goto;
conference

# Please evaluate this talk via the mobile app!

WORKED FINE IN DEV

OPS PROBLEM NOW

# Agile Tribes



Kanban Board · Definition of Done · Point Estimates · Planning Poker · Backlog · Given When - Then · BDD

Lead Time · Definition of Ready · Relative Estimation · Backlog Grooming · Role-Feature · ATDD

Task Board · Personas · Acceptance Test

Scrum of Scrums · Niko Niko Calendar · Burndown Chart · Story Mapping · Continuous Deployment · Ubiquitous Language · Mock Objects

Team Room · Three Questions · Story Splitting · Continuous Integration · Refactoring · TDD

Sign Up · Iterations · Velocity · User Story

Sustainable Pace · Pair Programming · Daily Meeting · Frequent Releases · Collective Ownership · Simple Design · Unit Test

Project Charter · Facilitation · Timebox · Automated Build · Rules of Simplicity · Exploratory Testing

Heartbeat Retrospective · INVEST · 3 C's · Quick Design Session · Usability Testing

Team · Iterative Development · Incremental Development · Version Control · CRC Cards

## Practice Areas

| | | |
|---|---|---|
| Extreme Programming | Scrum | Design |
| Teams | Product Management | Testing |
| Lean | DevOps | Fundamentals |

Laura Klemme
From: Agile Alliance

# What is your cycle time

**TMNS**

Process optimization.

# Cost and value

Invest in the delivery pipeline to decrease the cost of running your process
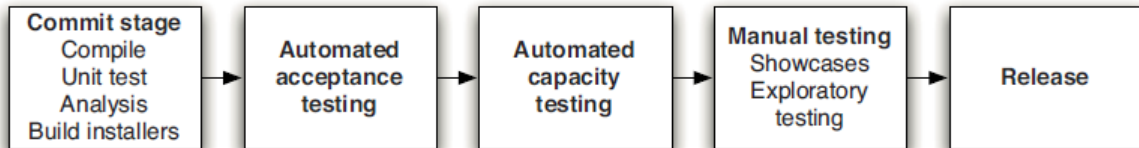
# Software and cars are different

**TMNS**

3. Create Knowledge

4. Defer Commitment

5. Deliver Fast

2. Build Quality in

1. Eliminate Waste

6. Respect People

7. Optimize the whole

# Deployment

# The deployment pipeline as a solution



```
Commit stage        Automated      Automated      Manual testing    Release
Compile             acceptance     capacity       Showcases
Unit test           testing        testing        Exploratory
Analysis                                           testing
Build installers
```
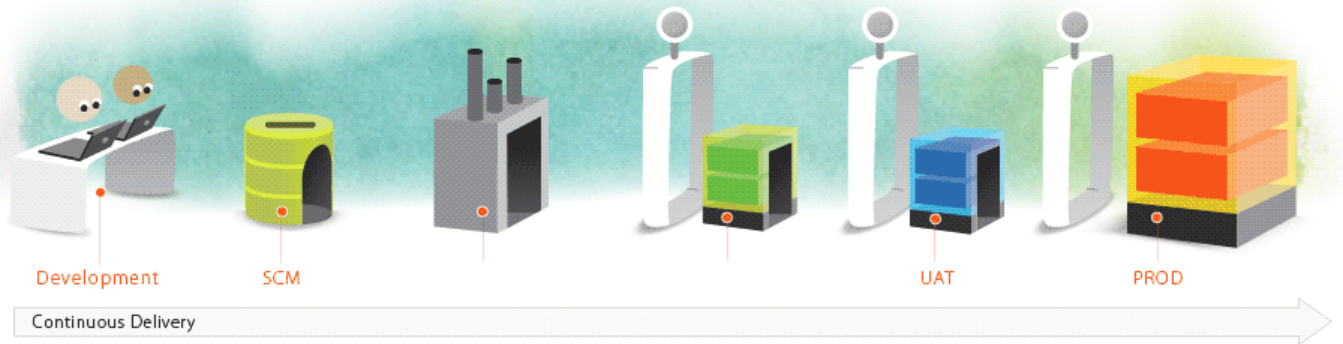
- Every change should trigger the feedback process
- Feedback must be received as soon as possible
- The delivery team must receive the feedback and act on it

