

DevOps Flow 2019: Intro

Sergii Sinienok
Cloud Architect @ Dev-Pro.net

Few words about myself



- 12+ years in IT
- Passioned about enterprise-grade solutions
- Cloud- and service-oriented mindset
- [LinkedIn](#)

Agenda

- The origins of DevOps - the story behind.
- Traditional vs Digital products
- 3-Tier Monolith vs Microservices
- Organization structure and DevOps
- Core DevOps Practices
- DevOps roles - examples
- DevOps Flow 2019 - overview

DevOps - Definition

DevOps is an organizational and cultural movement that aims to increase software **delivery velocity**, improve service **reliability**, and build **shared ownership** among software stakeholders. It is **not Who** (engineer), it **is What** (Values, Processes, Practices).

- Automation
- Human to Machine knowledge sharing
- Teamwork (Dev, Ops, QA, etc.)

DevOps - The Story Behind

- If you need software - guess what?
You need to build it somehow!
- The need to standardise and
optimize the Software
Development Lifecycle (SDLC)
- Huge demand to formalize SDLC



"Session Lab"

Information Technology Infrastructure Library

- First attempts to take the challenge of **predictable** software delivery.
- ITIL - "Great Britain, State As A Service".
- ITIL Service Lifecycle stages, processes and functions: Service Strategy => Design => Transition => Operations => Improvements



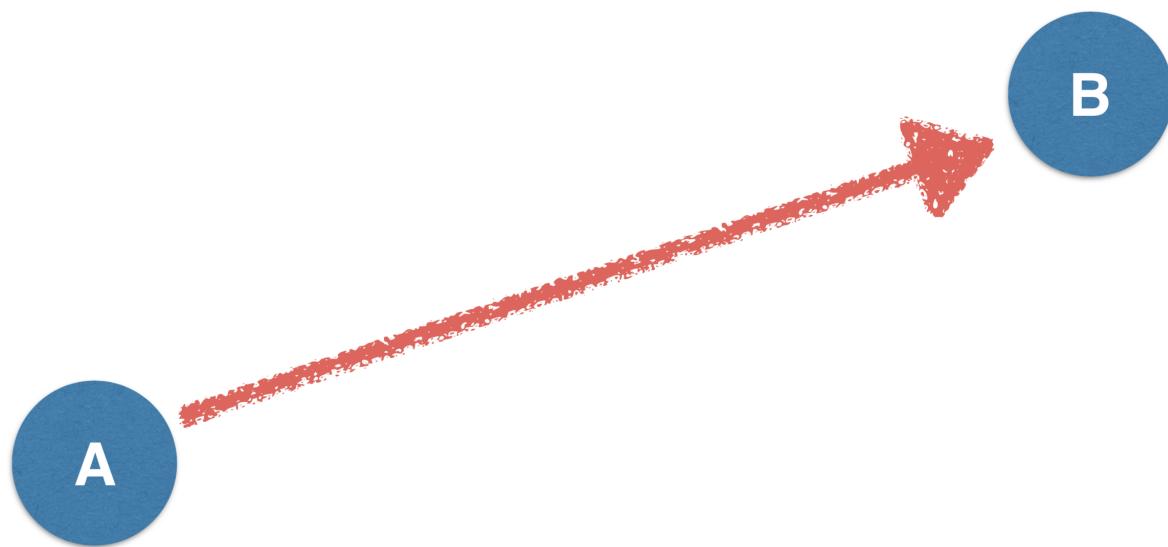
"Tritec Education"

The Origins of DevOps

- IT role transformation: Automate Well Known Processes (Secondary role)
-> Digital product, IT as a business (Primary role).
- 2009 John Allspaw (Ops) & Paul Hammond (Dev) @ Flickr - "[10 Deploys Per Day Dev and Ops Cooperation at Flickr](#)" [slideshare](#)
- "[DevOps origins: Patrick Debois](#)"

