PRIOCEPT

Culture Shock

Winning People to DevOps

Matthew Skelton, Priocept Ltd.

DevOps Days Europe 2010, Hamburg

Agenda

- How can we communicate the effectiveness of DevOps?
- What metaphors and examples help?
- What kind of people should we hire?

Who?

- Matthew Skelton, Technical Director at Priocept, based in London, UK
  - Control engineering background – CEng/Eur.Ing.
  - Thinking and doing elements of “DevOps” for at least 5 years
- Highly available, content-rich internet systems using .Net and Java
  - Development teams also responsible for designing and supporting:
    - Build, deployment, environments, diagnostics, reporting
    - 3rd line support – based on ITIL best practices
- London Stock Exchange www.londonstockexchange.com (2004+):
  - Weekly deployments of new features, including e-Commerce
  - Huge traffic spikes

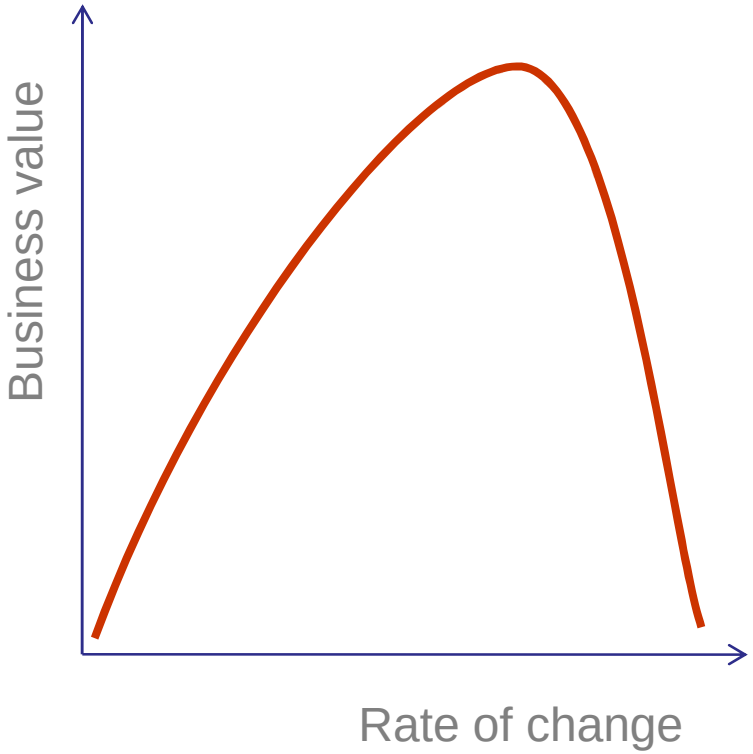

Custom Development Solutions
SOA and Business Process


Why DevOps?

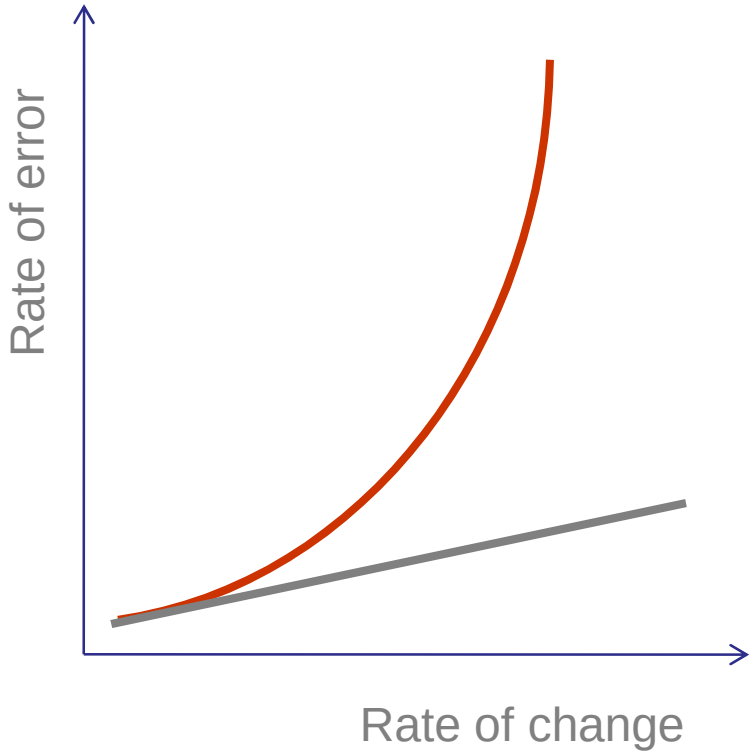
- Applicability – any large, frequently-changing software system
- Target is much broader than SaaS (revenue model)
- Many-user systems; internet-based or internal
- As much a *change in attitude* as tools & techniques
- Respond rapidly and reliably to changing business requirements

Why DevOps?

