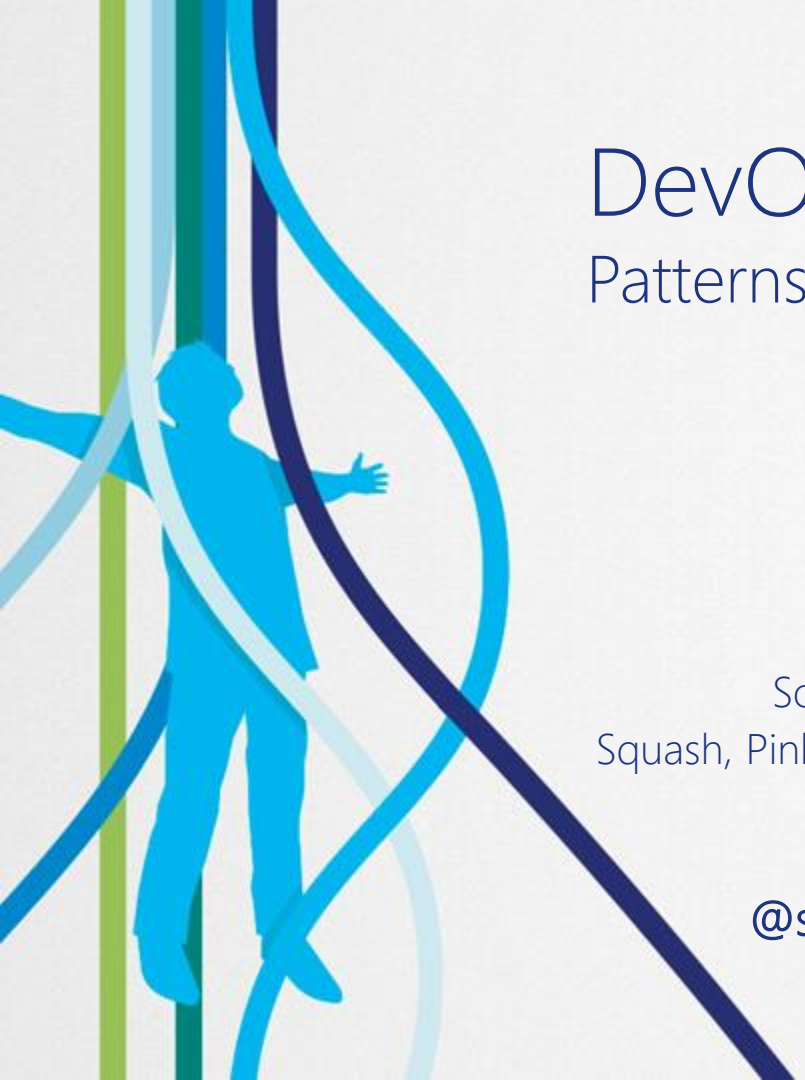

DevOps
Patterns and Anti-Patterns

Stefan Koell
Microsoft MVP

Software Developer (Royal TS)
Squash, Pinball, Arcade, Audio and Home Theater Enthusiast

@stefankoell, code4ward.net

Microsoft

iQSol
Security made in Austria.

?

Experts Live Austria

Upcoming Events

- 23.8. – 25.8. Experts Live Europe - http://expertslive.eu
- 14.9. Experts Live Café Wien
- 5.10. Experts Live Café Linz
- 7.11. Experts Live Austria

http://expertslive.at

What to Expect?

- An overview of DevOps for IT Professionals
- An idea on how DevOps can help
- Some patterns and anti-patterns
- How to get started with DevOps?
- No fancy slides

What is DevOps?

… is a *culture, movement or practice* that emphasizes the **collaboration and communication** of *software developers and other information technology professionals* while automating the process of software delivery and infrastructure changes.

https://en.wikipedia.org/wiki/DevOps

DevOps in other words …

… two or more teams (operations and development), responsible ***together*** for the entire product lifecycle.

DevOps is not about technology, it's about collaboration.

Real World Example

Operations

Software Development

Real World Example

How can DevOps help?

Be Agile!

- Continuous Integration
  automated builds and test
- Continuous Delivery
  always be able to release
- Continuous Deployment
  automatically put into production

How can DevOps help?

- Better product quality
- More reliable releases
- Improved productivity and efficiency
- Improved customer satisfaction

Important: DevOps works everywhere!

Pattern vs. Anti-Pattern

Pattern

Antipattern

Death March

- Everyone knows the project is doomed
- Everyone is forced to keep on going
- A lot of overtime demanded
- Result of unrealistic and overly optimistic expectations in scheduling and/or feature scope

Death March

- Train project management
- Policy for monitoring and controlling projects
- Manage risks and issues
- Provide appropriate resources
- Constrain project scope

As a team member

- **Communicating the issues**
- **Leave project**

Death by Planning

- Project never takes off
- Too many meetings
- Leadership wants to plan every aspect and detail

- Planning is still important and essential!

Death by Planning

- Specific Agendas
- Do standups where possible
- Only invite required people
- Track meeting costs

Feature Creep / Scope Creep

Feature Creep / Scope Creep

- Additional functionality is continually added
- Design by committee
- Product is ready, unnecessary features are added
- Delays shipping / delivery

Over-Engineering

- Endless polishing of product or code
- Too many abstractions
- Too much flexibility
- Balanced decisions:
  - Backend just needs to work
  - UX must be good from the beginning
  - KISS – Keep it simple, stupid

Shipping is a Feature

A 50%-good solution that people actually have solves more problems and survives longer than a 99% solution that nobody has because it's in your lab where you're endlessly polishing the damn thing.

Find the MVP (minimum viable product).

Shipping is a feature.
A really important feature. Your product must have it.

Duct-Tape or Cowboy Coder

- Creates amazing stuff in short time by hacking code fragments together
- Produces code which is hard to maintain
- Affects the whole team
- Not always “bad” – can sometimes be useful

Dogfooding

More Patterns (Practices)

- Whole Team Activity
- Pair programming
- Collective Code Ownership
- Common Architectural Vision
- Coding Standard / Readability
- Rubber Duck Debugging

More Anti-Patterns

- Reinvent the Wheel
- Not Invented Here (Distrust of foreign code)
- Found on the Internet
- Copy Folder Versioning – Use Source Control!
- Copy Paste Programming
- Spaghetti Code
- Broken Window
- Shiny Toy
- Magic Strings and Numbers / Don’t Repeat Yourself (DRY)

How to get started with DevOps?

- DevOps is a management and cultural change
  Gamification, Safe to Fail, Everyone must be on board
- Automation is key
- Incremental adoption
- Start small and keep on improving
- DevOps is not a goal, it's a journey

Sources

- DevIQ
  http://deviq.com/category/antipatterns/
- Broken Windows Theory
  Academic theory proposed by James Q. Wilson and George Kelling in 1982 that used broken windows as a metaphor for disorder within neighborhoods.
  https://www.britannica.com/topic/broken-windows-theory
- The Internets

Thank you!

DevOps
Patterns and Anti-Patterns

Stefan Koell
Microsoft MVP
@stefankoell, code4ward.net