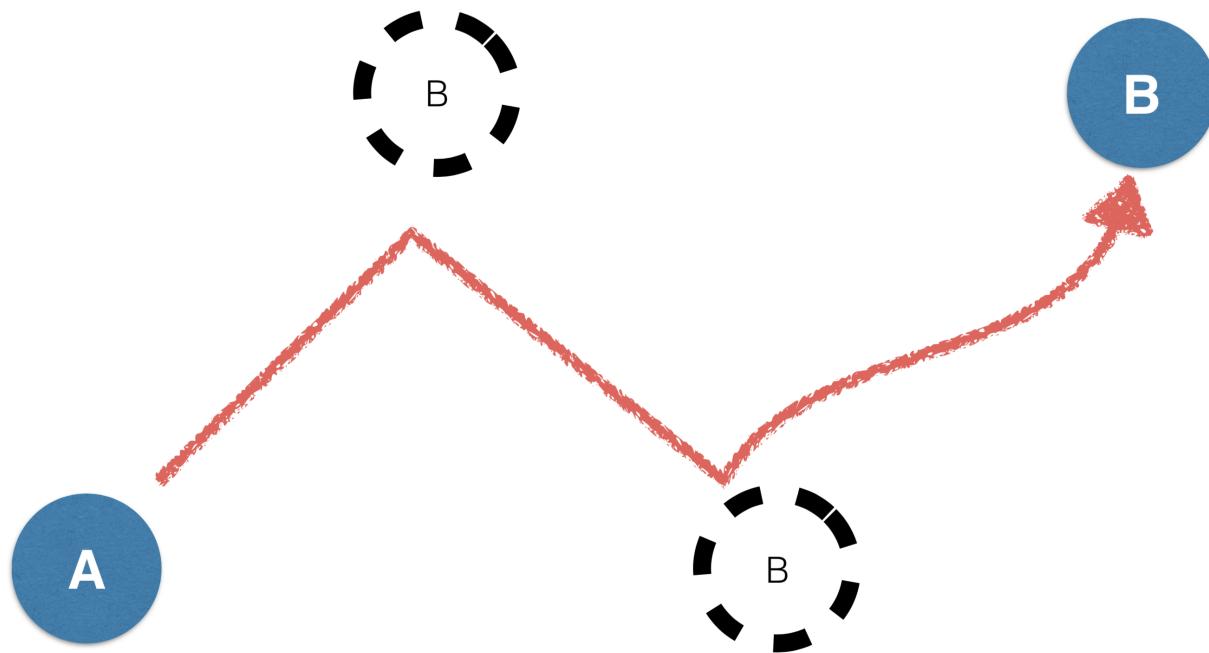
IT Role - Enterprise and Traditional Products

Enterprise processes automation: A => B. **The Idea** remains the same, **details** are subject to change. Straightforward long term strategy.