Release software **automated** and **frequent**

# Deployment pipeline in detail
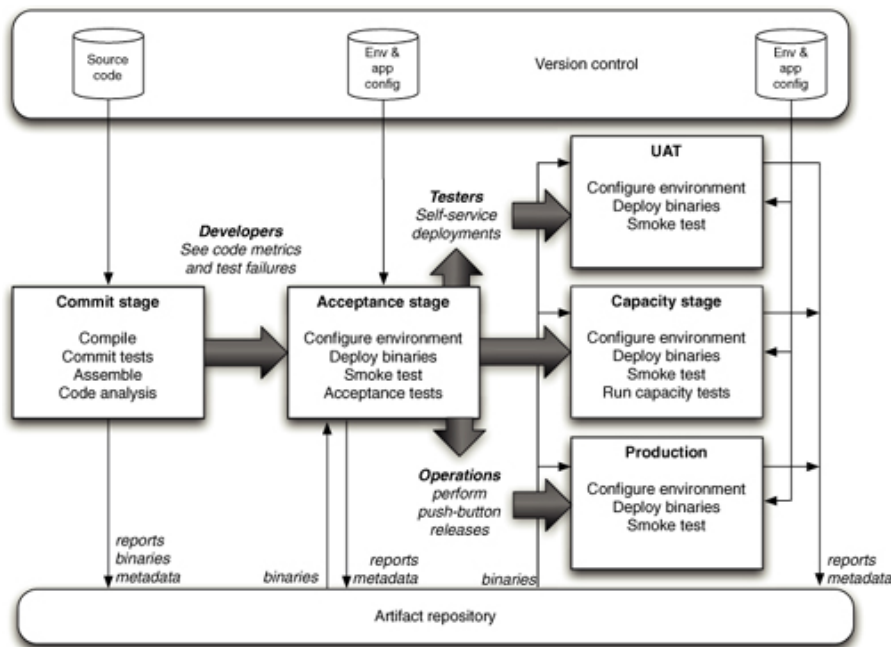


As described by Jez Humble and Dave Farley

Development     SCM                        UAT         PROD

Continuous Delivery

# Cost Reduction Estimate

| Phase cost / Project Type | New projects / CRs | | Upgrade Projects | |
|---|---|---|---|---|
| | **Current** | **Target** | **Current** | **Target** |
| **Architecture** | 2 | 2 | 0 | 0 |
| **Design** | 8 | 8 | 5 | 5 |
| **Development** | 25 | 13 | 10 | 7 |
| **E2ETest** | 20 | 4 | 25 | 5 |
| **UAT Test** | 25 | 16 | 30 | 20 |
| **Performance** | 5 | 4 | 5 | 4 |
| **Go-Live** | 2 | 1 | 15 | 10 |
| **Post PROD** | 3 | 1 | 5 | 4 |
| **Operations** | 5 | 5 | 5 | 5 |
| **PMO** | 5 | 5 | 0 | 0 |
| **Total** | 100 | 65 | 100 | 60 |
| **Estimated Cost Savings** | **30% to 40%** | | **40% to 45%** | |

Note: Automation will be introduced on items in blue.
Items in black will receive reduction through indirect impact

