ThinkingLabs

# Continuous Delivery is more than just Tooling, It's a Culture

ITAKE Unconference 2016
Thierry de Pauw, @tdpauw

ThinkingLabs

When speaking about Continuous Delivery what do most people think about ?

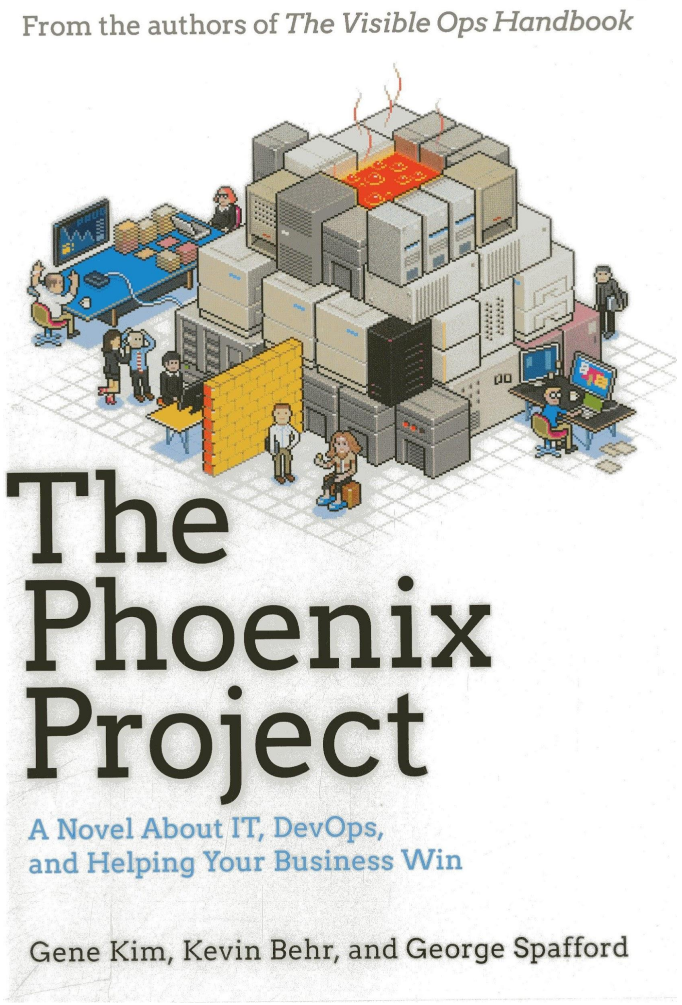
Continuous Inspection
sonarqube
JSLint
JSHint
Pylint
Star your Python code!
Maven™
Gradle
<APACHE ANT>
buildr
go
sbt
Nexus
archiva
JFrog Artifactory
Cabal
mercurial
Spinnaker
docker
behat
TestNG
JUnit
RAKE
SCONS
PAVER
SUBVERSION
GRUNT
Gulp
webpack
MODULE BUNDLER
CHEF
ANSIBLE
SALTSTACK
puppet
mocha

**Continuous Delivery** is NOT just tooling.
It is a **mindset** to adopt.

The goal of Continuous Delivery is

to **sustainably minimise** the **lead time** to create positive **business impact**.

ThinkingLabs



From the authors of *The Visible Ops Handbook*

The Phoenix Project

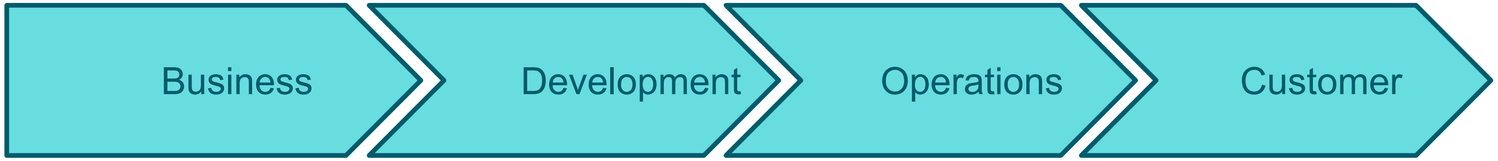
A Novel About IT, DevOps, and Helping Your Business Win

Gene Kim, Kevin Behr, and George Spafford

# The Phoenix Project

## The Principle of the 3 Ways

Strategies for
improving Software Delivery



# #1 - From Left to Right

Maximise the flow of the software delivery process.