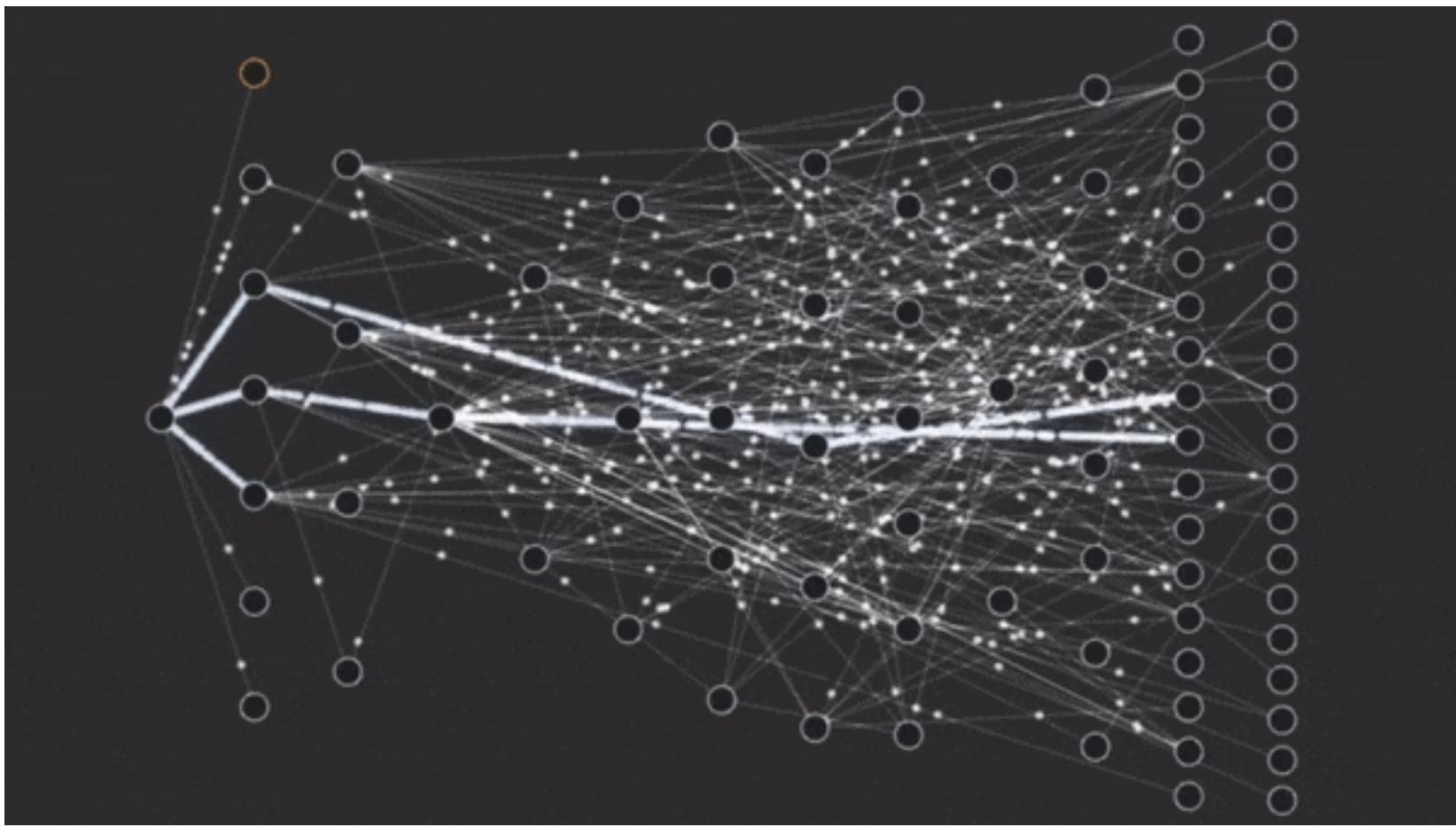
IT Role - Agile and Digital Products

Agile: A => B? => B? => B!!! . The idea itself is a subject of transformation.



Why Not DEV And OPS Separately Moving Forward?

- Complexity of software increases in magnitude
- IT became a business itself instead of serving business.
- Time to Market
- ITIL efficiency is concerned assuming all the above.

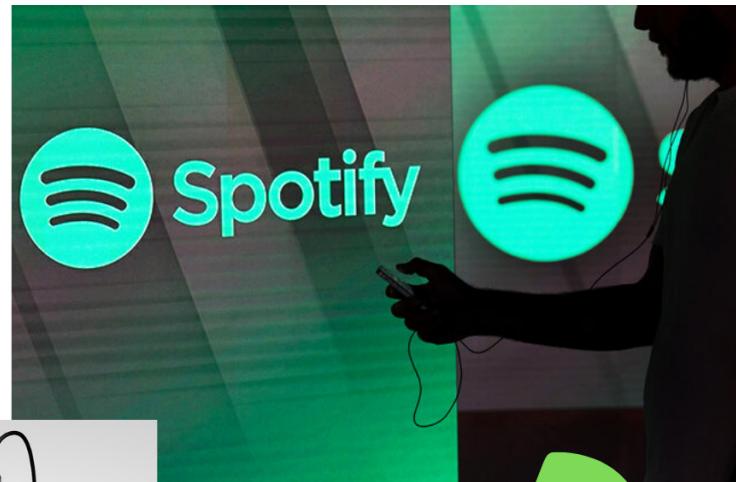
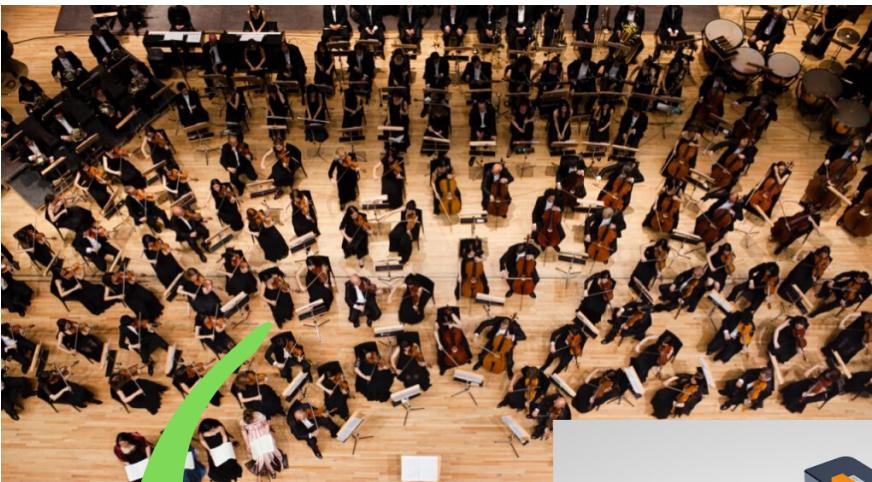


