

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-
Compartilhual 3.0 Não Adaptada



Mais informações visite

[http://creativecommons.org/licenses/by-nc-sa/
3.0/deed.pt](http://creativecommons.org/licenses/by-nc-sa/3.0/deed.pt)





Crosscutting Concerns



3



Business Considerations

“If you want to make God really laugh, show him your Business Plan.
—Barry Gibbons”

Introduction

- If you are in a **startup**, your **management structure** is going to be **sparse**, with **low levels of bureaucracy**.
- As your organization **grows**, the structure changes in all but the rarest of cases.
- Introducing a substantial **new technology** in an **established enterprise** is typically both a **bottom-up** and a **top-down** process
 - The proponents for DevOps are asking for **major changes** in **organizational structure** and in **how** the organization **interacts** with **external stakeholders**



Business Case



6



Sections of the Business Case for the Introduction of DevOps

Section Title	Content
Problem	Why is introducing DevOps practices going to be good for the organization?
Costs	What are the expected costs of the introduction?
Stakeholder impact	What is the impact on stakeholders, both internal and external?
Risks and mitigation	What are the organizational and technical risks associated with introducing DevOps practices? How are these risks to be mitigated?
Rollout plan	What is the plan for rolling out the DevOps practices?
Success criteria	How will we know if the introduction of DevOps practices is successful?

The Problem and Benefits from Solving the Problem

- The overall case for using agile is to **reduce** the **time** between a **business concept** and its **deployment to users**
- DevOps is about **reducing** the **time** between **committing a change** to a system and the change being placed into normal **production**, while **ensuring high quality**



Setting targets

- First, there are a limited number of quantitative reports on the effectiveness of DevOps practices
- Second, there are five different categories of DevOps practices and each practice has some impact on achieving the target values (lecture #2)
- Finally, every organization is different!



9



Achieving goals

- Organizational change **requires champions**
~> at both the technical level and the managerial level ([Garcia, 2010](#))
- Should include representatives of both primary affected groups – Dev and Ops
- Responsible for preparing the business case



Costs

- The costs associated with DevOps are partially **continuing** and partially **one-time** costs
 - The **continuing** costs are associated with **tools** and **people**
 - A **one-time** cost is the **expense** of the **introduction** of DevOps practices and the **modification** of existing systems to **support** DevOps practices



Stakeholder Impact

- Internal Stakeholders ~> Dev and Ops
 - Dev group gains additional responsibilities and control
 - Ops group loses responsibilities and control, and
 - DevOps role will be new
- We discussed **additional** and **shifting responsibilities** in terms of the five categories of DevOps processes we identified in lecture #2



Stakeholder Impact

- External Stakeholders
 - What's the **main goal** of the DevOps practices?
 - Business and management stakeholders need to understand that they are **making a tradeoff** when adopting DevOps deployment practices
 - They are **giving up the visibility** afforded by a formal release process in order to **achieve faster cycle times**.



Risks and Their Mitigation

- The risks associated with the introduction of DevOps practices are both organizational and technical.



Organizational Risks

- Breaking down **barriers** between Dev and Ops
 - Barriers exist because these two organizational units have **different missions**, **different cultures**, and **different incentives**
- The creation of a **new role** of DevOps engineer also causes **stress** within an organization
- Placing scripts and configurations **under version management** and **controlling how new versions of systems are deployed**
- One suggested solution to **mitigate these risks** is to **adjust the key performance indicators** (KPIs) of **each group** to reflect overall rather than individual success in deployment



Technical Risk

- What **changes** are **required** to existing production architectures of applications?
- How is the **integrity** of the production database going to be **maintained**?



16



Changes Required to Existing Production Applications

- State management
 - Components should be stateless if at all possible
 - Stateless components are more resilient to failure because replacing a failed component is not difficult
- Feature toggles
 - If a rolling upgrade deployment model is being used, then feature toggles should be used to control new features
 - A feature toggle manager should be introduced to control the feature toggles.



Maintaining the Integrity of the Production Database

- The **integrity** of the **production database** can be **compromised** in one of two fashions:
 - Data from a test can be **mistakenly included** in the production database
 - A deployment into production **compromises** the database
 - rollback/roll forward plans to **correct erroneous data**



Rollout Plan

- You could implement a “big bang” delivery where everything is done at once, or an incremental delivery where practices are introduced
 - DevOps maturity model
- In lecture #2, we identified **five different aspects of DevOps**. They can be used as a **guide** to rolling out a **set of DevOps practices**.

