Carnegie Mellon

institute for SOFTWARE RESEARCH

DevOps: Evolution or Revolution?

Len Bass

Velocity of new releases is important

- Traditionally organizations deployed a new release quarterly or monthly.
- In the modern world, this is too slow.
- Internet companies deploy multiple times a day
- Velocity of releases translates into time to market

Release schedule statistics

- Etsy releases 90 times a day
- Facebook releases 2 times a day
- Amazon has 1000s of releases a day

Definition

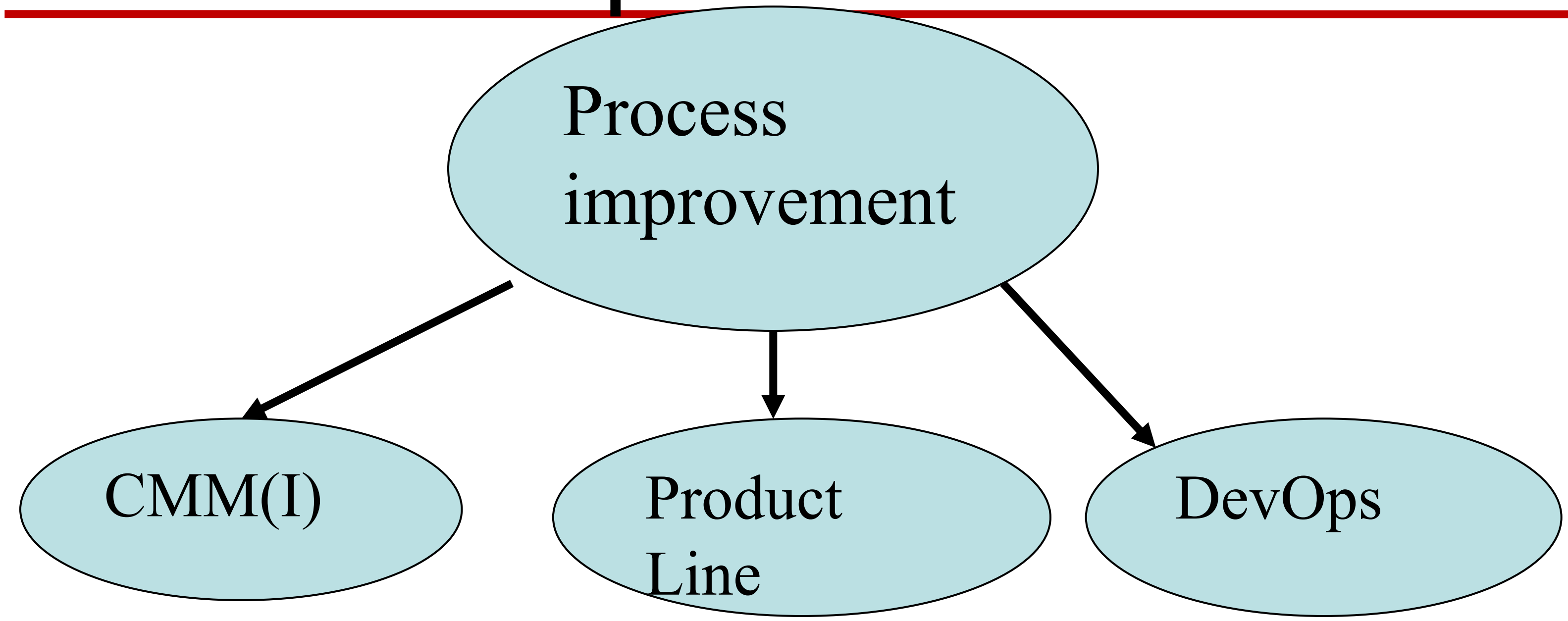
DevOps is a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality.

- DevOps practices involve developers and operators, architectures, and tools.
- They also affect organizations and organizational culture
- DevOps is also a movement – like agile.

DevOps is a Process Improvement Effort

- Time between commit of code and deployment to production is one focus of DevOps.
  - The goal is to make it weekly or shorter
- Time to detect and repair incidents that occur after deployment is a second focus of DevOps
  - The goal is to reduce number and response time for incidents.