TMNS

Process

**TMNS**

# Do you recognize this?

- Software is deployed manually
- Deployment to a production-like environment only after development is complete
- Manual configuration management of production environments
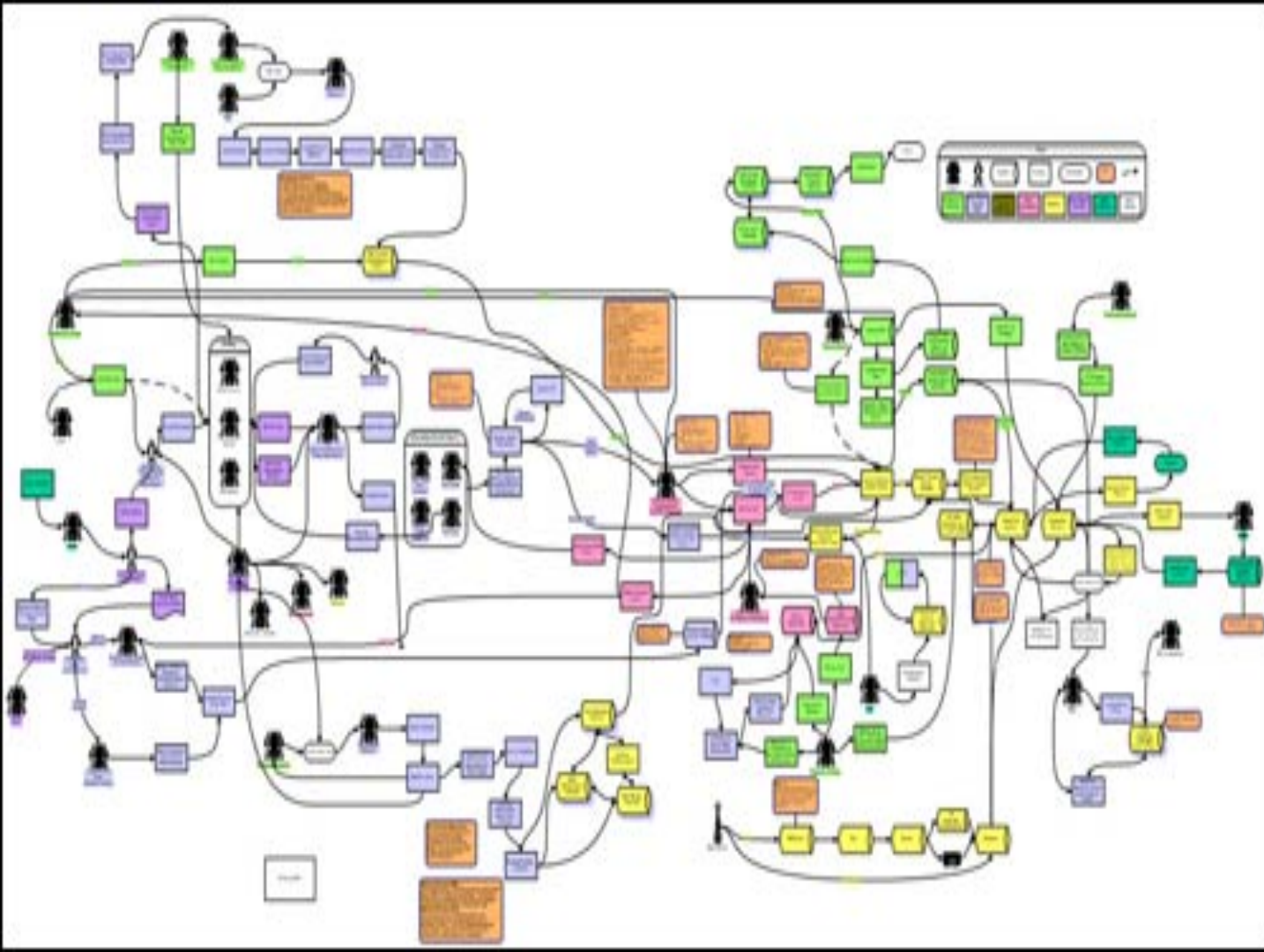
# Principle #7:
# Improve the System /
# See the Whole /
# Optimize the Whole

Teams

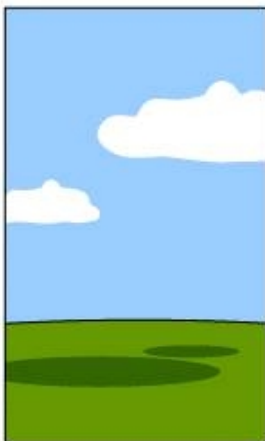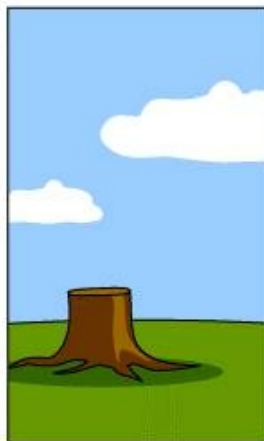| How the customer explained it | How the Project Leader understood it | How the Analyst designed it | How the Programmer wrote it | How the Business Consultant described it |
|---|---|---|---|---|
| How the project was documented | What operations installed | How the customer was billed | How it was supported | What the customer really needed |

