Desmistificando Microsserviços e DevOps: Projetando Arquiteturas Efetivamente Escaláveis

Prof. Vinicius Cardoso Garcia
vcg@cin.ufpe.br :: @vinicius3w :: assertlab.com

[IF1004] - Seminários em SI 3
https://github.com/vinicius3w/if1004-DevOps

Licença do material

Este Trabalho foi licenciado com uma Licença

Creative Commons - Atribuição-NãoComercial-CompartilhaIgual 3.0 Não Adaptada

Mais informações visite

http://creativecommons.org/licenses/by-nc-sa/3.0/deed.pt

AssertLab
Advanced Software and Systems Engineering Research Technologies

Background

AssertLab
Advanced Software and Systems Engineering Research Technologies

Centro de Informática UFPE 40 anos

What is DevOps?

“Someone told me that each equation I included in the book would halve the sales. I therefore resolved not to have any equations at all.”

— Stephen Hawking (Excerpt From: Bass, Len. “DevOps: A Software Architect’s Perspective”)

Introduction

- “Why should I care about DevOps, and what impact does it have on me?”

- the short answer is that if you are involved in building software systems and your organization is interested in reducing the time to market for new features, then you should care.”
  - DevOps practices will influence the way that you organize teams, build systems, and even the structure of the systems that you build

Defining DevOps

- DevOps has been classified as “on the rise” with respect to the Gartner Hype Cycle for Application Development in 2013

  https://www.gartner.com/doc/2560015/hype-cycle-application-development-

- “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.”
  - Excerpt From: Bass, Len. “DevOps: A Software Architect’s Perspective.”

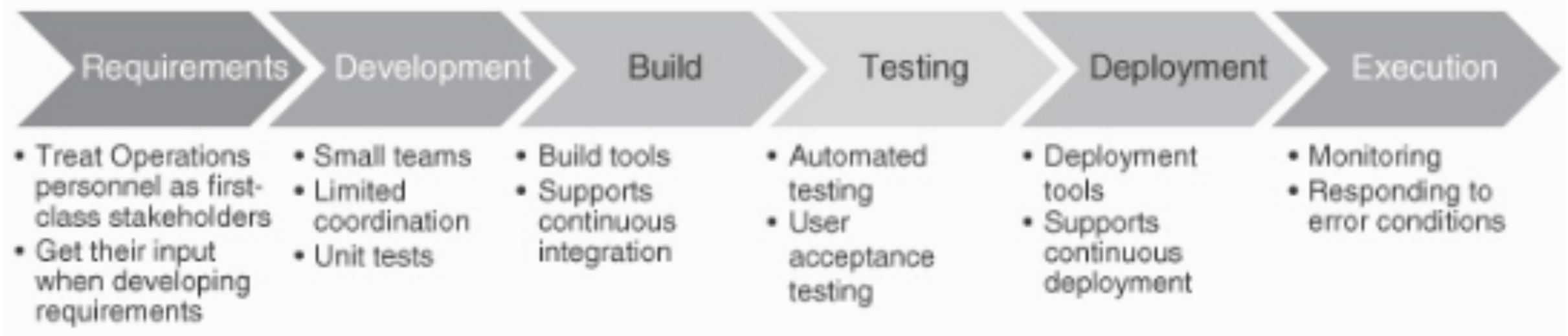
Implications…

- The quality of the deployed change to a system (usually in the form of code) is important ~> *ilities
- Reliability and the repeatability of the delivery mechanism should be high
- Two time periods are important
  - When a developer commits newly developed code
  - Deploying of that code into production
- Our definition is goal oriented
- DevOps practices is not only testing and deployment
  - it is important to include an Ops perspective in the collection of requirements
  - the goal is to ensure high quality of the deployed system throughout its life cycle

DevOps Practices

1. Treat Ops as first-class citizens from the point of view of requirements
   - Operations have a set of requirements that pertain to logging and monitoring
2. Make Dev more responsible for relevant incident handling
3. Enforce the deployment process used by all, including Dev and Ops personnel
   - Ensure a higher quality, avoids errors and the resulting misconfiguration
4. Use continuous deployment
   - Shorten the time between a developer committing code to a repository and the code being deployed
5. Develop infrastructure code, such as deployment scripts, with the same set of practices as application code

DevOps life cycle processes



Why DevOps?