Software Process Improvement



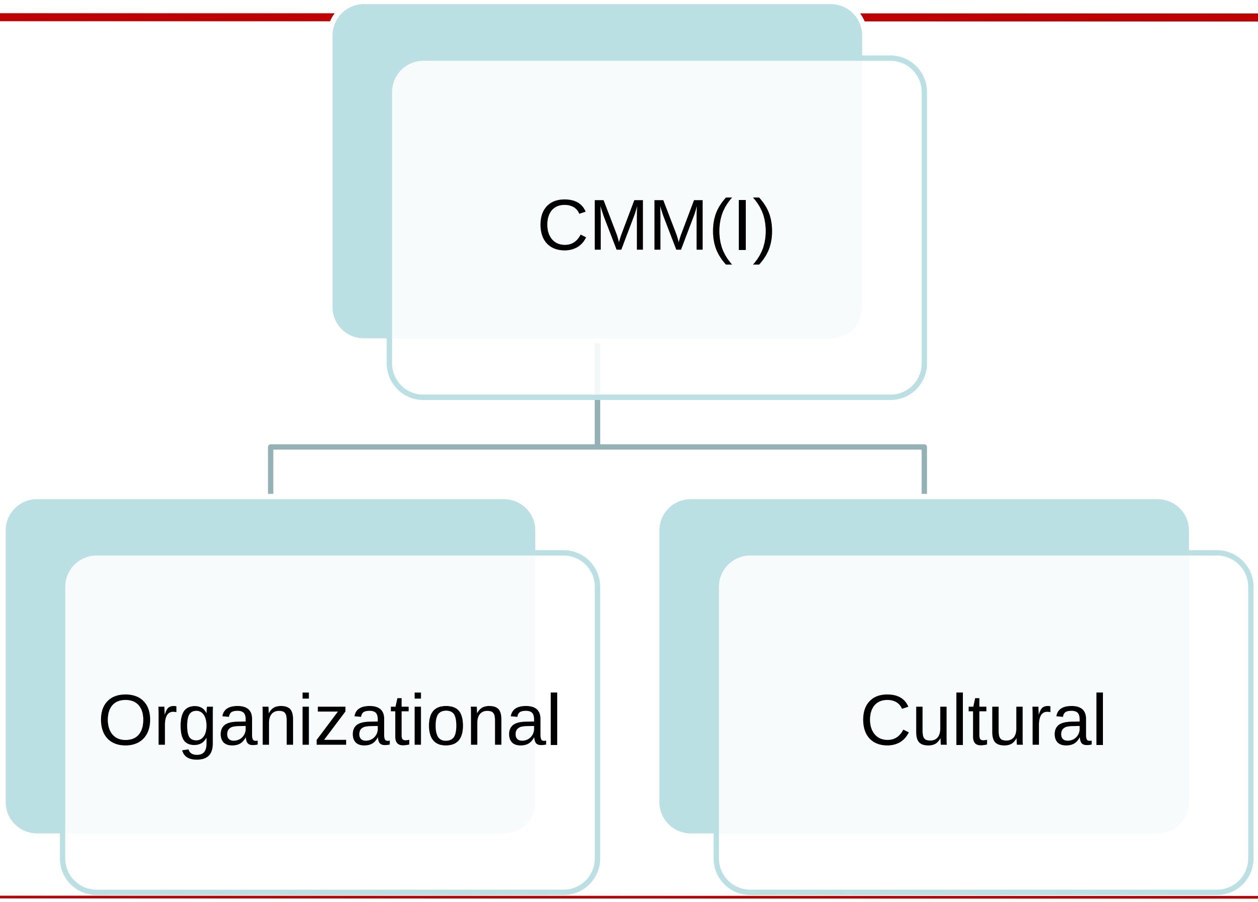
Late 1980s
Code production

Late 1990s
Product production

2010s
Deployment and Operations

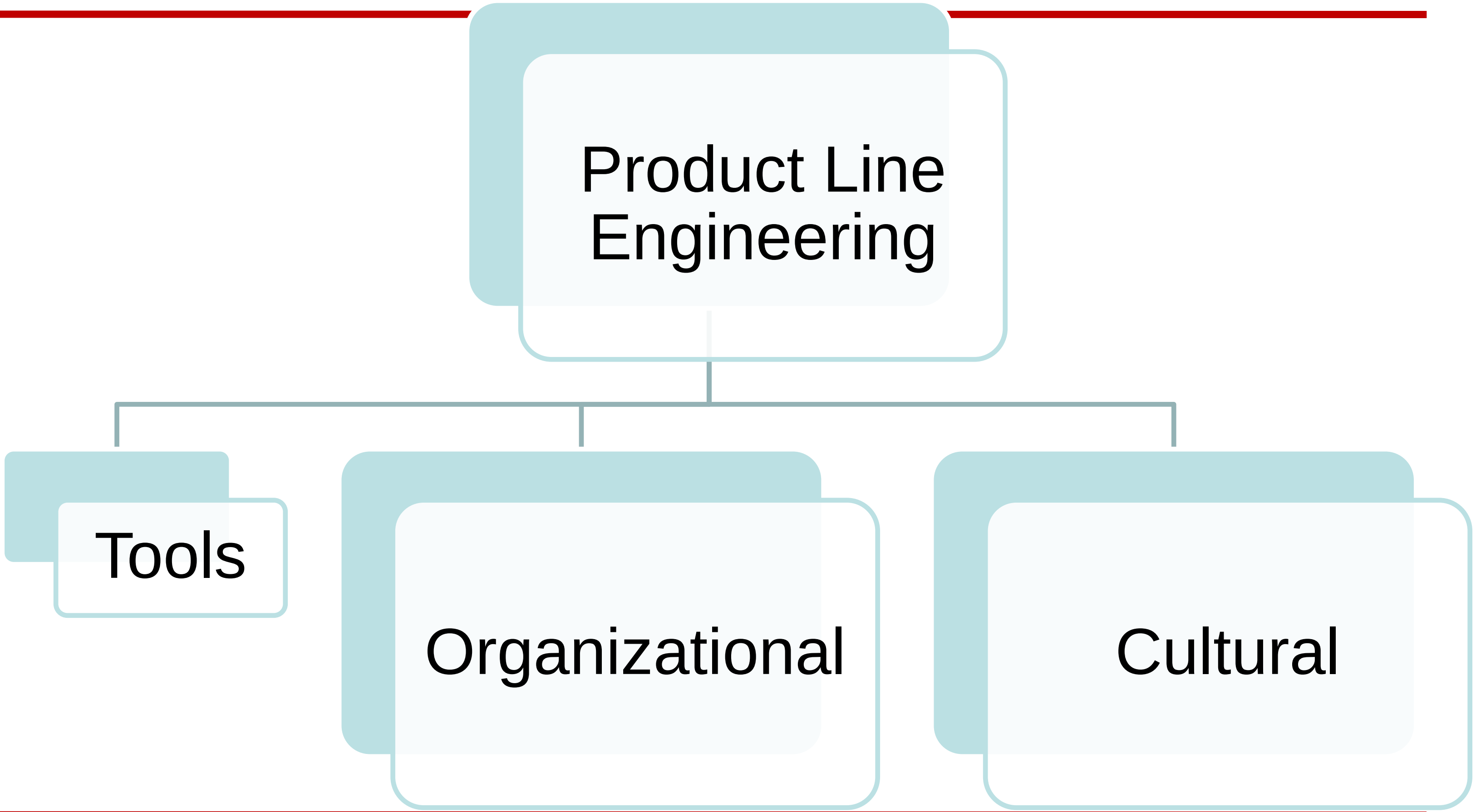
Carnegie Mellon University

CMM(I) Decomposition

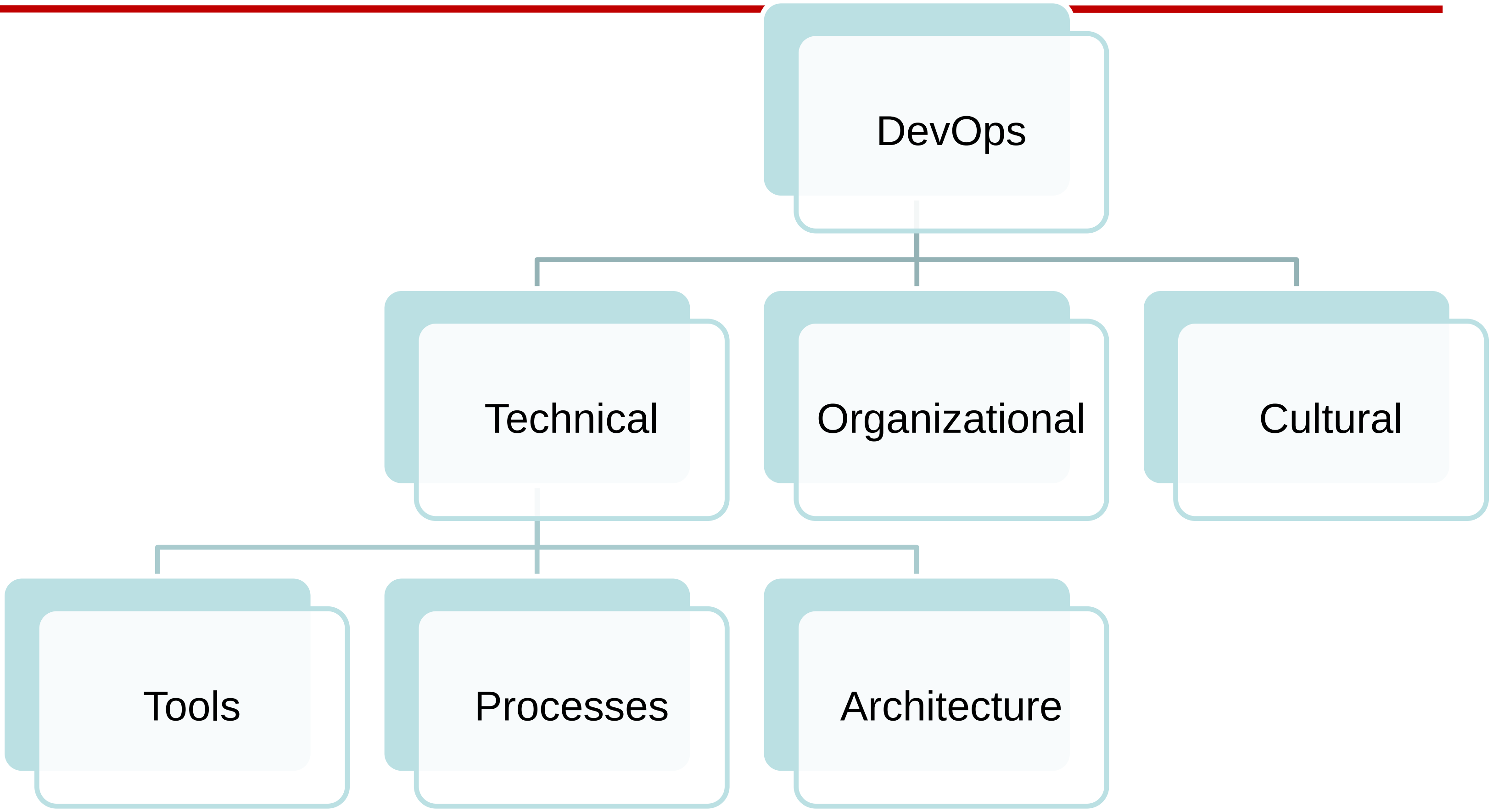


isr institute for SOFTWARE RESEARCH

Product Line Engineering Decomposition



DevOps Decomposition



Key elements of process improvement

- Organization structure
- Metrics

Organization structure

- The CMM(I) advocates a process improvement group to promote and monitor process improvement efforts
- Product line engineering distinguishes between product line group and product development team
- DevOps organizational issues dealt with later in this talk

Metrics

- The CMM(I) has a number of Key Process Indicators. These indicators measure the processes, not the product.
- Product line engineering measures time to develop a new product
- DevOps has metrics for deployment and operations

Carnegie Mellon University

Possible Deployment Metrics

- Deployment frequency.
- Deployment time.
- Customer tickets.
- Automated test pass %
- Defect escape rate.
- Failed deployments.
- Mean time to error detection (MTTD).
- Mean time to error recovery (MTTR).

isr institute for SOFTWARE RESEARCH

Operational metrics

- Performance - latency, page load speed
- Traffic – number of requests per unit time or number of users
- Availability – rate of failing requests or failing services
- Utilization
  - CPU
  - I/O

DevOps differs from prior process improvement efforts

- DevOps focuses on efficiency of deployment and operations
  - Other efforts focused on efficiency of software development
- DevOps is tool heavy.
  - Processes usually have supporting tools
  - This means that discussions of DevOps frequently are about automation and tools.