# Teams per product not per project

# KEEP CALM

# And

# RESPECT

# THE PEOPLE

Architecture

Decoupled releases
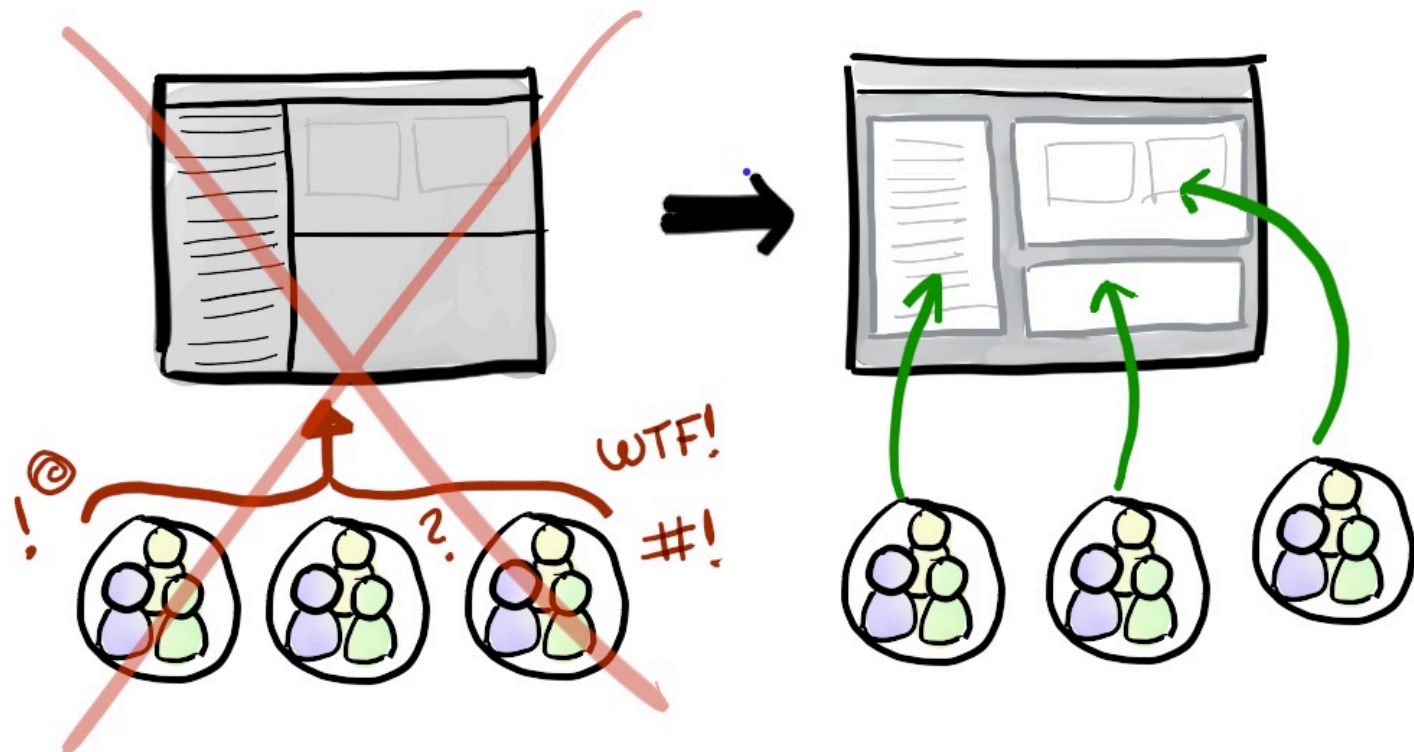
# Tools

# CA Release Automation

# Continuous Delivery Value Chain is Chaotic and Complex



| DEV | QA/TEST | PRE-PROD | PRODUCTION |
|---|---|---|---|
| SCM ⚒ CI ⚒ REP MGR | CONTAINER ⚒ TEST | CONFIG ⚒ CLOUD | SCRIPT ⚒ MANUAL |
| **Code \| Integrate \| Build** | **Functional & UA Testing** | **Performance Testing** | **Deploy to Production** |
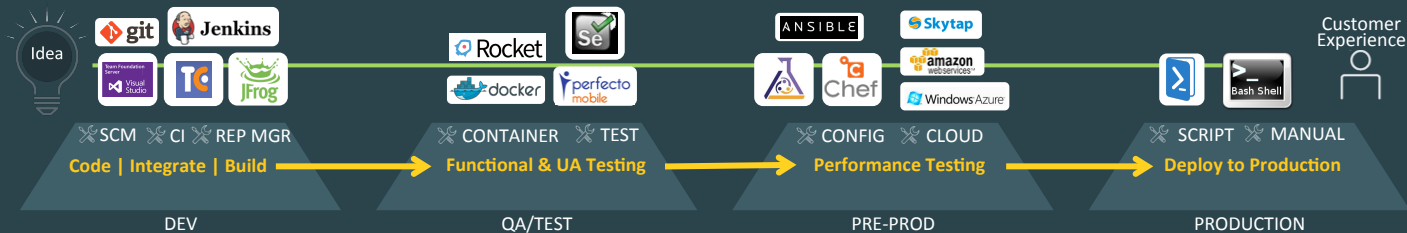
# Traditional "software factory"

| Developers Deadlocked | Constrained Environments | Manual Testing | Too little testing, too late in the cycle | Too many disparate tools | Manual Scripting | Poor customer experience |



