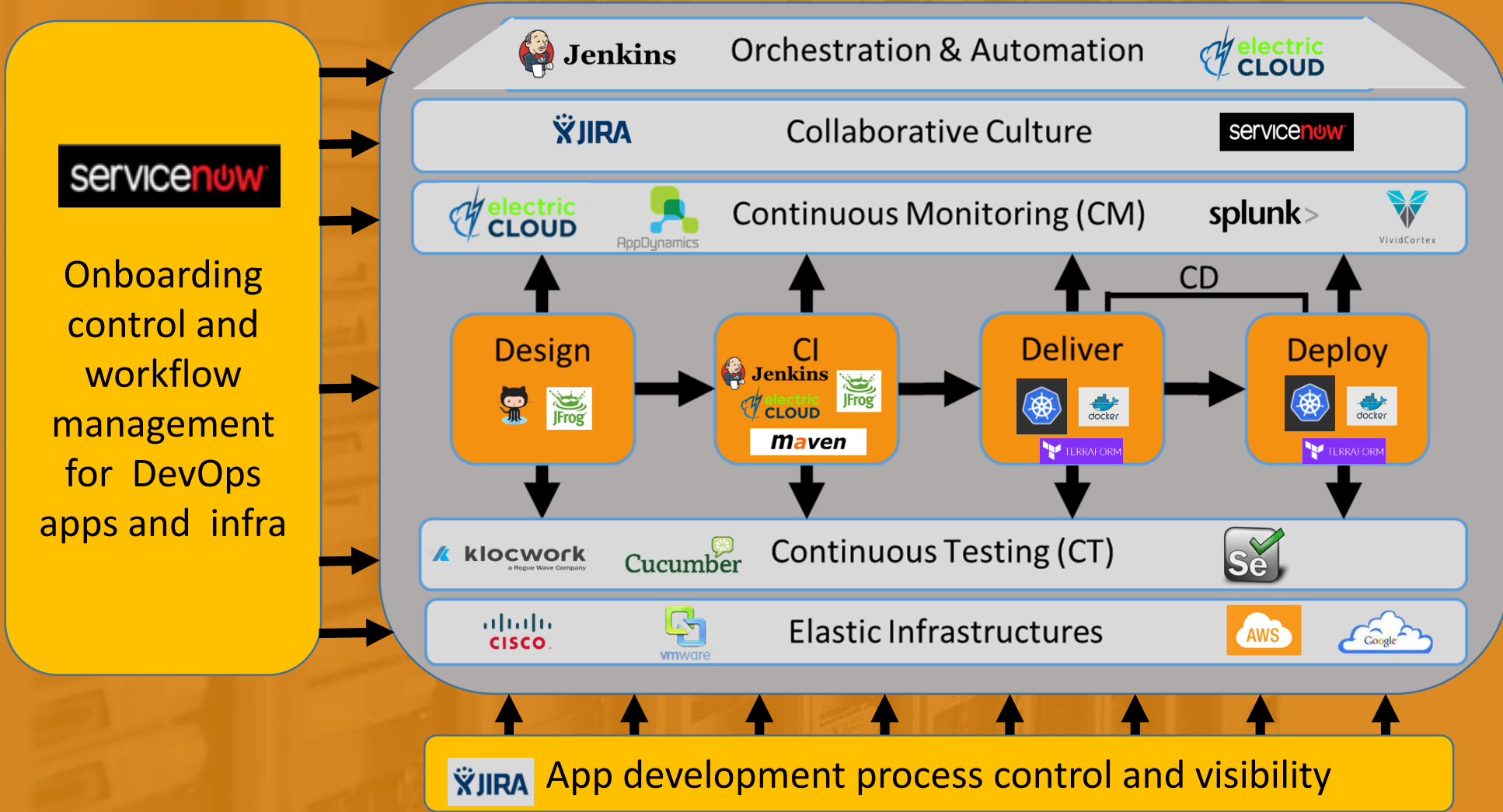
## Satisfaction

Employee frustration

**Collaborative end-to-end workflows**

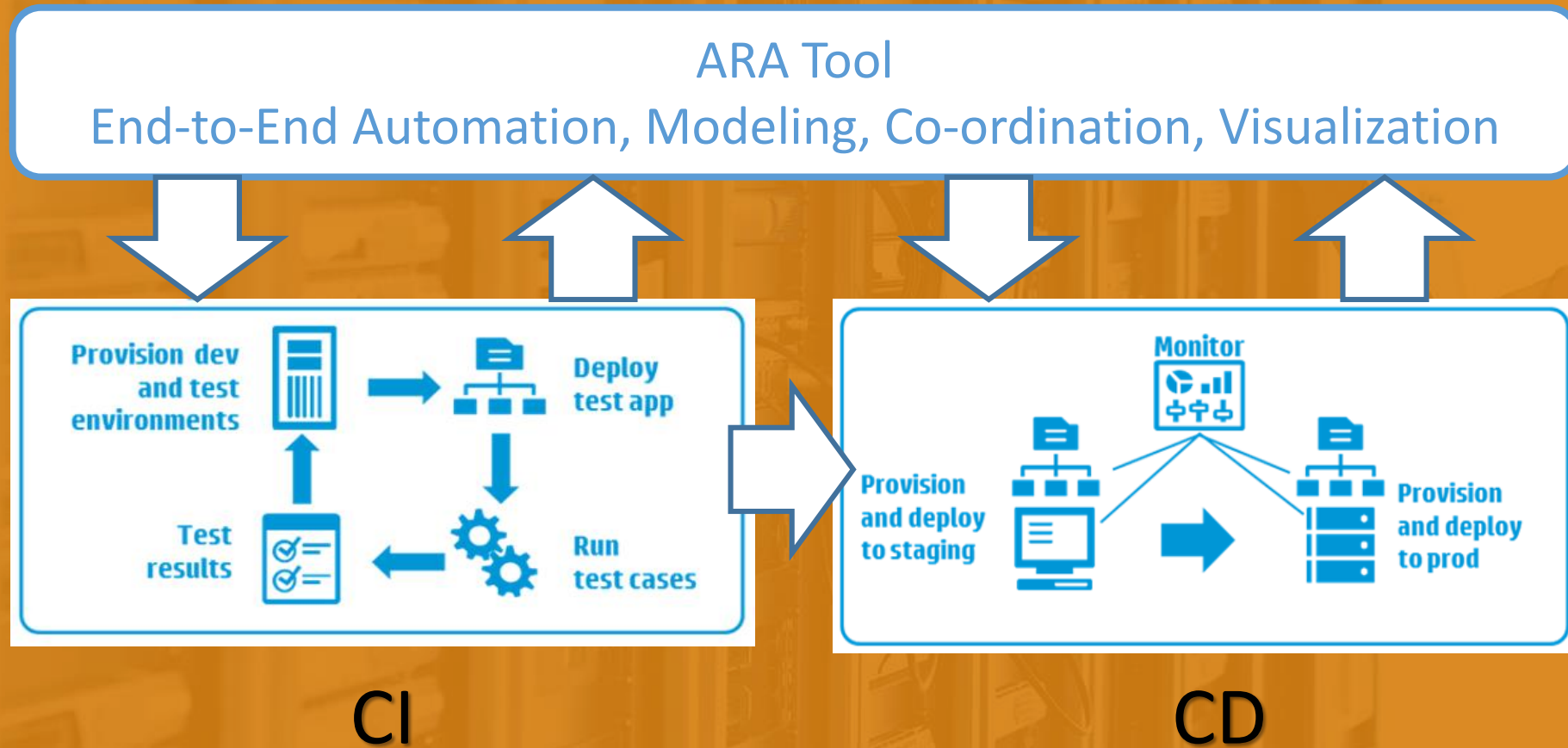


# Control DevOps Technology & Process Evolution



Transformative IT Authority

# Application Release Automation (ARA)



# Application Re-engineering

## Microservices and Containers



**From Monolith to Microservices - a**



**March 2nd, 2017**

Software application development and delivery often involves multiple development, infrastructure and operations teams, each with their own preferred "tools of the trade" for building, testing and deploying code changes



**Marc Hornbeek**  
Principal Consultant - DevOps at Trace3



Marc has over 37 years of experience architecting, designing, developing and managing high-performance solutions for IT infrastructures that are deployed in commercial and government applications globally. Marc has served as executive, senior management and solution architect

<http://webinars.devops.com/monolith-microservices-beyond>



# Integrate DevOps with other IT practices because DevOps is not an Island

## Intelligence Services

- Big Data Analytics
- Service Monitoring
- Security

## Innovation Services

- DevOps



## Infrastructure Services

- ITaaS
- Cloud Migration
- Storage Solutions
- DataCenter Migration

# Thankyou !

Questions?



[www.trace3.com](http://www.trace3.com)  
[mhornbeek@trace3.com](mailto:mhornbeek@trace3.com)  
<https://www.linkedin.com/in/marchornbeek>