- Response to the problem of slow releases

- Release in a continuous manner
  - continuous delivery or continuous deployment

Release Process

- One of the most sensitive steps in the software development cycle
  - releasing a new version opens the possibility of incompatibilities or failures
  - Organizations pay a great deal of attention to the process of defining a release plan…

A release plan

1. Define and agree on release and deployments plans with customers/stakeholders
2. Ensure that each release package consists of a set of related assets and service components that are compatible with each other
3. Ensure that the integrity of a release package and its constituent components is maintained throughout the transition activities and recorded accurately in the configuration management system
4. Ensure that all release and deployment packages can be tracked, installed, tested, verified, and/or uninstalled or rolled back, if appropriate

Seriousness of getting the deployment correct

- On August 1, 2012, Knight Capital had an upgrade failure that ended up costing (US) $440 million.
- On August 20, 2013, Goldman Sachs had an upgrade failure that, potentially, could cost millions of dollars.

- Excerpt From: Bass, Len. “DevOps: A Software Architect’s Perspective.”

Extent of deployment problems

- XebiaLabs did a survey in 2013 with over 130 responses.
  - 34% of the respondents were from IT services companies with approximately 10% each from health care, financial services, and telecommunications companies.
  - 7.5% of the respondents reported their deployment process was “not reliable”
  - 57.5% reported their deployment process “needs improvement”
  - 49% “reported their biggest challenge in the deployment process was “too much inconsistency across environments and applications”
  - 32.5% reported “too many errors”
  - 29.2% reported their deployments relied on custom scripting, and
  - 35.8% reported their deployments were partially scripted and partially manual.”

Extent of deployment problems

- CA Technologies commissioned a survey in 2013 that had 1,300 respondents from companies with more than (US) $100 million revenue
  - 53% said they were already seeing an increased frequency of deployment of their software or services and
  - 41% said they were anticipating seeing an increased frequency of deployment.
  - 42% responded that they had seen improved quality of deployed applications, and
  - 49% responded they anticipated seeing improved quality

Reasons for Poor Coordination

- Consider what happens after a developer team has finished all the tasks… the system needs to be placed into an environment where:
  - Only the appropriate people have access to it.
  - It is compatible with all of the other systems with which it interacts in the environment.
  - It has sufficient resources on which to operate.
  - The data that it uses to operate is up to date.
  - The data that it generates is usable by other systems in the environment.