**Repeatable** and **reliable** process for delivering software.

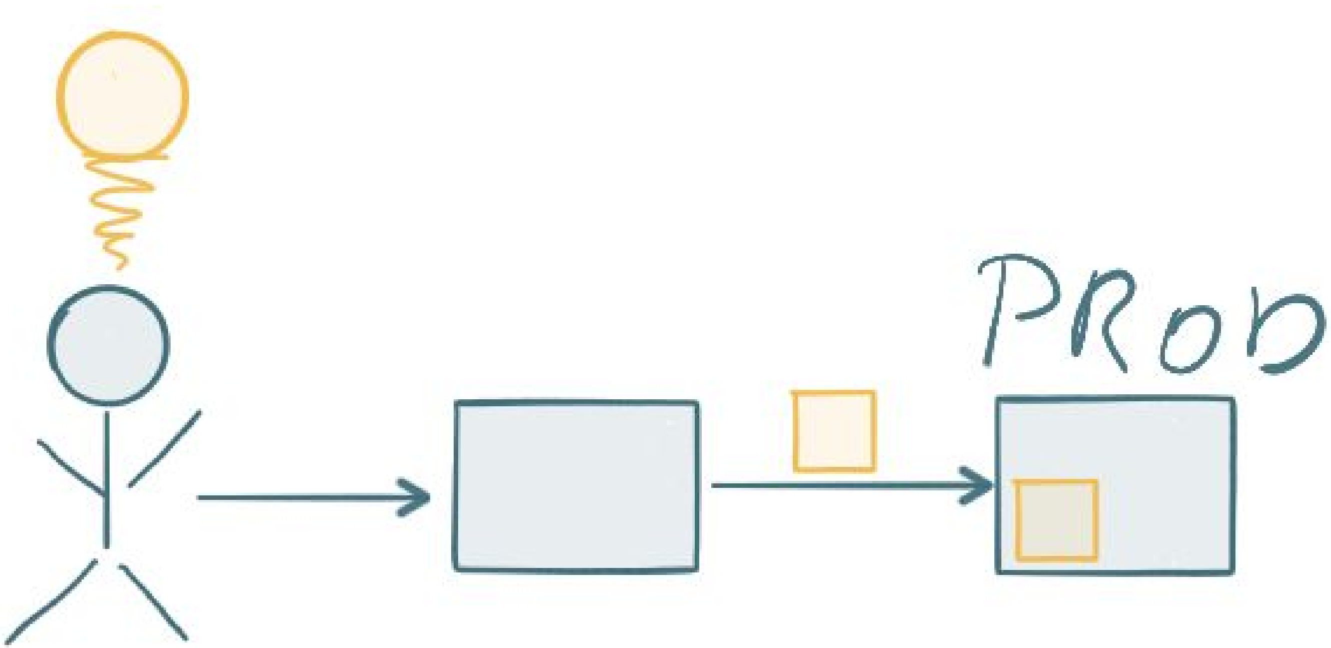
This means

- **automate** the hell out of it,
- and everything under **version control**.

Encourage **collaboration**.

# Create an empowered product teams.

Aim for a **single piece flow**.



