DevOps for Software Engineering

Khondoker Yasin Ahnaf Prio
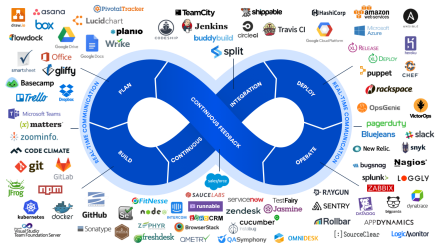
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA

4/20/2019
CSci Senior Seminar

The Big Picture

- Practices and methods to ensure efficient delivery and maintenance of software
- Gap between development and operation teams.
- DevOps closes this gap

DevOps Technologies

Outline

Outline

Software Design and Development Cycle

Multiple developers write code for different features that together make software

- Repository is a central location where source code lies
- Build process is where source code is converted to object code using a set of instructions

Working Example
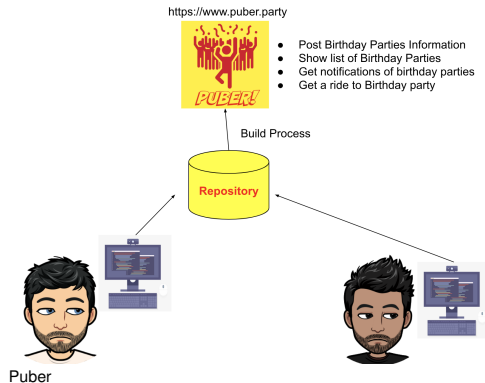
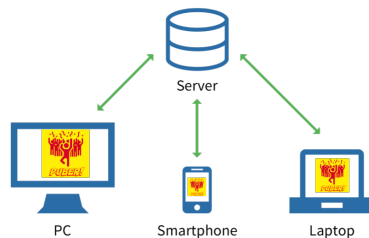# Software Operations and Maintenance

Client-Server Model

- Operations team moderate activities to make software run smoothly
- Test and verification that end-users are interacting as expected is done by monitoring platforms

Outline

DevOps for Software Engineering

1. Collaboration

- Share knowledge
- Cross functional teams and features
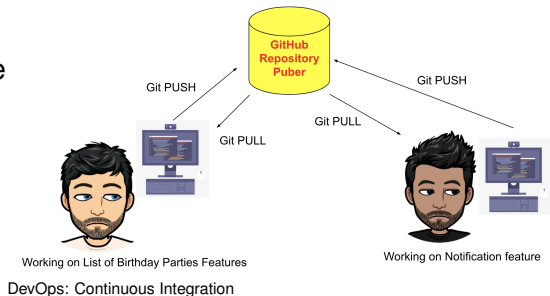- Promote DevOps Culture

DevOps: Collaboration

2. Automation

- Ensure processes are executed same every time
- Trigger automatic feedback at critical points
- Decreases chances of human error

Khondoker Prio   U of Minn, Morris

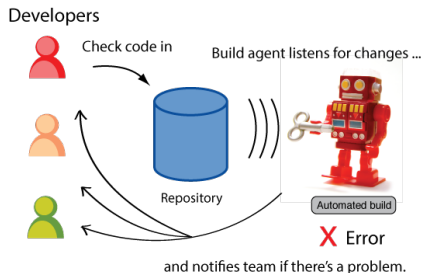DevOps for Software Engineering

3. Continuous Integration

- Development practice where developers integrate code into a shared repository frequently
- Deployment of actively integrated repository is called Continuous Deployment

GitHub Repository Puber

Git PUSH

Git PULL

Git PUSH

Git PULL

Working on List of Birthday Parties Features

Working on Notification feature

DevOps: Continuous Integration

4. Continuous Testing

- Test code in development as features are being built
- Run tests when build starts
- Test in continuous integration pipelines

Developers

Check code in

Build agent listens for changes ...

Repository

Automated build

X Error

and notifies team if there's a problem.