Oldschool - "Traditional" SDLC

- Main value of Computers - Business Processes Automation and efficiency
- Monolith, 3-tier software architecture
- Waterfall, OOP and Domain Driven Design
- Software helps the main product to interact with the market

Agile - "Digital" Products

- Software itself interacts with the market
- Software IS a product



Traditional/Physical VS Digital Products - Examples

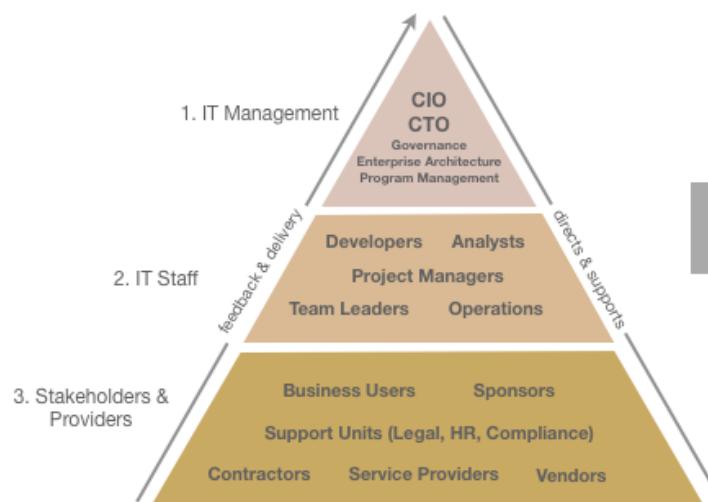
- Traditional: Drinks, Cars, Electricity, Food
- Digital: Uber, Monobank, Slack, Salesforce, Airbnb

Traditional vs Digital Companies - 3P Communication

- Processes
- People
- Products

Old Model of IT

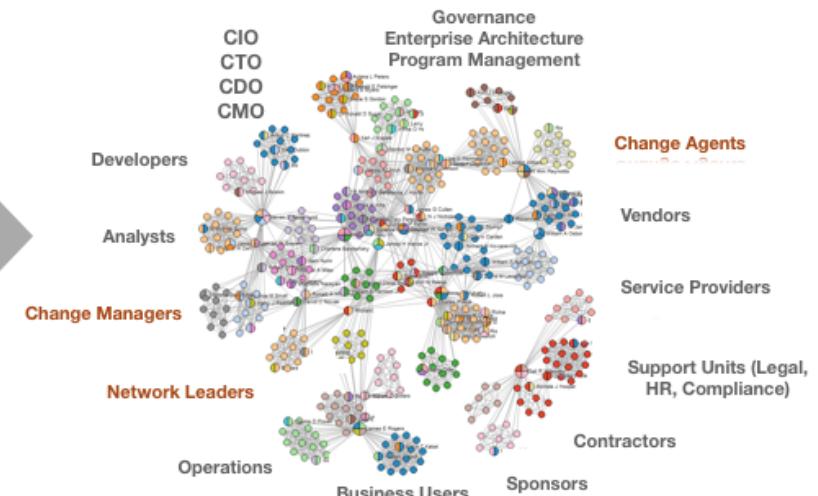
centralized hierarchical automation of business



- Applies technology to what the business does today
- Good at maintaining status quo
- Focus on efficiency, economy of scale, continuity
- Well-defined processes designed for monolithic IT

New Model of IT

decentralized network enablement of digital transformation



- Explores how technology re-imagines the business
- Good at managing constant technology change
- Focus on responding to opportunities at scale
- Dynamic self-organizing processes for small IT in volume



Some Rights Reserved. 2015. **adjuvi** by Dion Hinchcliffe

<https://pbs.twimg.com>

Conway Law

Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure.