**Idea**

Work Time
Wait Time

IDLE TIME      DELAYS      DELAYS      DELAYS

⚒ IDE ⚒ SCM ⚒ CI      ⚒ CONTAINER ⚒ TEST      ⚒ CONFIG ⚒ CLOUD      ⚒ PROVISION

**Code | Integrate | Build** — defect→ **Functional & UA Testing** — defect→ **Performance Testing** — defect→ **Deploy to Production**

RE-WORK      RE-WORK      RE-WORK

DEV      QA/TEST      PRE-PROD      PRODUCTION

Customer Experience

*Slow and inefficient - excessive idle time, delays and re-work*

**ca** technologies

# Continuous Delivery from CA

Developers Deadlocked

Constrained Environments

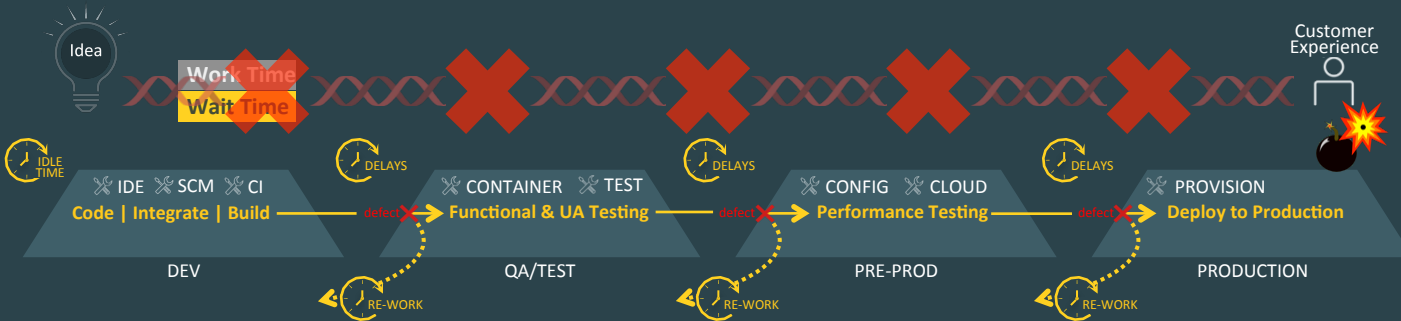Manual Testing

Too little testing, too late in the cycle

Too many disparate tools

Manual Scripting

Poor customer experience

**Parallel Work Streams**

**Quality Acceleration**

**End-to-end Automation**

Idea

Work Time

Customer Experience

IDLE TIME

DELAYS

DELAYS

DELAYS

**Code | Integrate | Build** → **Functional & UA Testing** → **Performance Testing** → **Deploy to Production**

DEV

RE-WORK

QA/TEST

RE-WORK

PRE-PROD

RE-WORK

PRODUCTION

*Tooling for highly automated, end-to-end software pipeline*

ca
technologies
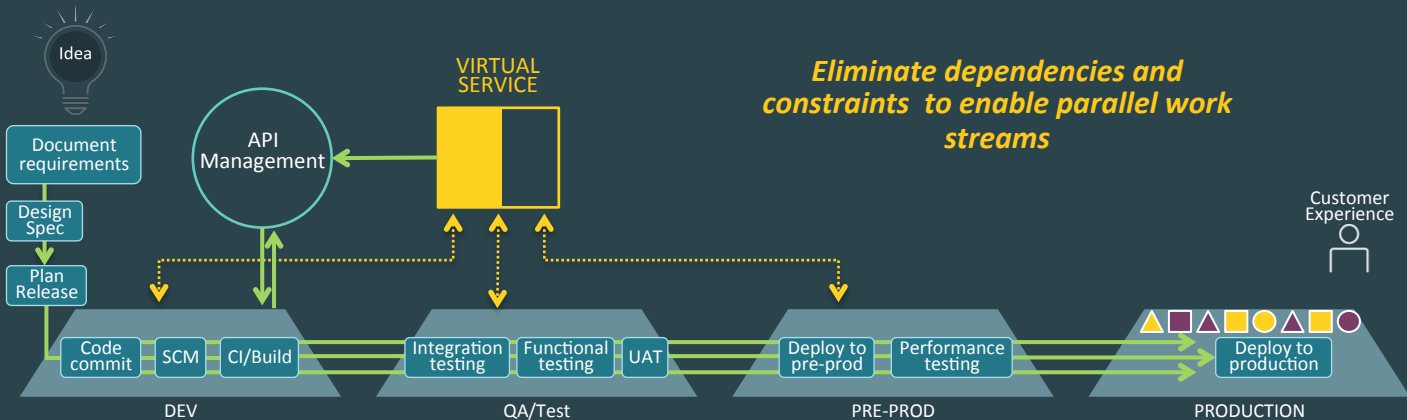
**Quality Acceleration**

