Legacy systems, critical systems and DevOps

Stuart Rance
Consultant, trainer and author
IT service management and information security
@StuartRance

optimal
Service Management Ltd

Agenda

- What are legacy systems and critical systems
- Bimodal and multi-speed approaches
- What is DevOps anyway?
- Which DevOps ideas can be used with legacy or critical systems?
- Summary
- What next?

What are legacy systems and critical systems?

Legacy systems typically have

- Traditional waterfall software development
- Many manual steps needed for deployment
- Poor test environments
- Limited skills availability
- Lots of technical debt

Critical systems typically have

- High security requirements
  - Confidentiality
  - Integrity
  - Availability
- Compliance requirements
  - Legal
  - Regulatory

Bimodal approach

- "Everyone knows" you can't use DevOps with legacy and critical systems
- Bimodal (from Gartner)
  - Mode 1: "Doing things right" - traditional, scalable, efficient, reliable, high security, high stability, systems of record
  - Mode 2: "Doing things fast" - agile, fast-changing, competitive, systems of engagement
- Use traditional approach for managing mode 1
- Use DevOps for mode 2

Multi-speed

Why stop at just two modes?

- Gartner ‘pace layered’ model has three modes
  - Systems of record
  - Systems of differentiation
  - Systems of innovation
- You don’t have to stop at three modes either
  - Could have four or even more pre-defined modes
  - Define your own classification based on customers risk appetite, agility needs, stability, security etc.

What is DevOps anyway?

- There are lots of different definitions of DevOps
- There are many different ideas in DevOps
- Let's separate some of these out so we can ask
  - Can we use this idea for legacy or critical systems?
  - If we can't then why not, and what could we do about it?

Some things that make up DevOps

- Multi-disciplined development teams
  - Owning the whole lifecycle of each service
- Collaboration within and between teams and orgs
- Small batch sizes, limiting WIP, managing flow
  - Using Agile and Scrum for software development
- Maximum use of automation
  - Continual integration, testing and deployment
  - Infrastructure as code
- Experimentation, fast failure, amplified feedback and continual improvement

And another thing…

Other ideas that inform DevOps culture

- Lean
- Kanban
- ToC
- Anti-fragile designs
- Managing technical debt

Which DevOps ideas can be used with legacy or critical systems?

Multi-disciplined development teams

- IT managers have been complaining about poor handover from Dev to Ops for many years
- Assigning ownership of the end-to-end service lifecycle to a multi-disciplined team could easily be adopted in legacy or critical systems
  - Critical systems may need to implement appropriate Separation of Duties or Dual Controls to preserve security

Collaboration within and between teams and orgs

- Improved collaboration within teams, between teams, and between organizations would benefit any IT service
- It can be hard to create the culture and environment needed for collaboration, but this is an issue that can be addressed

Small batch sizes, limiting WIP, managing flow

- Small batch sizes may be possible in some legacy and critical environments, but could be very difficult in others

- Limiting WIP is beneficial in any context
- Managing flow is beneficial in any context

Using Agile and Scrum for software development

- Agile and Scrum are probably not appropriate for some legacy systems and some critical systems

- Even when Agile and Scrum aren't suitable for software development they can be used for CSI, changes to operating procedures, ITSM processes etc.

Maximum use of automation

It can be difficult to automate some legacy systems

- But there are some great automation tools around

Automated integration, testing and deployment will benefit any system

- For critical systems this can reduce risks and improve the quality of audit logs

Infrastructure as code will only work with some types of infrastructure

- But it can improve reliability and audit so is ideal for critical systems

Experimentation, fast failure, amplified feedback and continual improvement

Some types of experimentation may not be appropriate for some legacy systems and some critical systems

- But some experimentation is always possible

Fast failure, amplified feedback and continual improvement are good in every environment

- We all need to get better at detecting and learning from incidents and problems

Lean and Kanban

Lean is a culture of continuous improvement

- LEAN is not a tool or a method
- It can be applied to any environment
- Eliminates waste by identifying value creation

Kanban is a way of controlling workflow

- Based on visualizing workflow, limiting work in progress and managing flow
- Kanban is suitable for any environment

ToC

Theory of Constraints is a set of management thinking tools

- ToC ideas have been adopted by DevOps people, but the concepts are not IT related
- ToC ideas can be adopted in any organization
  - Especially the concept that every system has a bottleneck and this is the only place where improvements should be made

Anti-fragile designs

Anti-fragile designs recognize that

- However rugged our systems they will always fail
- ‘Black Swan’ events can result in catastrophic impact
- Designing systems to recover quickly with minimal impact is generally better than trying to design them not to fail

These ideas can be used with any system. They don’t say to design things that break easily, but design things that can be fixed easily!

Managing technical debt

Technical debt is one of the biggest problems with legacy systems

- It may have accumulated over years, and be difficult to address due to the high risk of intervening
- Recognising and managing technical debt will help any system

Technical debt is not acceptable for critical systems

- It should be carefully managed and not allowed to threaten the business

Summary

Multi-disciplined development teams	✓
Collaboration within and between teams and orgs	✓
Small batch sizes, limiting WIP, managing flow	?✓
Using Agile and Scrum	?✓
Maximum use of automation	?
Experimentation, fast failure, amplified feedback and continual improvement	?✓

Lean	✓
Kanban	✓
ToC	✓
Anti-fragile designs	✓
Managing technical debt	✓

What next

Never say DevOps isn't suitable for your environment

Consider all the ideas that DevOps can bring to an organization and adopt the ones that will work for you

- Then adapt them to suit your environment
- Add more DevOps ideas over time (think Agile)

Don't worry if other people tell you you're not doing DevOps properly, you'll still get the benefits

Thank you

@StuartRance

StuartR@optimalservicemanagement.com

optimal
Service Management Ltd