["Melwin Conway's Official Website"](#)

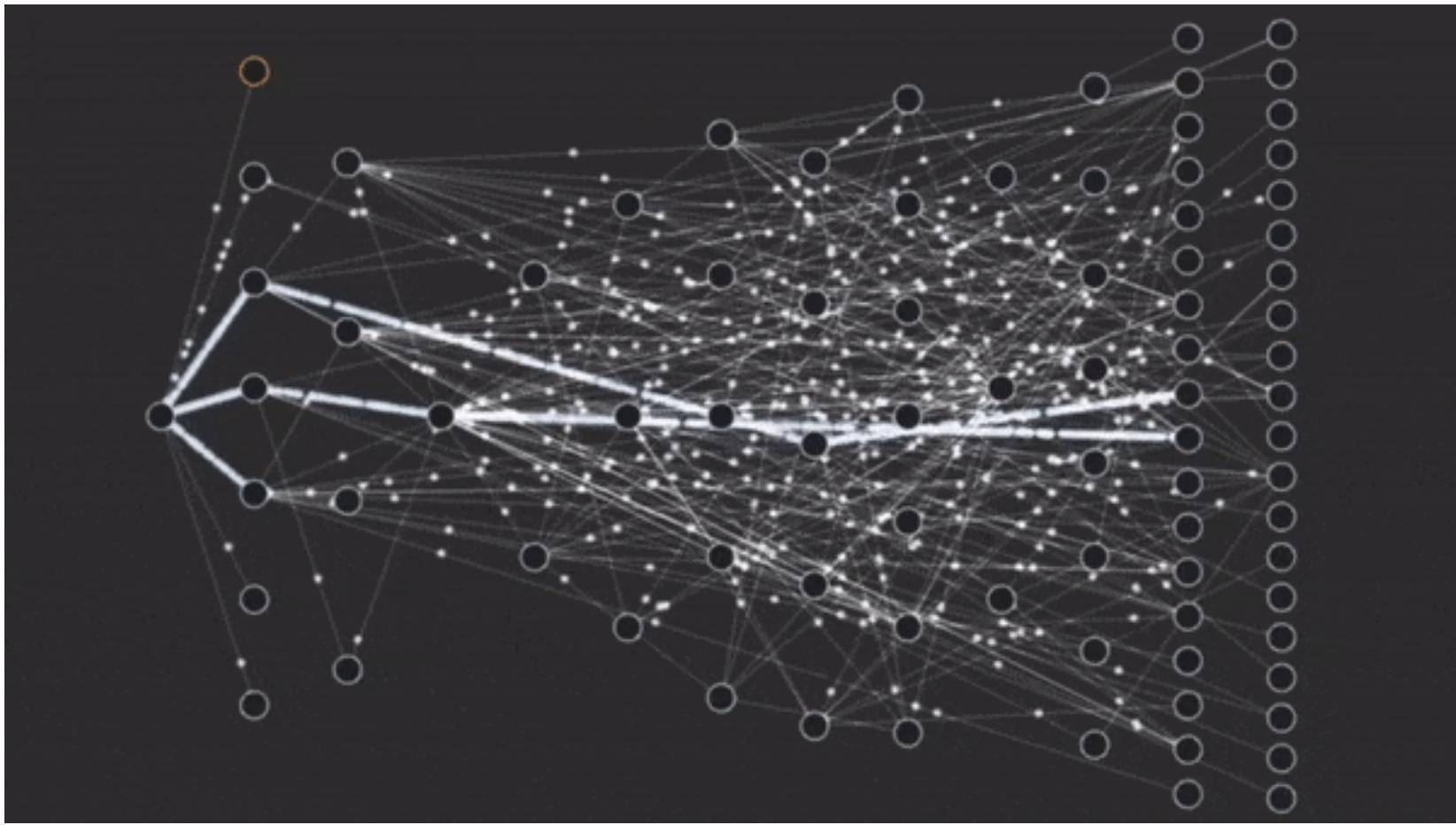


Dr. Conway is manager, peripheral systems research, at Sperry Rand's Univac Div., where he is working on recognition of continuous speech. He has previously been a research associate at Case Western Reserve Univ., and a software consultant. He has an MS in physics from CalTech and a PhD in math from Case.