| Extract real-world data to provision test environments | Automatically generate test cases and virtual services | Test on any device or platform mobile-to-mainframe |

*Eliminate expensive test /infrastructure costs and time intensive data management*

Idea

Document requirements

Design Spec

Plan Release

VIRTUAL SERVICE

TEST PLAN

real world data

Customer Experience

Code commit — SCM — CI/Build — **DEV**

Integration testing — Functional testing — UAT — **QA/Test**

Deploy to pre-prod — Performance testing — **PRE-PROD**

Deploy to production — **PRODUCTION**

**Find defects here**

**...instead of here**

**← SHIFT LEFT →**

ca technologies
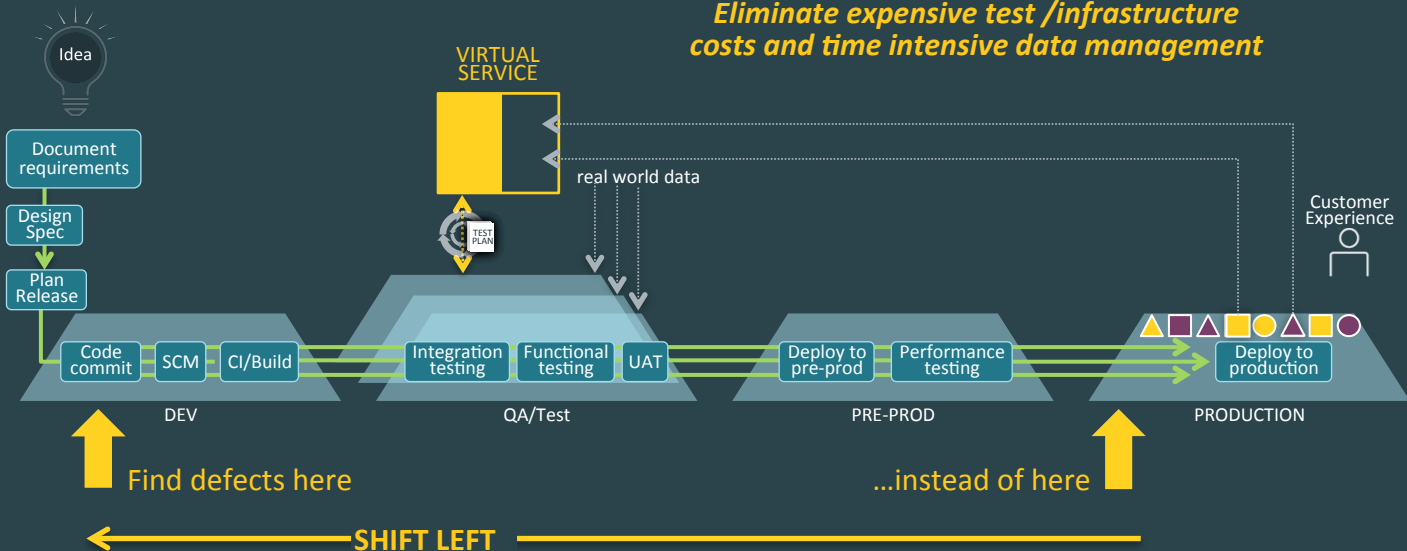
**End-to-end Automation**