Bigger batches
= bigger lead times
= money stuck in the system

Smaller batches = reduce risks

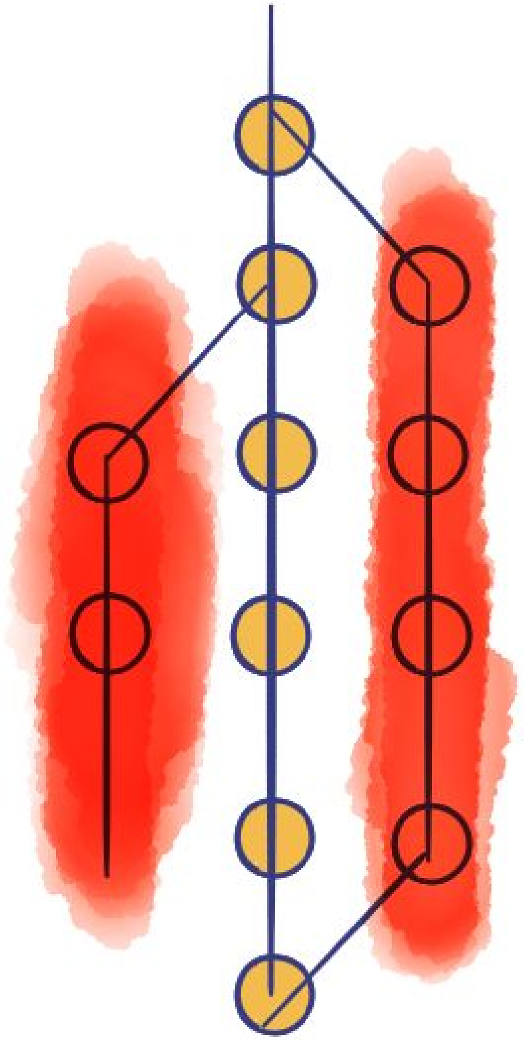
- Release process is more often practiced
- Finding out what went wrong is easier
- Rolling back small changes is easier

Your application is always in a
**releasable state on main line**.

Ready for release does not mean actually releasing.
Software with unfinished work can still be releasable.
Every build is a release.

## Trunk Based Development

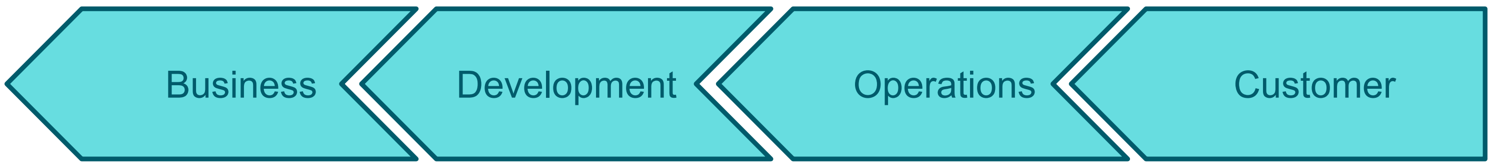
- Make changes incrementally
- Hide unfinished work
- Branch by Abstraction
- Feature Toggle as a last resort



*Feature Branching is "Doing Agile"*

*Continuous Delivery and Trunk Based Development is "Being Agile"*

**-- Dave Hounslow (@thinkfoo), You Are What You Eat, Pipeline Conf 2016**


Business

Development

Operations

Customer

# #2 - From Right to Left

Creating feedback loops at all stages of the pipeline.

Whenever something fails,
**stop the production line**.