["Showme Codes blog"](#)

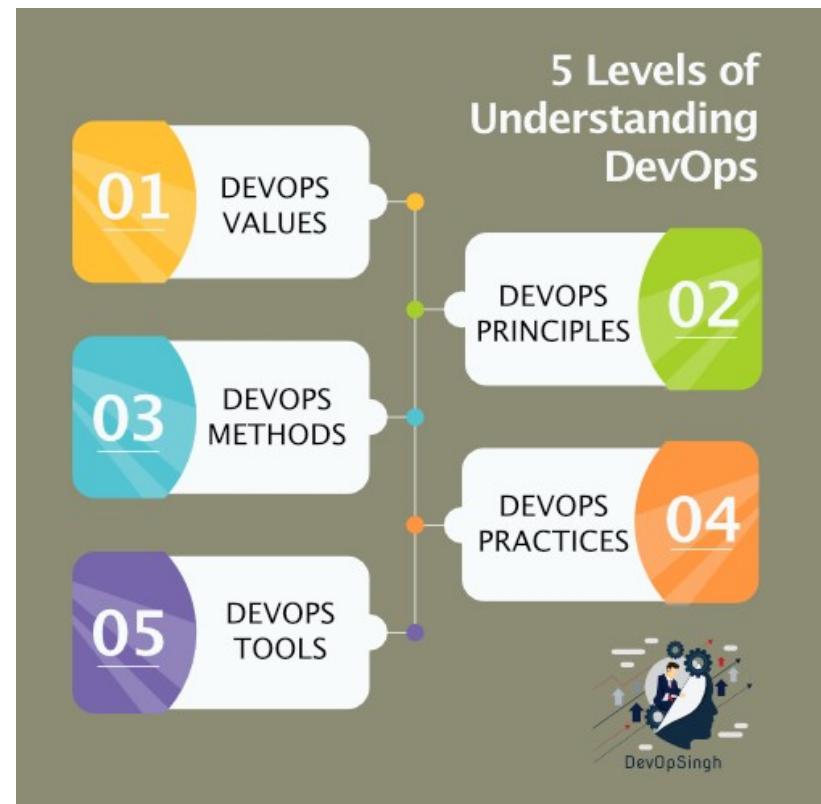
Conway Law - Consequences For Engineers

- Software architecture drift to micro services
- Solutions Complexity Increases
- Software quality degradation - we're learning how to build digital products