Limited Capacity of Operations Staff

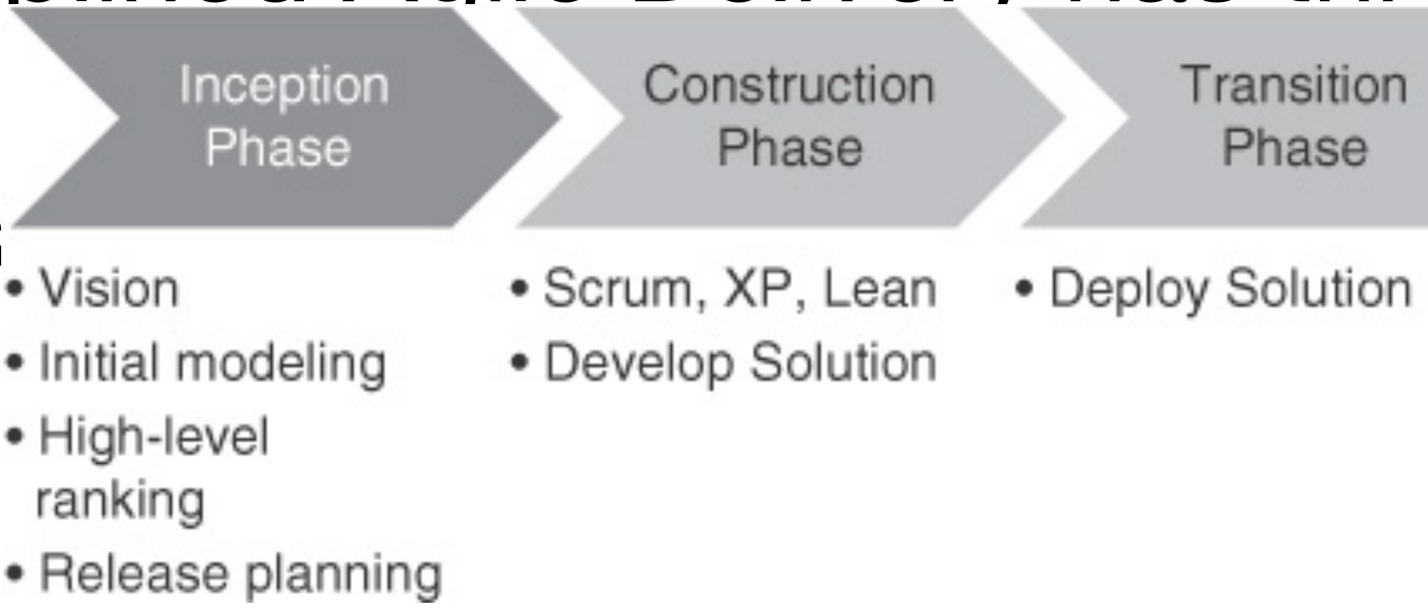
- Analyzing system logs and identifying potential issues with computer systems
- Introducing and integrating new technologies into existing datacenter environments
- Performing routine audits of systems and software
- Performing backups
- Applying operating system updates, patches, and configuration changes
- Installing and configuring new hardware and software
- Adding, removing, or updating user account information; resetting passwords, etc.
- Answering technical queries and assisting users
- Ensuring security
- Documenting the configuration of the system
- Troubleshooting any reported problems
- Optimizing system performance
- Ensuring that the network infrastructure is up and running
- Configuring, adding, and deleting file systems
- Maintaining knowledge of volume management tools like Veritas (now Symantec), Solaris ZFS, LVM

DevOps Perspective

- Automation
  - The steps from build and testing through execution can all be automated to some degree
    - Chef cookbooks or Amazon CloudFormation
    - “infrastructure-as-code"
  - Reduce the incidence of errors and will shorten the time to deployment
- Development team responsibilities
  - Accepts DevOps responsibilities: it delivers, supports, and maintains the service