**Build quality in**to the code from the start,
not testing it in later.

2 kinds of inspections

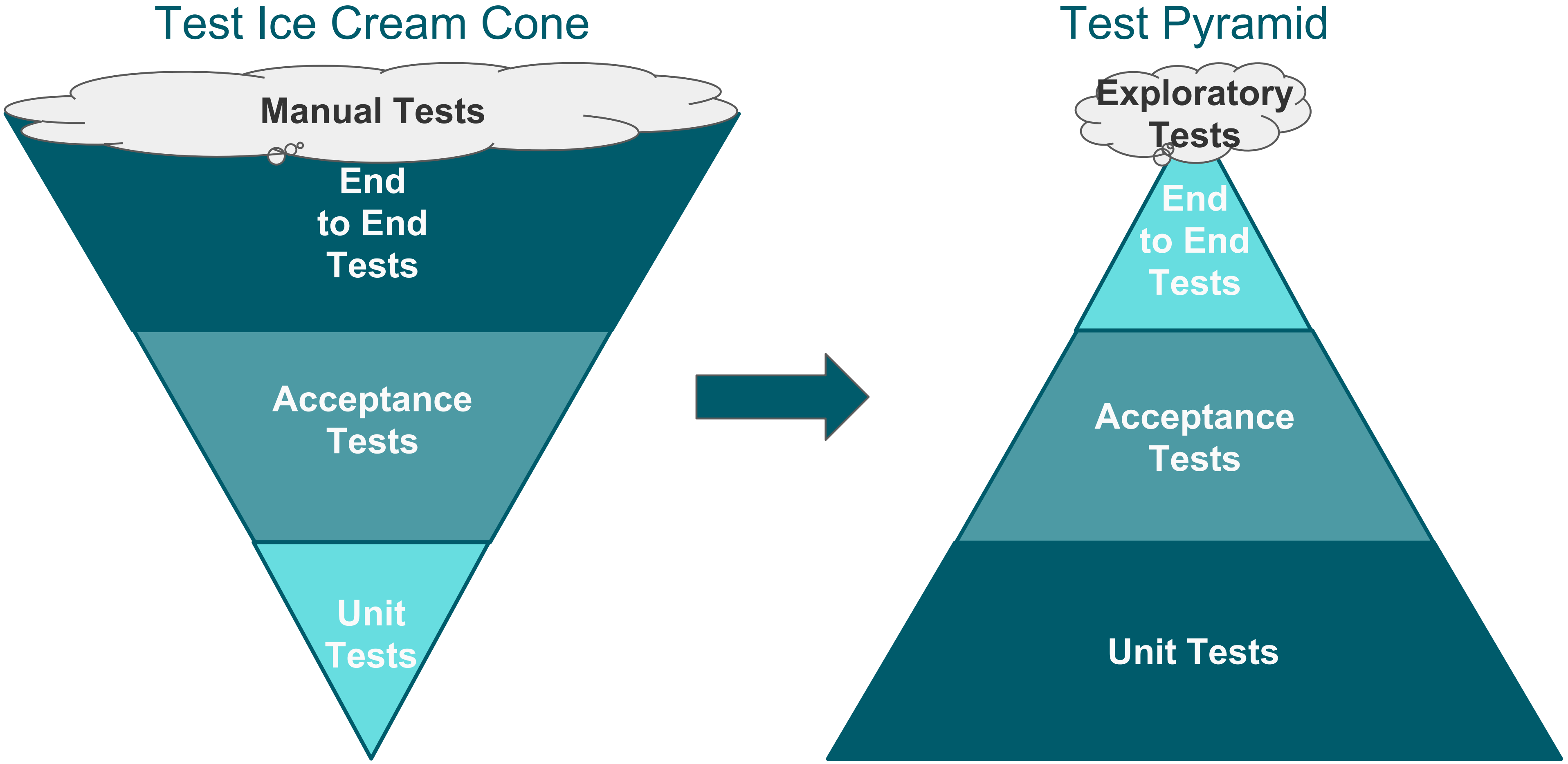
- Inspection after the defect occurred
- Inspection to prevent defects

**Testing is not a phase**,
it is part of your development process.

Everybody is responsible for the quality.

ThinkingLabs

# Continuous Testing

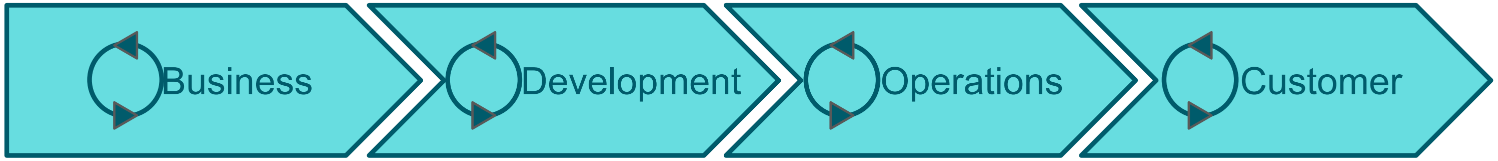


@tdpauw

*Source: Stephen Smith, Always Agile Consulting*

thinkingslabs.io

Continuous Delivery works because it has this **crazy focus on Quality**.