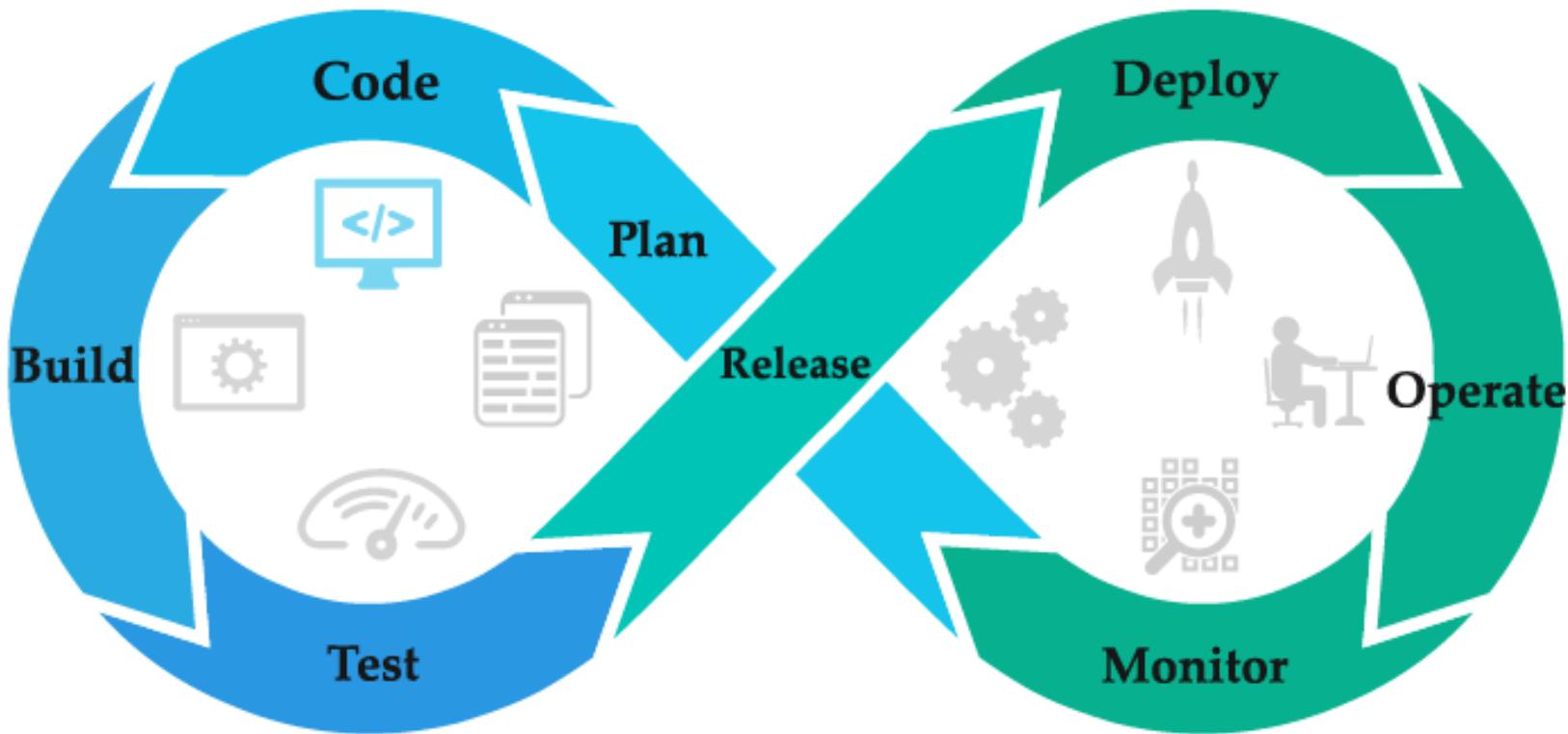


How The Hell It Is All Related To DevOps?!

- Values
- Principles
- Methods
- Practices
- Tools



"DevOpSingh"



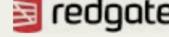
DevOps Research And Assessment (DORA)

- Launched in 2016 by Gene Kim, Jez Humble and Dr. Nicole Forsgren
- The aim is to improve technology and organizational performance
- Prioritizing improvement efforts
- Attracting and retaining talent
- Establishing your brand as a high-performing organization

Presented by  DORA
DEVOPS RESEARCH & ASSESSMENT

Diamond Sponsor 

Gold Sponsors

  XebiaLabs Enterprise DevOps  pagerduty  redgate 

 cloudbees  Datical  sumologic  TRICENTIS 

 Electric Cloud  Microsoft Azure  Pivotal  aws technologies 

DevOps And The Industry: DORA Assessment Report, 2018

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

DevOps, Scrum and Cross-Functional/Full-Stack Teams

- "Full-stack Team" - group of people with the ability, technical knowledge, and the skills to deliver the solution from its idea to working software.
- Scrum Guides: "Development Team => Team"



What Is DevOps - Levels-Based Model

- **CAMS Values** (Culture, Automation, Measurement, Share Knowledge) - "Culture eats strategy for breakfast"
- **Principles:** Full-stack teams (devs, ops, qa, etc.), 2 pizza's teams, Conway law, Agile Ops
- **Practices:** Infrastructure as Code, Source control, Continuous Integration/Continuous Delivery, Continuous Monitoring, Config Management, etc.

DevOps - Values Level. CAMS Values (WHY?)

- Culture - (...) interaction of people and groups, driven by behavior. (...) there is a mutual understanding of others and their goals and responsibilities.
- Automation - get rid of boring routine.
- Measurement - (...) taking a look at the entire operation and assessing it as a whole, not focusing on small parts.
- Sharing the Knowledge (Communication)

"The CAMS Model - Medium"

DevOps - Principles Level (WHAT?)

- Agile processes
- Full-Stack Teams
- Infrastructure Platforms and Cloud Computing
- Automation Everywhere
- Continuous Improvements

DevOps - Fundamental Practices (HOW?)