Automation

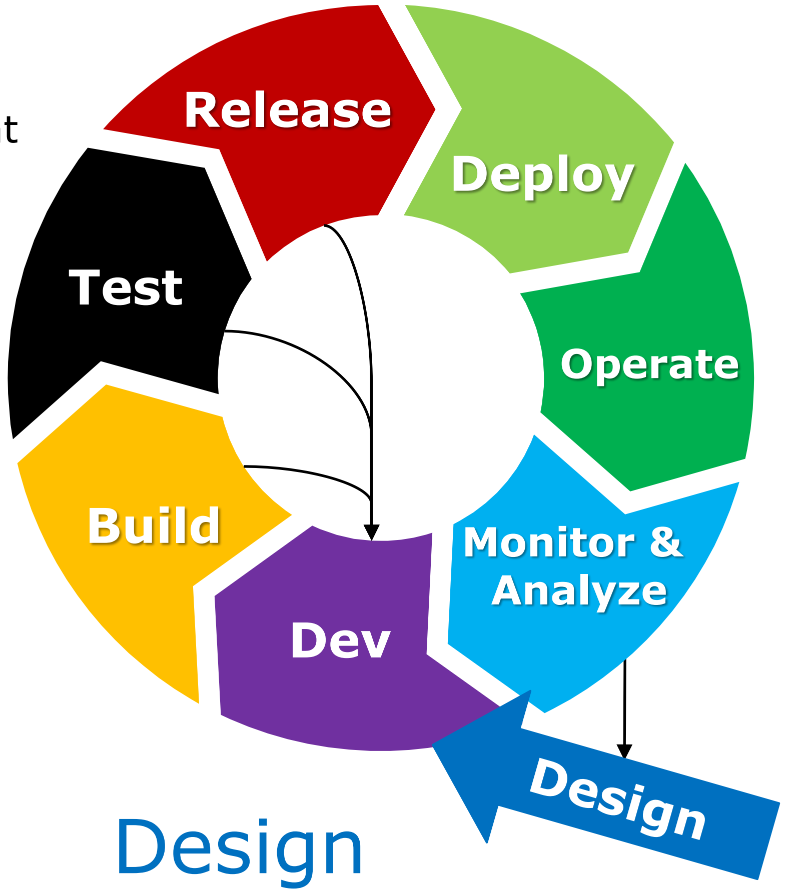
- Infrastructure as code (IAC)
- Scripts that control actions of tools
- Must be written, tested, maintained, and shared
- Allows for repetitive actions to be invoked with a single click
- Scripts should be version controlled

Continuous Deployment

- Code committed to a version control system is automatically placed into production
  - if it generates no build or testing errors.
- This process reduces the time to deployment

Carnegie Mellon University

DevOps Processes



Release

Approve for deployment

Test

Ensure high test coverage & automate tests as much as possible

Build

Create an executable artifact

Dev

Perform normal development activities

Create scripts for other activities

Design

Design architecture to support other activities

Deploy

Move into production environment

Operate

Execute system and gather measurements about its operation

Monitor & Analyze

Display measurements taken during operation & analyze the data

isr institute for SOFTWARE RESEARCH

Architectural support for Continuous Deployment

- Microservice architecture
- Version skew

Microservice Architecture

- A microservice architecture is
  - A collection of independently deployable processes
  - Packaged as services
  - Communicating only via messages

~2002 Amazon instituted the following design rules - 1

- All teams will henceforth expose their data and functionality through service interfaces.
- Teams must communicate with each other through these interfaces.
- There will be no other form of inter-process communication allowed: no direct linking, no direct reads of another team’s data store, no shared-memory model, no back-doors whatsoever. The only communication allowed is via service interface calls over the network.

Carnegie Mellon University

Amazon design rules - 2

- It doesn’t matter what technology they[services] use.
- All service interfaces, without exception, must be designed from the ground up to be externalizable.

- Amazon is providing the specifications for the “Microservice Architecture”.