DevOps and Agile

- Disciplined Agile Delivery has three phas trans



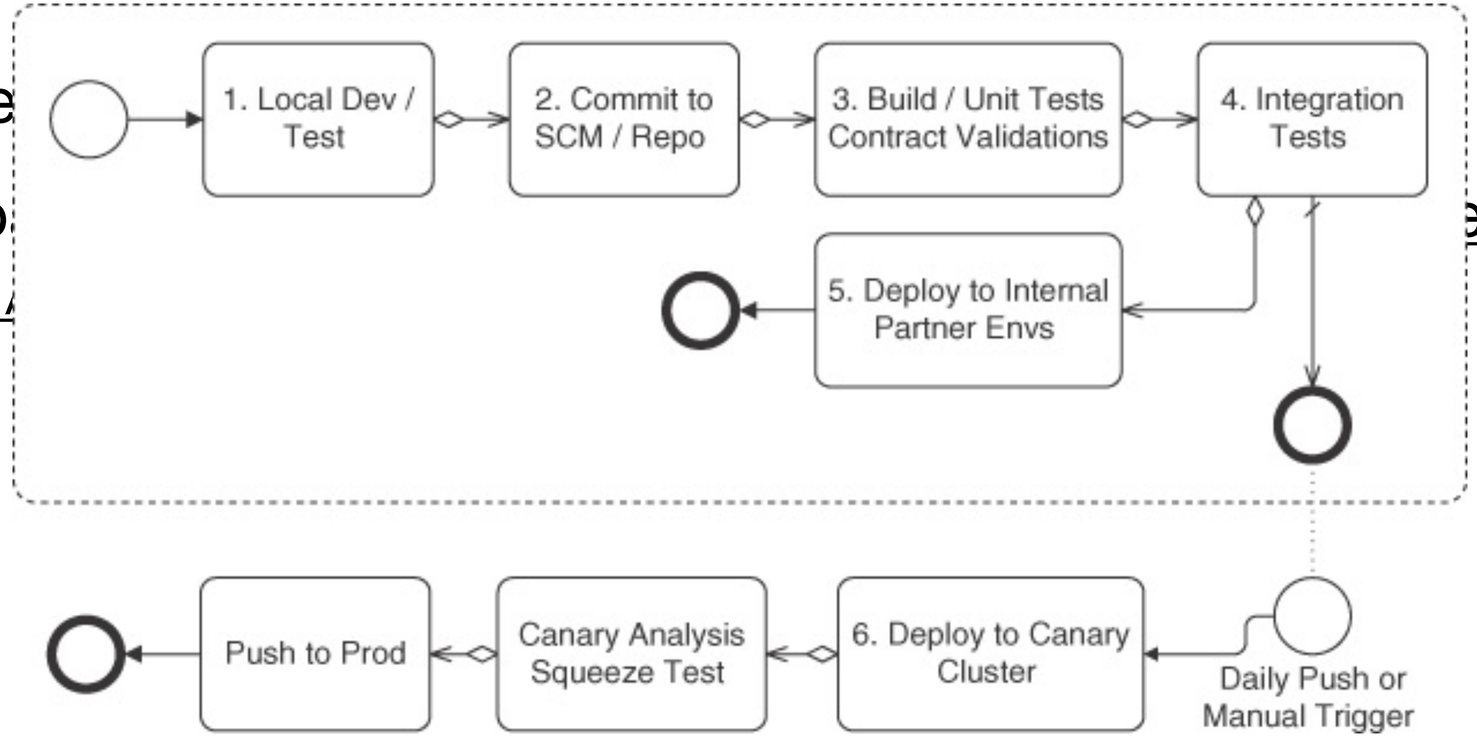
d

Team Structure

- Team size
  - Amazon has a “two pizza rule”
  - Advantages of small teams?
- Team role
  - Team lead ~> "Scrum Master”, team coach, project lead…
  - Team member ~> developer, programmer
  - … service owner, reliability engineer, gatekeeper, and DevOps engineer

Team Role

- Gatekeepe
  - Netflix p etflix.com/ 2013/11/



Coordination

- One goal of DevOps is to minimize coordination in order to reduce the time to market
- Forms of Coordination (pros and cons?)
  - Direct: the individuals coordinating know each other (e.g., team members)
  - Indirect: the coordination mechanism is aimed at an audience known only by its characterization (e.g., system administrators)
  - Persistent: the coordination artifacts are available after the moment of the coordination (e.g., documents, e-mail, bulletin boards)
  - Ephemeral: the coordination, per se, produces no artifacts “(e.g., face to face meetings, conversations, telephone/video conferencing)
  - Synchronous: individuals are coordinating in real time, (e.g., face to face).
  - Asynchronous: individuals are not coordinating in real time (e.g., documents, e-mail)

Cross-team Coordination

- Team coordination mechanisms are of two types—human processes and automated processes
- Cross-team coordination is the most time-consuming factor
- There are two reasons for a development team to coordinate with other development teams
  - Making the code pieces work together (architecture and design decisions)
    - Allocation of responsibilities; Coordination model; Data model; Management of resources; Mapping among architectural elements; Binding time decisions
  - Avoiding duplication of effort

Barriers

If DevOps solves long-standing problems with development and has such clear benefits, why haven’t all organizations adopted DevOps practices?