- Continuous Delivery
- Continuous Monitoring
- Everything as a Code (Infrastructure, Configs, Docs, etc.)

Continuous Delivery

- Predictable delivery on demand
- Delivery Pipeline. Has value for every teammate involved into delivery.
- Delivery Pipeline - image, roles, examples.



["DZone"](#)

Continuous Delivery - Principles

- Repeatable and reliable delivery process
- Automate everything
- Feedback and retrospective actions
- Embedded Quality Control
- Definition of Done > Done === Released to prod
- Continuous Delivery Pipeline is a shared responsibility
- Continuous Improvements

Continuous Monitoring - Principles

- Monitoring as a service
- Centralized and traceable logs
- Visual Representation



Figure 1

"BDO"

Everything as Code - Principles

- Code Development and Collaboration practices everywhere
- Everything as Code, stored in VCS
- Decomposition: Breakdown to Simplify the Complexity

```
252     changePhotoDescription( cell );
253     document.getElementById(bigImageDesc).innerHTML = descriptions[page * 9 + i - 1];
254   }
255 
256   function updatePhotoDescription() {
257     if (descriptions.length > (page * 9) + (currentImage - 1)) {
258       document.getElementById(bigImageDesc).innerHTML = descriptions[page * 9 + currentImage - 1];
259     }
260   }
261 
262   function updateAllImages() {
263     var i = 1;
264     while (i < 10) {
265       var elementId = 'foto' + i;
266       var elementIdBig = 'bigImage' + i;
267       if (page * 9 + i - 1 < photos.length) {
268         document.getElementById(elementId).src = 'images/marco/' + photos[page * 9 + i - 1];
269         document.getElementById(elementIdBig).src = 'images/marco/' + photos[page * 9 + i - 1];
270       } else {
271         document.getElementById(elementId).src = '';
272       }
273     }
274   }
275 }
```

"Archsmaster"

“Base, Service, App” DevOps Appliance Model

- **Base** - OS, Windows, Linux, Runtime, Infrastructure As A Service (Cloud)
- **Service** - Webserver, Database, Cache, Logging Infrastructure, Queues, Auth
- **Application** - our App's code

DevOps - Definition

DevOps is an organizational and cultural movement that aims to increase software **delivery velocity**, improve service **reliability**, and build **shared ownership** among software stakeholders. It is **not Who** (engineer), it **is What** (Values, Processes, Practices).

World of DevOps - Role Examples

- Software Engineer with production infrastructure knowledge
- Infrastructure Engineer
- Infrastructure Services Engineer
- Release/Delivery Manager

Software Engineer With Production Infrastructure Knowledge

- Writes app code
- Automated Tests
- Monitoring
- Infrastructure code.
- Dev: JS, C#, Java. SRE. Systems Integration Engineer

Infrastructure Engineer

- Write code for Infrastructure as a Service.
- Provides and maintains infrastructure for Developers.
- Kubernetes, Ansible Playbooks, Bash :)

Infrastructure Services Engineer

- Writes infrastructure code and maintenance for specific platform services
- Storage (MongoDB Replica Set)
- Monitoring (Prometheus as a service)
- Logging (ELK)
- etc.

Release/Delivery Manager

- Manages the process of the product's delivery
- Identify and track dependencies
- Manages Systems integration process.

DevOps Flow 2019: Practices And Methods

- **VCS: Git** (Everything as a Code).
- **Modern Cloud Providers** (Everything as a Code, Continuous Monitoring, Continuous Delivery).
- **CI/CD** (Continuous Delivery, Continuous Monitoring).
- **Configuration Management** (Everything as a Code).
- **Infrastructure Provisioning Tools** (Everything as a Code, Continuous Monitoring, Continuous Delivery).

DevOps - Definition

DevOps is an organizational and cultural movement that aims to increase software **delivery velocity**, improve service **reliability**, and build **shared ownership** among software stakeholders. It is **not Who** (engineer), it **is What** (Values, Processes, Practices).

Summary

- DevOps is **What**, not **Who**
- DevOps is here as an answer to **Digital Products challenge**
- This course will be focused on DevOps **Practices and Tools**
- This course is mostly aimed to **Infrastructure Engineers**
- Always ask **what** does people **mean by DevOps** :)

Q/A

Thank you!