Integrate 100+ open source and 3rd party tools

Automate application deployment & promotion

Create continuous feedback loops

Idea

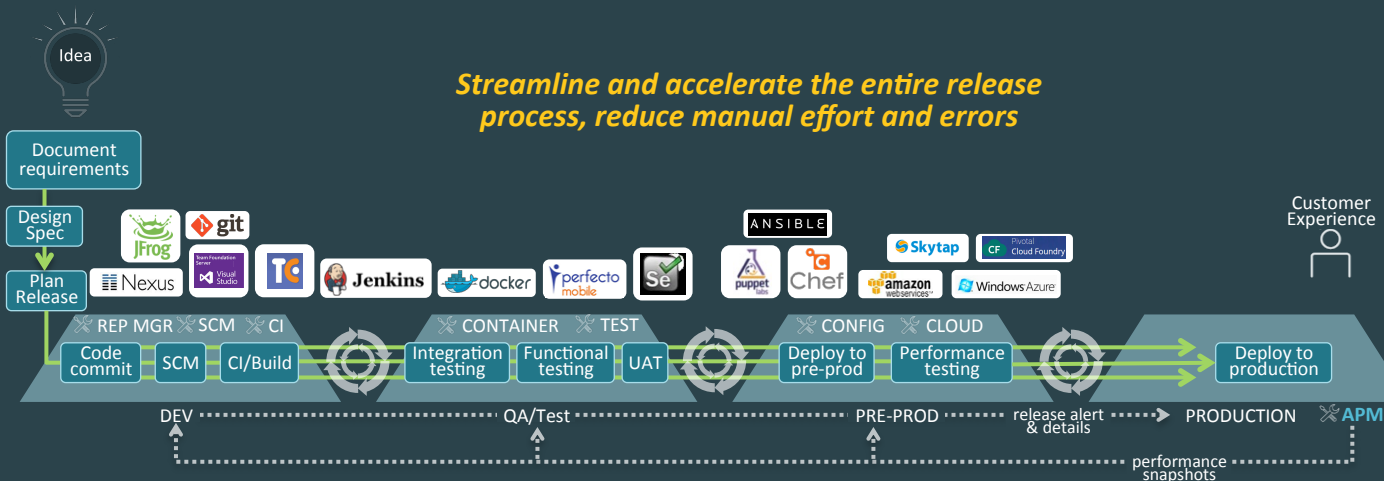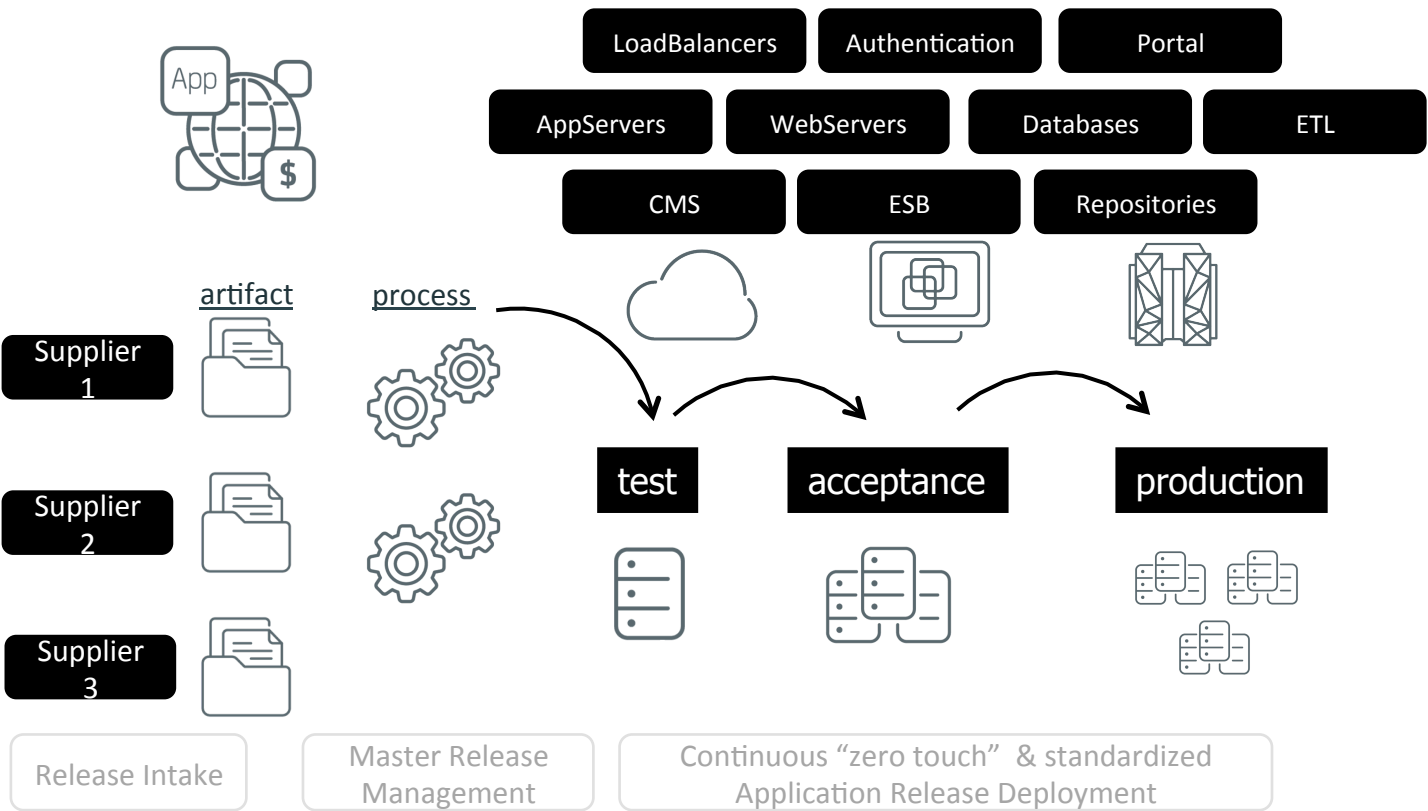*Streamline and accelerate the entire release process, reduce manual effort and errors*

