CONTINUOUS DELIVERY

CONTINUOUS DELIVERY (CD) is a design practice used in software development to automate and improve the process of software delivery. Techniques such as automated testing and continuous integration allow software to be developed to a high standard that is easily packaged and deployed to test environments. The result is the ability to rapidly, reliably and repeatedly push out enhancements and bug fixes to customers at low risk and with minimal manual overhead.

Benefits of CD

Rapid Development
More frequent testing and deployment delivers new features in a matter of hours rather than days or weeks.

Improved Quality
Automated QA and best-practices reduce mistakes due to manual testing and configuration.

Operational Efficiency
Frequent development and release schedule leads to a smoother effort with less risk. Increased responsiveness leads to more favorable customer engagement.
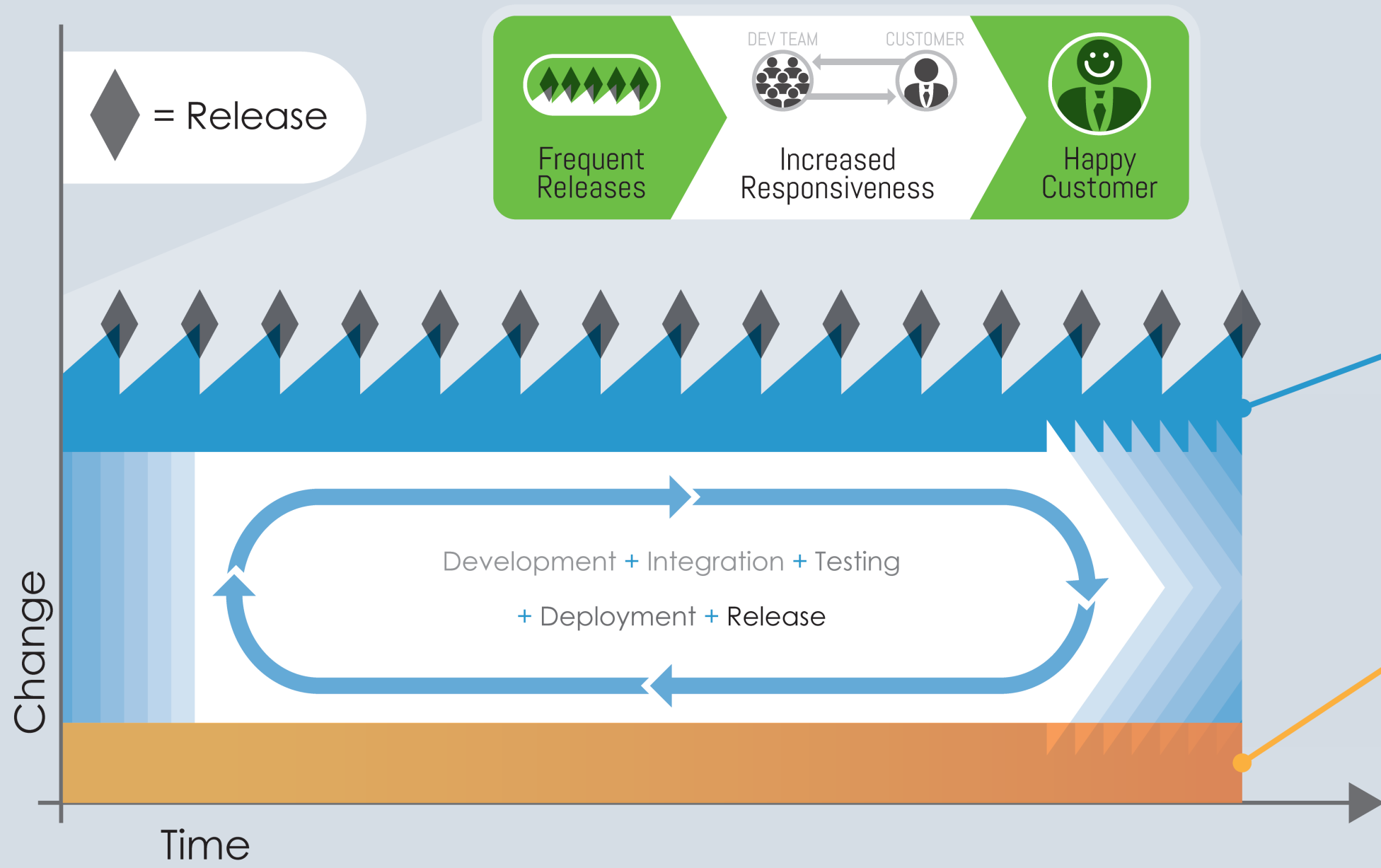
Increased Business Focus
High level of automation creates the ability to focus on core business functions and processes.

