Microservices and DevOps for Telstra TV3

Grant Simonds, Agile Ways of Working Coach
4 months to deploy a new application
84 Outages over 2 years
The TTV3 Dilemma

• More complex than any other TTV with new technical challenges including voice controls
• Critical to get voice response within 5 sec
• 3 months to deliver TTV3, Voice and Box Office on the same day
How are we going to do that?
Microservices!
Monolith

Application

Database

→

Microservices

Diagram showing the transition from a monolithic application to microservices architecture.
"Find Game of Thrones"

"No problem! Here it is..."
Old World Challenges

- No support for HTTP/2
- Implementing MA-SSL wasn't straightforward
- Hard to add additional resources
- Production change time = 5 days
"Find Game of Thrones"
Architecture Today

- ~44 individual services
- Multiple squads work together
- Flexible technology stack
- Production change time 2-4 hours
Microservices are good...

- Resilience
- Scalability
- Time to Market
- Cost Reduction*
- Easy to Debug
- Flexibility & Improved Parallelism
...but there's also a LOT more services to manage
DevOps to the rescue
COMPARING THE ELITE GROUP AGAINST THE LOW PERFORMERS, WE FIND THAT ELITE PERFORMERS HAVE...

- 46 TIMES MORE frequent code deployments
- 2,555 TIMES FASTER lead time from commit to deploy
- 7 TIMES LOWER change failure rate (changes are 1/7 as likely to fail)
- 2,604 TIMES FASTER time to recover from incidents
What is DevOps

• Developers and Operators working together with end to end responsibility
• More higher quality deployments
• Higher frequency of learning
I want change

I want stability!

Dev

Ops

Wall of Confusion
DevOps for Telstra TV

- Continuous Integration Pipeline built by Devs early on
- Continuous Delivery built by Integrators before Prod deployments
- No risky changes: In-service upgrade
- Automate functional and unit tests
- Operational acceptance streamlined
- Over 500 deployments of Telstra TV backend in 6 months
- Zero major outages
CI/CD Pipeline

- Jira Issue
- Write Code
- Commit /PR
- Build
- Test
- Deploy To Dev
- Test
- Deploy To Stage
- Test
- Deploy To Prod
- Test
- Monitor
Stack

Java 8/11
SpringBoot
Docker
Kubernetes
Rancher
AWS

Pipeline

Jira/Confluence Cloud
IntelliJ
Bitbucket
Bamboo
Nexus
Checklist for DevOps

- Developers and Ops in the same team
- Operate what you build
- Continuous Everything
- Minimise Dependencies
- Devs take ops perspective
- Ops take dev perspective
How the world is better

• From 4 months to 2 hours
• Scales within 5 minutes
• Responses within 10’s of milliseconds
• Tolerates failure
Team dynamics
Monitoring
Tools
Flexibility vs. Chaos
Graceful failure
Cost
Automation
Progress over perfection
Service boundaries
The microservice complex
Thank You!