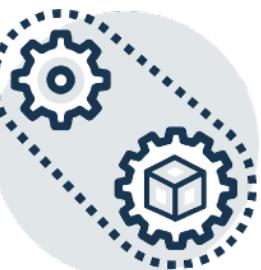


Success Criteria

- The success criteria are based on both the rollout plan and the rationale for adopting DevOps
- The rollout plan provides metrics for the categories in each of the periods identified.
- Focus your efforts on metrics that you can collect, both before the introduction of DevOps and after.



Measurements and Compliance to DevOps Practices



21



Measurements and Compliance to DevOps Practices

- Measurement should be designed with specific goals
- What kinds of measurements that are of interest to the business with respect to DevOps and its adoption?
 - How well the DevOps practices are succeeding?
 - What are the cases of noncompliance to DevOps practices?
 - What is the level of stakeholder satisfaction with the DevOps practices?



Measuring the Success of DevOps Practices

- The goals of the DevOps practices and the stages of the pipeline dictate the **types of measurements** that should be taken
 - the **times from commit to production** and **from error to fix**



Reducing the time between commit and deployment

- The deployment pipeline at each developer spending time in queue
 - At the continuous integration server, measurements should be taken of the number of branches active over time, the time between the creation of a branch and its merge into the trunk, and the time it takes to run tests
-
- The diagram illustrates a software development pipeline. It starts with a developer icon (a single person) and a 'Pre-commit Tests' box, which leads to a 'Commit' box. This is followed by a diamond decision point marked with an 'X'. If the path continues, it goes through 'Build and Integration Tests', 'UAT/Staging/Performance Tests', and 'Deploy to Production'. Finally, it reaches a 'Promote to Normal Production' step, indicated by a circle with a plus sign. A second developer icon (three people) is shown with a similar workflow, suggesting parallel or sequential developer cycles.

Reducing the time between the discovery of an error and its repair

- Errors in this context mean errors in the production version of a service
 - Does the automation of the **various stages** of the pipeline **increase** or **decrease** the **number and severity of errors** that escape into production?
 - Secondly, has the time between **discovery** and **repair** of a problem **changed** as a result of introducing DevOps practices?



Measuring Compliance to DevOps Practices

- We identify two practices where compliance might be an issue
 - Launching VMs: An example of noncompliance with the practices is when an operator launches a VM from the console during some incident
 - Removing feature toggle code: At this point, an entry can be created in the issue tracking database that identifies the removal of the feature toggle code as an activity to be performed



Measuring Stakeholder Satisfaction

- **Short questionnaires:** Internal stakeholders can be asked to rate their satisfaction on a scale of, say, 1 to 5
- **Crises:** One of the goals of any process improvement effort is to remove the necessity for heroic efforts
- **Outages:** When an outage occurs, there is usually no time for performing a deep analysis of the cause
- **Inadequate lead time for events:** Events such as a security audit, a rollout of a change to an existing system, or an installation of a patch have the potential to be disruptive



Points of Interaction Between Dev and Ops



28



Points of Interaction Between Dev and Ops

- Several points of interaction occur between Dev and Ops that we have not yet discussed in detail
- The two points this section is concerned with are licensing and incident handling



Licenses

- A software license is a legal agreement governing the use or redistribution of software
- We identify three situations where both Dev and Ops are involved in issues associated with licenses
 - Expiration ~> Typically, the responsibility for renewing licenses lies with Ops
 - License unavailable ~> Some licenses are “floating licenses”
 - Software audit



Incident Handling

- Once an incident occurs, there are three possible cases:
 - The incident is clearly related to an application
 - The incident is related to a hardware or infrastructure failure
 - The cause of the incident is not clear



Summary

- Implementing DevOps practices requires management buy-in, which, in turn, requires champions who can convince management that DevOps practices are of benefit
- A business case for DevOps covers costs, benefits, risks and their mitigation, a rollout schedule, and success criteria
- Once a DevOps adoption process is under way, it is important to measure the success of the adoption, the compliance with the associated practices, and how well stakeholders are responding to the changes to their environment



For Further Reading

- You can find more information about business considerations at
 - The blog “[DevOps Considerations](#)”
 - The book [Communications Networks in R&D Laboratories](#) (Allen,1970)
 - Wikipedia’s entry on [change management](#)
- Maturity models for DevOps, “[A Continuous Delivery Maturity Model](#),” July 17, 2013, InfoQ
- For understanding more about measuring rework, [Damon Edwards’ article](#) is helpful