CONTINUOUS Delivery   VS   Traditional Incremental Development

◆ = Release

Frequent Releases → Increased Responsiveness → Happy Customer

DEV TEAM — CUSTOMER

Development + Integration + Testing + Deployment + **Release**

Effort

Risk

Change / Time

◆ = Release

Development | Integration | Testing | Deployment | Release | Development | Integration | Testing | Deployment | Release | Development | Integration | Testing

Change / Time

Continuous Delivery's frequent release events generate **CONSISTENT and SMOOTH EFFORT** + **LOW RISK**

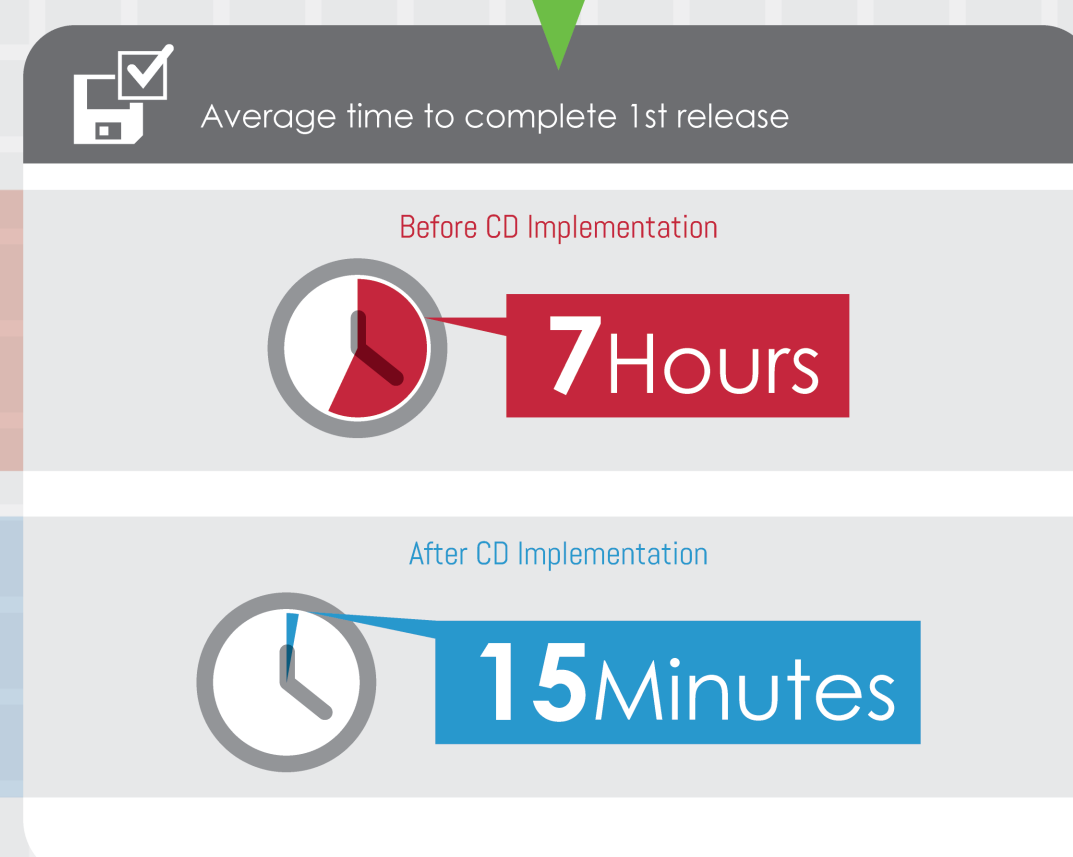**Traditional Incremental Development's** Infrequent releases lead to **INCONSISTENT and INCREASED EFFORT** + **HIGH RISK**