Document requirements

Design Spec

Plan Release

Customer Experience

JFrog · git · Nexus · Team Foundation Server Visual Studio · TC · Jenkins · docker · perfecto mobile · Se

ANSIBLE · puppet labs · Chef · amazon web services · Skytap · CF Pivotal Cloud Foundry · Windows Azure

REP MGR · SCM · CI | CONTAINER · TEST | CONFIG · CLOUD

| Code commit | SCM | CI/Build | | Integration testing | Functional testing | UAT | | Deploy to pre-prod | Performance testing | | Deploy to production |

DEV ........... QA/Test ........... PRE-PROD ........ release alert & details ......> PRODUCTION · APM

performance snapshots

32

© 2015 CA. ALL RIGHTS RESERVED.

ca technologies

App $

LoadBalancers | Authentication | Portal

AppServers | WebServers | Databases | ETL

CMS | ESB | Repositories

**artifact** | **process**

Supplier 1

Supplier 2

Supplier 3

test

acceptance

production

Release Intake

Master Release Management

Continuous "zero touch" & standardized Application Release Deployment

App

$

artifact

prod

**Supplier 1**

**Supplier 2**

**Supplier 3**

**Standardization**
Shared Components – No Scripting - Capture Knowledge

**Agile Transformation**
Increase from 10 to 4000 release deployments/month

**Consistency and Efficiency**
Reduce release failures and manual effort by 90%

**Continuous Delivery Transformation**
Reduce Cycle Time from 13 weeks to 1 week

Release Intake

Master Release Management

Continuous "zero touch" & standardized Application Release Deployment

# CA Continuous Delivery Customer Value

### REGIONAL BANK

- Cut 3 ½ Months from Dev. time
- Cut 65-85% testing cycle costs
- Cut Dev. Env set up time 42 days to 3 days

### GLOBAL TELCO

- Reduced major defects per release
- Saved at least US$ 280K/yr.
- Access to end-to-end environments 90% of the time.

### HEALTHCARE PROVIDER

- 25% efficiency gain
- Shortened cycle by 4 weeks
- Better Testing found 30 fewer bugs - a $600K saving

### MULTINATIONAL GROCER

- Reduced release cycles from 3 days to 20 min.
- Updates to international online grocery app.
- $420K/month saving

### GLOBAL CREDIT CARD

- Nearly 100% uptime for Virtual Services
- Zero defects in UAT
- $319K/year net annual savings estimate
- 15 month ROI

### GOVERNMENT

- Mainframe cost drop 50%-60% expected
- Improved product quality and expedited time-to-market
- Replaced testing environment

### INSURANCE

- Reduced regression cycles by 40%-70%
- Reduced 4 weeks of first release of new solution
- $1.2M efficiency gained in first 2 years after implementation

### HEALTHCARE GROUP

- ROI in 8 Months
- 300% increase in weekly deployments
- 200% increase in number of deployed environments

ca technologies

# Contact



**Gert Jan van Halem**
**Senior Consultant**
gjvhalem@tmns.com
++31 6 42 43 22 29

Please evaluate this talk via the mobile app!