- Change vs. Value



- Change vs. Errors



Why DevOps?

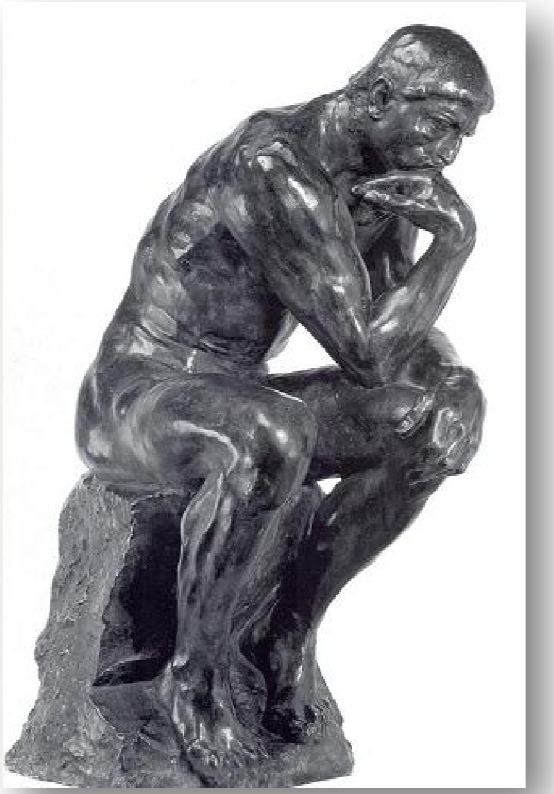
- Deliver more value to the organisation, without increasing major errors or failures
- What does the business care about?
  - Next product version
  - Matching or exceeding the competition
  - Satisfying customers
  - First-to-market
- Reliable, frequent changes to software services

Metaphors – Child and Hoop

essentialvermeer.com

- A perfectly round circle is neither necessary nor sufficient
- Constant input or adjustments needed
- The maker of the hoop knows its faults and quirks
- The maker is thus best-placed to keep it rolling

Art – Sculpture



Rodin

- Expert design and craftsmanship
- Unique
- No plans for changing it
- Difficult and costly to change later

→ Pre-DevOps software systems

Art – Live Performance

www.partsuspended.com / Matthew Skelton

- Not limited to a single pose
- Repeatable - daily
- Please the audience
- Requires orchestration / coordination

→ DevOps-driven software services

Auto Engineering – Concept Car

blogs.cars.com

- Ideal design
- Unique
- No plans for changing parts

→ Pre-DevOps software systems

Auto Engineering – Racing Car

racing.priocept.com

↑ My boss!

- Pragmatic design
- Replaceable components
- Pit-lane repairs
- Parts changed every ~30 minutes

→ DevOps-driven software services

Shock #1

- Change. Is. Good.
- The system will change
- The system will never be “finished”
- Aiming for perfection is asking for failure
- “Perpetual beta”

Shock #2

- Operations. Can. Improve. Development.
- Experience of diagnosing production problems leads to better software
- “Design for Diagnosis” & “Design for Failure”
- Culture change needed in Dev teams
  - Mutually responsible, mutually supportive
- “Dev > Ops” is dangerous and simply incorrect

Shock #3

- Failures. Are. Acceptable.
- Roll-back vs. roll-forward
- Failures as an opportunity to learn about the system
- Confidence provided by log and statistic data
- “Evidence-based testing” – cf. medicine

SLAs, ITIL and Change

- SLAs often work *against* change
- ITIL done badly means risk-averse
- Concept of a *Change Level Agreement* (CLA)?
  - Up to 5%, 10%, 15% of the system to change
  - Per day, week, month
- SLA + CLA to provide a stability-agility metric
- Admits the need for regular, reliable change

What Skills Do We Need?