1901 Group experienced the following benefits (which were then passed on to a large government client) after implementing **CONTINUOUS Delivery**
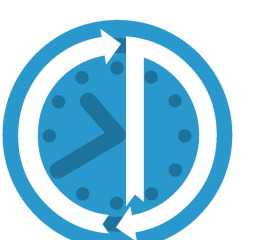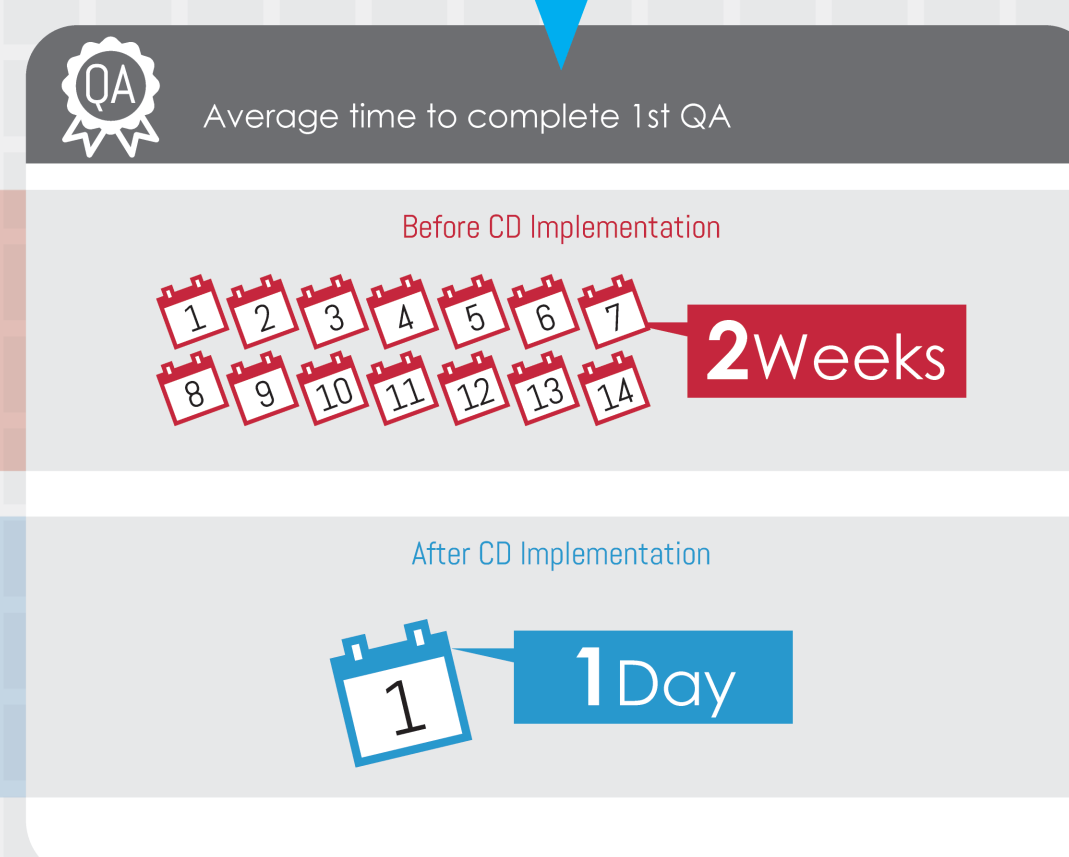
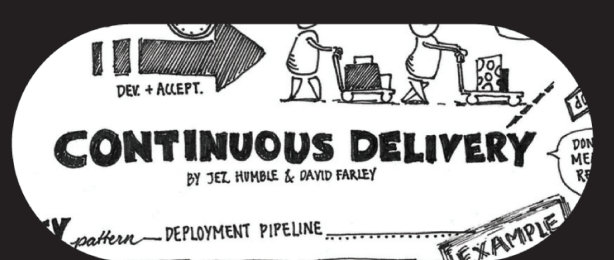### OVER 300% FASTER
completion of user stories

Time to complete 2 user stories

Before CD Implementation
2 WEEKS | 2 WEEKS | 2 WEEKS | 2 WEEKS → **8 Weeks**
4 Sprint Cycles (8 Weeks)

After CD Implementation
2 WEEKS → **2 Weeks**
1 Sprint Cycle (2 Weeks)

OVER 25X FASTER
release schedule

Average time to complete 1st release

Before CD Implementation
7 Hours

After CD Implementation
15 Minutes

OVER 14X FASTER
Quality Assurance

Average time to complete 1st QA

Before CD Implementation
1 2 3 4 5 6 7 8 9 10 11 12 13 14 → **2 Weeks**

After CD Implementation
1 → **1 Day**

HOW CD WORKS!

CONTINUOUS DELIVERY
DEPLOYMENT PIPELINE

For more detail on how Continuous Delivery works. Scan the QR code or visit the site linked below where you will find a wonderfully detailed CD visualization by Nhan Ngo, a QA engineer at Spotify.

http://bit.ly/1AaMoBE