5. Continuous Monitoring

- Reporting of operations to developers and testers
- Response time certain calls
- End user peak timings
- Requests for specific services
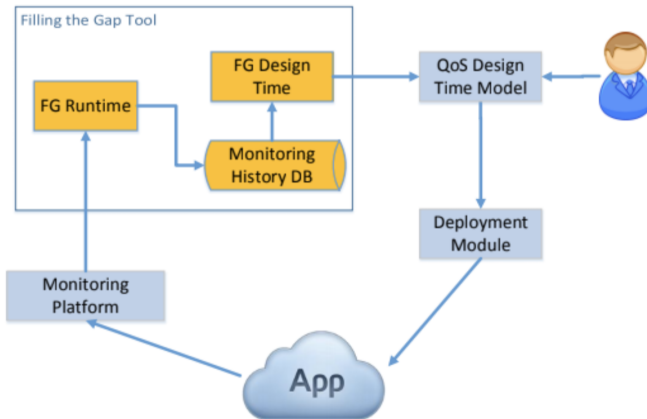
DevOps for Software Engineering

Outline

Objectives

- To parametrize software performance by identifying Service-Layer-Objective and measuring Quality-Of-Service
- To provide the developer with a report of application behavior at run time
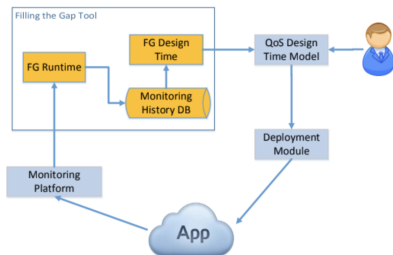
FG tool [1]

FG Tool Architecture

FG tool [1]

Metrics from Monitoring Platform

Parameter	Data Required
Population	Total Number of requests
Resource Consumption	Utilization
	Throughput
	Queue Length
	Response Times
	Queue Length (arrival)
Think time	Throughput
	Total Number of Requests
	Mean Number Requests
Stage duration, transition probs. and efficiency	Start-up duration
	Availability (Up/Down)
	CPU Steal

Metrics Derived [1]

FG Tool's QoS model:

- Resource
- Workload
- Environment

Info set	Item	Struct.	Estim.
Resource	Resources (CPUs, disks)	X	
	Resource multiplicity (number of cores)	X	
	Resource scheduling policy	X	
Workload	Request classes (URIs)	X	
	User population		X
	Sequence of resources used by each request class	X	
	Resource consumption of each request class		X
	Users' think time		X
Environment	Environmental stages	X	
	Average duration of each stage		X
	Transition probabilities between stages		X
	Efficiency factor in each stage		X

QoS Model [1]

Outline

Case Study of companies in Sri Lanka

- Research of Perera, et al [1] which evaluated usage of DevOps
- Identified end goals for success of a software based company

Methods used for Study

- Listed a set of variables and their respective indicators of success in software
- 150 organizations were each sent a questionnaire with those metrics and their usage of DevOps
- Identify relationship between DevOps: (a) Quality, (b) Responsiveness to business needs and (c) Adaptation to newer technologies

Khondoker Prio   U of Minn, Morris

DevOps for Software Engineering

Concept	Variables	Indicators
Quality	Product Quality	1. Functionality 2. Reliability 3. Efficiency 4. Maintainability 5. Usability 6. Portability
	Quality of Development Process	1. On time delivery 2. Budget 3. Rework level
Responsiveness	Responsiveness to changes in features due to business needs	1.Number of defect fix releases 2. Number of releases for new requirements 3. Frequency of Software releases 4. Responsiveness to Business needs
Adaptation to new technologies	Agility	1. Flexibility 2. Speed 3. Leanness 4. Learnings 5.Responsiveness

Models for Goals [2]

DevOps Correlation

The Pearson correlation coefficient measures strength of the linear relationship between two variables.

	Pearson Correlation Coefficient with DevOps
Quality	0.789
Responsiveness	0.641
Agility	0.753

Khondoker Prio   U of Minn, Morris

DevOps for Software Engineering

Coeffecients in Regression Model

Regression coefficients represent the mean change in the response variable for one unit of change in the predictor variable while holding other predictors in the model constant.

Model	Quality	Responsiveness	Agility
Culture	1.174	.780	.763
Automation	.209	.249	.173
Measuring	.166	.160	.142
Sharing	.159	.226	.259
Continuous Deployment	.113	.017	.126

Khondoker Prio   U of Minn, Morris

DevOps for Software Engineering

Outline

- DevOps combines Development teams and operations team to bridge gap and enhances both
- DevOps Key Aspects
- We learn how a DevOp tool would work
- More research to evalaute DevOps and performance modeling

Acknowledgement

- Nic McPhee and Elena Machkasova
- Friends and Family

Thanks!

Thank you for your time and attention!

Questions?

Refereces:

- J. F. Perez, W. Wang, and G. Casale. Towards adevops approach for software quality engineering. In Proceedings of the 2015 Workshop on Challenges in Performance Methods for Software Development.

- P. Perera, M. Bandara, and I. Perera. Evaluating the impact of devops practice in sri lankan software development organizations. In 2016 Sixteenth International Conference on Advances in ICT for Emerging Regions