- Generalising Specialists (*Scott W. Ambler*)
  - http://www.agilemodeling.com/essays/generalizingSpecialists.htm
- No big “software guru” or “sysadmin” egos!
- No more technical silos
- Excellent communications:
  - Within team
  - With other teams
  - With the client
  - Across different spoken (and computer) languages
- Willingness to work with the best technologies, whatever they are

What Skills Do We Need? (2)

- Understand and expect hardware and environmental limitations
- Bandwidth, read/write speed, storage limits, inertia, friction, etc.
- Developers often assume that software will run on “ideal” machines
- The most beautiful software is useless if such limits are forgotten
- The “cloud” does *not* avoid these limits – only makes them
  - Higher (Petabytes instead of Terabytes)
  - Softer (elastic up to a certain threshold, but not beyond)

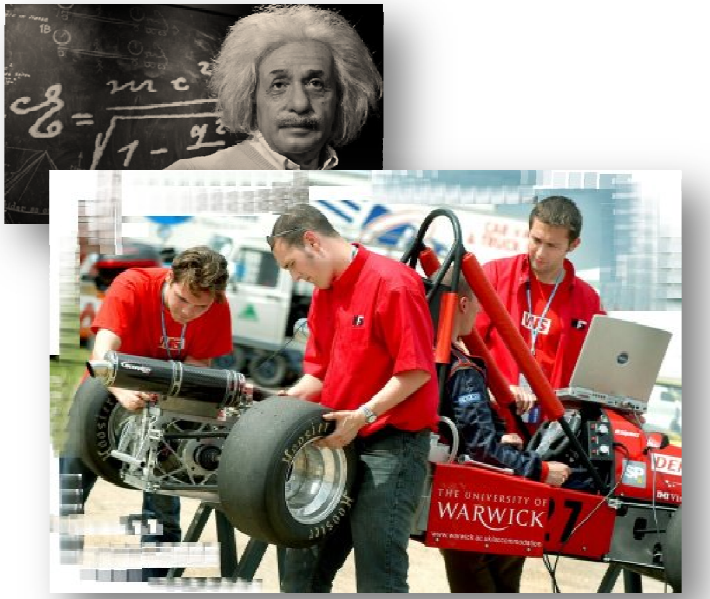
cocoatech.com

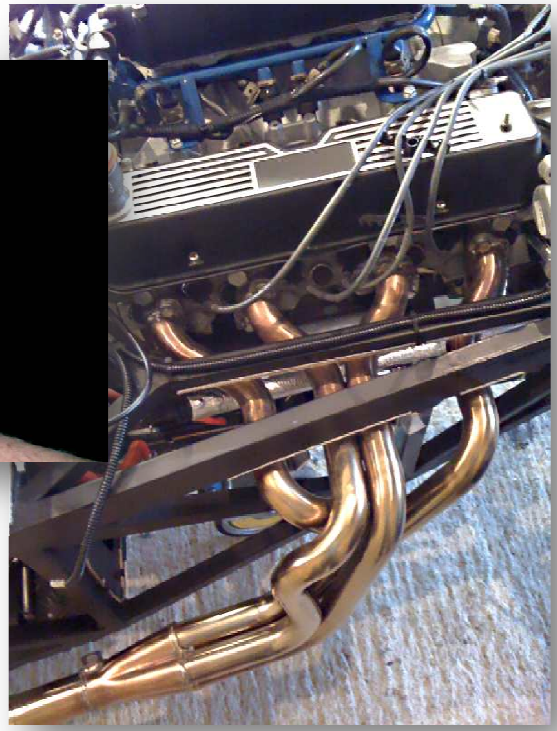
VS.

register.co.uk - ventblockers

Who Should We Hire?

- Mechanical Engineers, Electrical Engineers, Control Engineers
  - With software training and experience
- The occasional mathematician or “pure” software developer ☺





businessweek.com, warwick.ac.uk

Summary

1. DevOps allows reliable, frequent changes to software services

2. Think repeatable performances, not one-off wonders

3. Hire mechanical engineers, not mathematicians!

PRIOCEPT

- Thank you for listening

- Thanks also to:
  - London DevOps group (http://groups.google.com/group/london-devops/)
  - BCS London (http://www.bcs.org/)
  - Rob Thatcher

- matthew.skelton@gmail.com

- Priocept:
  - http://www.priocept.com/
  - @Priocept