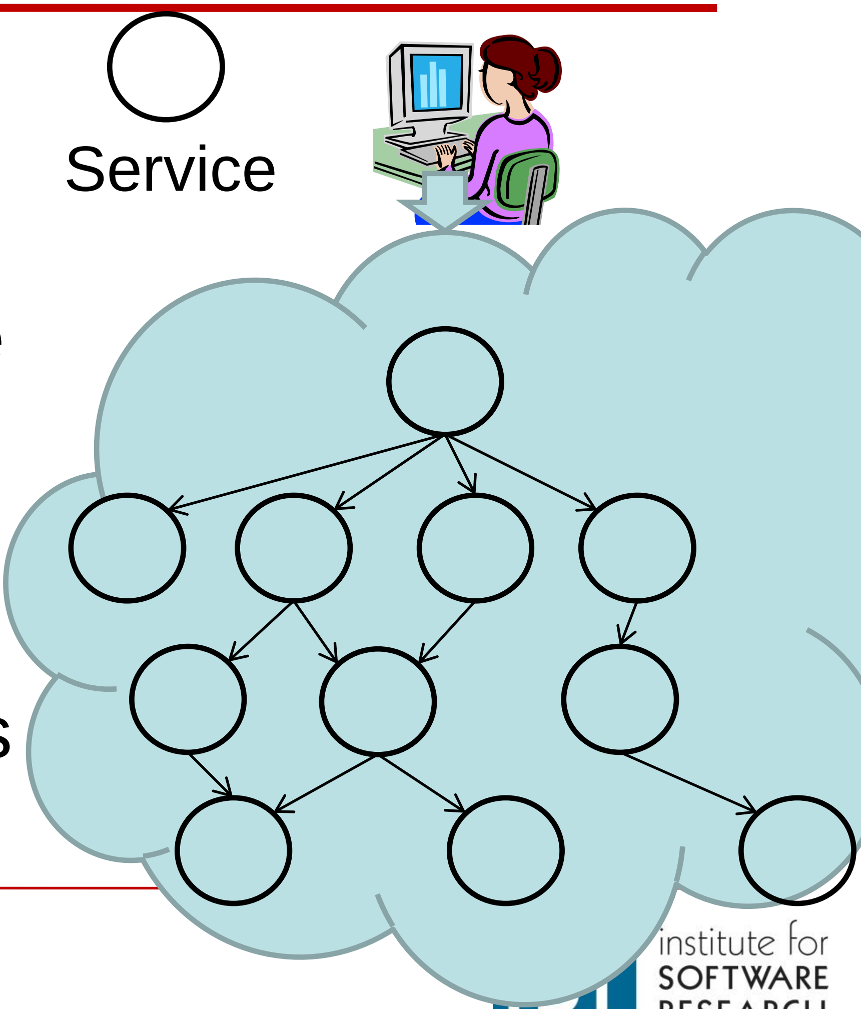
institute for SOFTWARE RESEARCH

In Addition

- Amazon has a “two pizza” rule.
- No team should be larger than can be fed with two pizzas (~7 members).
- Each (micro) service is the responsibility of one team
- This means that microservices are small and intra team bandwidth is high
- Large systems are made up of many microservices.
- There may be as many as 140 in a typical Amazon page

Micro service architecture

- Applications are collections of microservices
- Each user request is satisfied by some sequence of services.
- Most services are not externally available.
- Each service communicates with other services through service interfaces.



Microservice architecture and continuous deployment

- Teams can deploy without coordination with other teams.
- When a team completes revisions on their service
  - They commit it to a version control system
  - This triggers the deployment pipeline
  - If no errors are discovered, it goes directly into production

Version skew

- The elasticity of the cloud will adjust the number of instances of each service to reflect the workload.
- Teams can deploy new version of a service without coordinating with other teams. Other teams are managing clients and dependent services.
- This leads to inconsistencies among versions
  - Client may have been updated to new version whereas server has not
  - Vice versa

Managing version skew

- Messages are tagged with version number of interface
- It becomes the responsibility of the server to manage messages reflecting different versions
  - If message is assuming an older version of a service, service must interpret it correctly
  - If message is assuming a newer version of a service, response must indicate error cause.

Organizational aspects of DevOps

- Incident handling
- Tool management
- Quality Assurance

Incident handling

- Incident - an event that could lead to loss of, or disruption to, an organization's operations, services or functions.
- In software terms – two different types of incidents:
  - Performance or availability problem with running system – topic of this lecture.
  - Security problem with network – handled separately.

Page is sent to first responder

- The first responder can be a developer. The Amazon “You build it, you run it” model
- The first responder can be a separate organizational entity. Site Reliability Engineer (SRE). This is the Google model.
- The SRE model is being adopted by other organizations.

Tool management

- Some organizations have a separate department responsible for tool management.
- Some organizations mandate tools to be used.
- Other organizations allow development teams to choose tools.

Quality Assurance

- If automated testing is used, it changes the responsibilities of the QA group.
  - Development teams supply unit tests
  - QA group supplies system wide tests. Functional, quality.

Summary

- DevOps is a process improvement effort concerned with deployment time and incident handling time.
- Metrics reflect emphasis on deployment and incident handling.
- The deployment pipeline is supported by a large range of tools.
- DevOps processes have an impact on organization structure and responsibilities.

More Information

- "Deployment and Operations for Software Engineers" is available from Amazon.