

# ThoughtWorks®

*Getting company buy-in for*  
**Implementing CD**



# Introduction

---

**VLADIMIR SNEBLIC**

**ThoughtWorks®**

<http://thoughtworks-studios.com>  
@vsneblic

# CD is an umbrella term

---

## Continuous Delivery Definition

Continuous delivery is an approach to delivering software that reduces the cost, time, and risk of delivering incremental changes to users. You're doing continuous delivery when:

- Your software is releasable throughout its lifecycle;
- Your team prioritizes keeping the software releasable over working on new features;
- Anybody can get fast, automated feedback on the production readiness of your systems any time somebody makes a change to them;
- You can perform push-button releases of any version of the software on demand.

# Real world Example #1

---

- \* “They don't need a deployment pipeline, they need to talk to each other much more” - Client employee leaving the project

## Real world Example #2

---

\* CD stories were around 25% of scope

**ThoughtWorks®**

*Selling CD to the*  
**Development Team**



Recapture the start-up spirit

---

**We need an engaged team.**

# Recapture the start-up spirit

- \* Work as a single unit across functions



# Recapture the start-up spirit

- \* Team responsible for project success

Increase confidence in what we're building

**In order to go faster we need better and faster feedback.**

Increase confidence in what we're building

\* Automate almost everything

# Increase confidence in what we're building

- \* Need high quality tests at all levels (unit, integration, acceptance)

# Increase confidence in what we're building

- \* Tests are owned by the whole team (not just QAs)

# Increase confidence in what we're building

- \* Build system executes tests on every check in

# Increase confidence in what we're building

- \* Build failures are fixed as a matter of priority

Validate production readiness early and often

**Late integration is the cause of many late deliveries and failed projects.**



# Validate production readiness early and often

- \* Everyone should develop on the Mainline (trunk)

# Validate production readiness early and often

- \* Use feature toggling instead of feature branching

# Validate production readiness early and often

- \* Test system integration in a production like environment early and often

# ThoughtWorks®

*Selling CD to the*  
**Operations & Central Functions**

---

Be part of the delivery team

---

**The only way we can insure we build a quality product, quickly, is to work together.**

Be part of the delivery team

\* Central functions are represented in the delivery team

# Be part of the delivery team

\* Knowledge and skills are cross-pollinated

# Be part of the delivery team

\* Aligned goals



The team is responsible for the delivery

\* “You build it, you run it” - Werner Vogels (Amazon CTO)

# Repeatable and reliable deployments

---

**We need to reduce the cost and increase reliability of deployments.**

# Repeatable and reliable deployments

---

- \* Review your current deployment process

# Repeatable and reliable deployments

---

- \* Automate almost everything (e.g. no “snowflakes”)

# Repeatable and reliable deployments

---

- \* Use the same process in all environments

# Repeatable and reliable deployments

\* Version control everything

# Real world Example #3

---

\* Dev teams using the same tools as operations

# ThoughtWorks®

*Selling CD to the*  
**Business**





# Putting the business into the driver's seat

---

**In order to retain/gain the role of market leader we need to innovate and experiment faster than our competitors.**

# Putting the business into the driver's seat

---

- \* Introduce concept of MVPs (minimum viable products)

# Putting the business into the driver's seat

- \* Frequent delivery to real users

# Putting the business into the driver's seat

---

- \* Use customer's feedback to reprioritise

# Putting the business into the driver's seat

---

\* Don't be afraid to "pivot"

# Putting the business into the driver's seat

---

\* The art of NOT doing things

# ThoughtWorks®

*Selling CD to the*  
**Management**



IT needs to regain the role of trusted advisor

**Our IT organisation needs to be a differentiator and an enabler for the organisation**



IT needs to regain the role of trusted advisor

\* Focus on business value

IT needs to regain the role of trusted advisor

\* Business stakeholders are an integral part of the delivery team

IT needs to regain the role of trusted advisor

\* IT playing the role of trusted advisor

# Organise around delivery teams

**A single team with a single focus has a much higher chance of success.**

# Organise around delivery teams

- \* Form service teams around a single goal or responsibility

# Organise around delivery teams

\* Focus on adding business value

# Organise around delivery teams

\* Scale by building more single focus teams

# Organise around delivery teams

- \* Every team needs well defined customers



Visibility instead of rigid control

**Simple, transparent, semi-automated processes will give us a faster more compliant platform.**

# Visibility instead of rigid control

- \* Reduce the number complex, manual processes

# Visibility instead of rigid control

\* Increase visibility and accountability for actions

# Visibility instead of rigid control

- \* Understand the true requirements from compliance/regulation

# Conclusion

---

\* Gaining buy-in is the hard part

# Conclusion

---

\* Focus on most painful things

# Conclusion

---

\* It helps if you have a plan

# Conclusion

---

\* Easier when the message comes from outside



# The inconvenient truth

---

- \* Sometimes people have to leave before change can happen

# The inconvenient truth

---

\* If all else fails use the power of chocolate