Business

Development

Operations

Customer

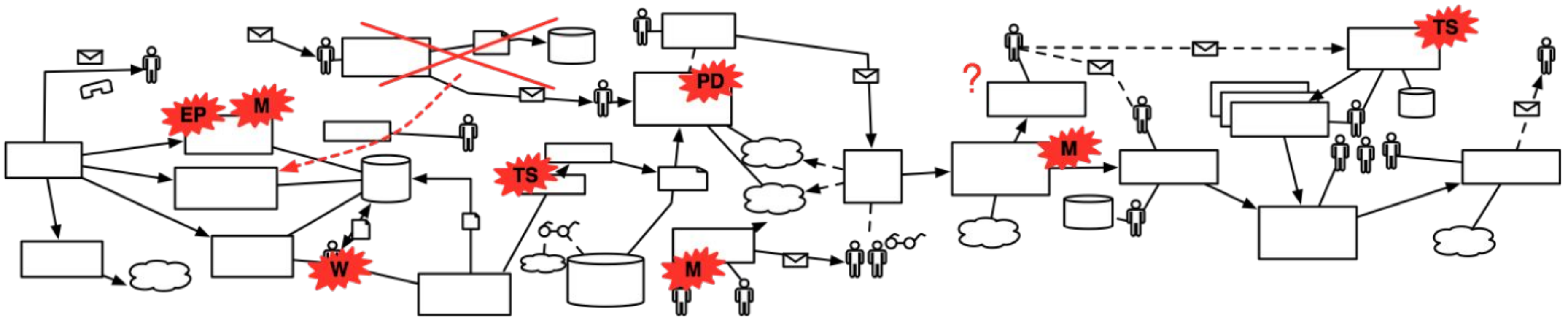
# #3 - Creating a culture of improvement

Your delivery pipeline will not come over one night.

Instead have regular **retrospectives** with everybody involved in the software delivery process.

# Make use of the **Value Stream Map**.

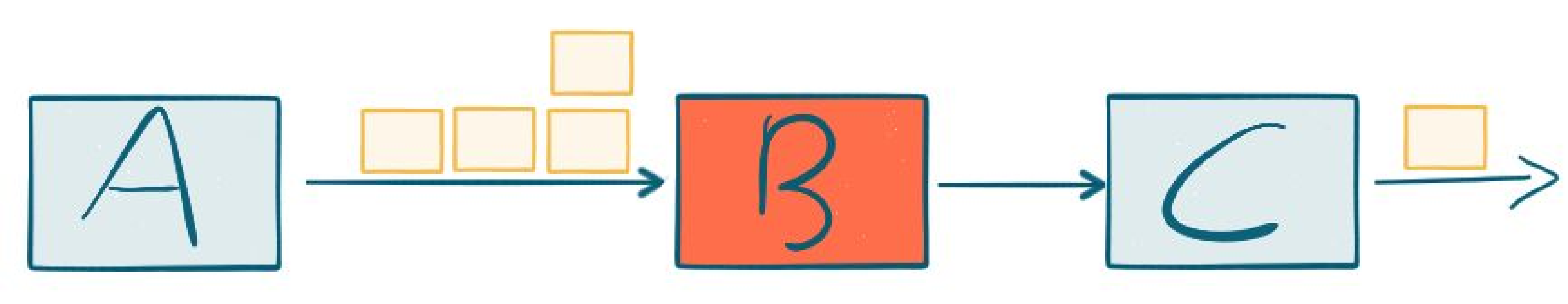
Identify wastes and bottlenecks.



*Source: Damon Edwards, DTO Solutions*

*"Spending time optimising anything other than the critical resource is an illusion."*

**-- W. Edwards Deming**



Repetition and practice leads to mastery.

**If it hurts, do it more often.**
**Bring the pain forward.**

# Conclusion

Optimising for throughput and for speed
gets you to **lower costs**
and **higher quality**.

Where in the past optimising for cost and
quality got you neither.

Continuous Delivery is
**not easy nor cheap** !

It is a cultural change.

# Hello, I am Thierry de Pauw, @tdpauw

Full Stack Software Engineer

Agile Technical Coach

http://thinkinglabs.io