Culture and Type of Organization

- Benefits of reduced time to market versus the risks of something going awry
  - Almost all organizations worry about risk
- For some organizations the risks of problems occurring outweigh a time-to-market advantage
  - Organizations that operate in regulated domains
  - Organizations that operate in mature and slow-moving domains
  - Organizations whose customers have a high cost of switching to another supplier

Culture and Type of Organization

- For other organizations, nimbleness and fast response are more important than the occasional error caused by moving too fast.
  - Organizations that rely on business analytics to shape their products
  - Organizations that face severe competitive pressure

Adopting a DevOps

- If you are considering adopting a DevOps practice then you need to be aware of three things:
  - What other practices are implicit in the practice you are considering?
    - You cannot do continuous deployment without first doing continuous integration
  - What is the particular practice you are considering?
    - What are its assumption, its costs, and its benefits?
  - What is the culture of your business, and what are the ramifications of your adopting this particular DevOps practice?
    - The difficulty of adopting a practice is related to its impact on other portions of the organization

Organizational & Personnel Issues

- Type of Department ~> what kinds of results are incentivized?
- Silo Mentality
- Tool Support: expertise and common processes
- According to the Datamation 2012 IT salary guide, a software engineer earns about 50% more than a systems administrator.

Summary

- People have defined DevOps from different perspectives
  - operators adopting agile practices or developers taking operations responsibilities
- DevOps faces barriers due to both cultural and technical challenges

Summary

- Some of the tradeoffs involved in DevOps are as follows:
  - Creation of a need to support DevOps tools
  - Moving responsibilities from IT professionals to developers
  - Removing oversight of new features and deployment

For Further Reading

- Gartner's Hype Cycle [Gartner] categorizes DevOps as on the rise: http://www.gartner.com/DisplayDocument?doc_cd=249070 (needs login)
- AgileAdmins explains DevOps from an agile perspective: http://theagileadmin.com/what-is-devops/
- XebiaLabs has a wide range of surveys and state of industry reports on DevOps-related topics that can be found at http://xebialabs.com/xl-resources/whitepapers/
- CA Technologies' report gives some insights into business' different understanding of DevOps and can be found at http://www.ca.com/us/collateral/white-papers/na/techinsights-report-what-smart-businesses-know-about-devops.aspx

For Further Reading

- Jenkins has many third-party plug-ins including some workflows extending into continuous deployment: http://www.slideshare.net/cloudbees
- UrbanCode, a continuous delivery tool suite: http://www.infoq.com/articles/Continuous-Delivery-Maturity-Model
- ThoughtWorks also released its own continuous deployment pipeline suite called Go: http://www.go.cd/

For Further Reading

- Wikipedia links
  - One definition of DevOps we refer to is found at http://en.wikipedia.org/wiki/System_administrator
  - The steps in a release or deployment plan are adapted from http://en.wikipedia.org/wiki/Deployment_Plan
  - The duties of an operator are listed in http://en.wikipedia.org/wiki/DevOps
  - The 5 Whys originated at Toyota Motors and are discussed in http://en.wikipedia.org/wiki/5_Whys

For Further Reading

- Scott Ambler blog: https://www.ibm.com/developerworks/community/blogs/ambler/?lang=en
- Preparing the Netflix API for Deployment: http://techblog.netflix.com/2013/11/preparing-netflix-api-for-deployment.html
- Continuous Deployment at IMVU: Doing the Impossible Fifty Times a Day: http://timothyfitz.com/2009/02/10/continuous-deployment-at-imvu-doing-the-impossible-fifty-times-a-day/

HW 2.1

- Reading the article "Facts, trends and challenges in modern software development” (https://cl.ly/lvtv) and correlated the 10 adages in CD (from HW 1.4) with these facts, trends and challenges.