

Survey on Challenges in DevOps

Mr.Veeranna Kotagi, Mrs. Swathi .K, Dr. Piyush Kumar Pareek

Abstract — DevOps could be a culture, movement or observe that emphasizes the collaboration and communication of each computer code developers and different information-technology(IT) professionals whereas automating the method of computer code delivery and infrastructure changes. It aims at establishing a culture and atmosphere wherever building,testing, and emotional computer code will happen speedily, frequently, and a lot of dependably. This paper speaks regarding technology of DevOps and challenges and probable solutions .

Index Terms — DevOps, collaboration & code delivery.

I. INTRODUCTION

There square measure metrics to be found in any respect stages of the DevOps pipeline. It's vital to understand that of those metrics goes to be helpful to you by reviewing your existing processes. So as to understand if your DevOps practices square measure having a positive impact, you would like an honest place to begin to live against.

From a business perspective, you must shrewdness typically you're deploying to production. you must shrewdness several of the deployments have resulted in outages or bugs with a measurable impact on the user base.

You should grasp the typical time it takes your team to live through outages. you must perceive, at-a-glance, what your up-time is, and if you're meeting any SLAs that you just could also be certain to. There square measure many extra business level metrics value trailing, though that'll be one thing every company usually clarifies for themselves and their groups.

The technical aspect of DevOps can worth totally different metrics. Knowing however long your CI method takes is vital. the typical latent period of your REST services or the quantity of synchronic users at any given time represents helpful information that will amendment the method developers solve for specific issues.

Knowing however code is acting on the servers permits your engineers to quantify the impact code changes wear performance. This dovetails into understanding however your production servers square measure playing, and if

you're over or below provisioned. Your operations team ought to have all the metrics they have to make sure that they're running the foremost elastic and secure infrastructure do able.

This religious doctrine of "quantify everything" may be a bit nebulous, as a result of the quantity of information is very large and growing in any respect totally different levels. Knowing what to trace is crucial to any prosperous DevOps arrange.

In fact, if you're unaccustomed DevOps, here square measure a number of key performance indicators you must track,

To urge you started:

1. Frequency of deployments
2. Frequency of unsuccessful deployments
3. Mean time to recovery (MTTR)
4. Mean time to discovery (MTTD)
5. Lead time
6. up time

Customer criticism volume Service performance Start by capturing the maximum amount data concerning your current method as doable. Once you are feeling you have got an honest handle on your current metrics.

you'll have one thing to live your DevOps efforts against. The journey of k miles begins with the primary step. Once you have got a heavy understanding of your metrics, the DevOps philosophy becomes more and more clear. There's a significant art gap at a downtown repository and group action is five-hitter of projections. One thing is wrong. Imagine a Google Satellite image on your screen. What you see appears like a grey block. You zoom out and realize a building. Zoom out additional, and block becomes apparent. Additional and you notice the bridge over a stream is blocked by an accident and traffic is stalled. You've known the matter and perceive that parties should join forces for a good answer. The police ought to work with hearth in clearing the accident and caring for any probably separated parties. The town road groups should examine the bridge for additional injury, just in case structural damages occurred. The art gap ought to expand hours or re-schedule. The DevOps philosophy supports this sort of research, discovery, cooperative action.

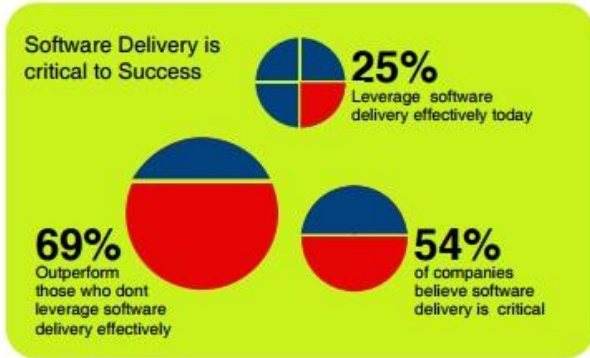
Manuscript received September 15,2016

Mr.Veeranna Kotagi, Computer science & Engineering, Visvesvaraya Technological University Belagavi / East West College of Engineering /Bengaluru/India/9986144592

Mrs.Swathi .K , Computer science & Engineering, Visvesvaraya Technological University Belagavi/K.S.Institute of Technology,Bengaluru/India/9742569970

Dr.Piyush Kumar Pareek, Computer science & Engineering, Visvesvaraya Technological University Belagavi / East West College of Engineering/Bengaluru/India/8095372283

It doesn't matter whether you are in Cloud, Enterprise or Mobile. For each one of you, **stable software delivery on time** is the key to your business success.



Some of the serious issues blocking your software delivery are:

- Building and maintaining servers – Time consuming and unproductive
- No environment management - Differences in development and production environments
- Slow deployments – Costly error prone manual process and efforts
- No shared ownership - Lack of feedback and proper metric leads

Source: <http://www.happiestminds.com/whitepapers/devops.pdf>

- No proper configuration management - Discrepancies in managing configurations
- Deployments are a blocker - Upgrade risk due to manual management of multiple application configuration and versions - Dependency on specific deployment engineer
- Production downtime - Due to lack of improper deployment instructions / checklist
- Hacking – Fixing directly in production (instead of a proper hotfix process) and forgets to check-in into source control



Realize your entire business gets impacted if you do not have Continuous Delivery. To enable that, you must adopt DevOps

Source: <http://www.happiestminds.com/whitepapers/devops.pdf>

II. LITERATURE SURVEY

Because DevOps may be a cultural shift and collaboration between development, operations and testing, there's no single DevOps tool, rather a collection or “DevOps tool chain” consisting of multiple tools. Generally, DevOps tools match into one or a lot of those classes, that is reflective of the package development and delivery method

- Code – Code development and review, continuous integration tools
- Build – Version management tools, code merging, build standing
- check – Test and results confirm performance
- Package – physical object repository, application pre-deployment staging
- Release
 - amendment management, unleash approvals, unleash automation
- Configure – Infrastructure configuration and management, Infrastructure as Code tools
- Monitor – Applications performance observance, user expertise

Though there square measure several tools offered, sure classes of them square measure essential within the DevOps tool chain setup to be used in a corporation. Some tries to spot those basic tools is found within the existing literature.

Tools like manual laborer (containerization), Jenkins (continuous Integration), Puppet (Infrastructure as Code) and Vagrant (virtualization platform) among several others square measure usually used and regularly documented in DevOps tooling discussions.

Organizations that have adopted agile software system development square measure seeing increasing quantities of releases. DevOps was basically born from this increasing quality of agile development. Agile and DevOps square measure similar, however disagree in a very few necessary aspects. Agile represents a amendment in thinking, whereas DevOps truly implements structure cultural amendment. One goal of DevOps is to determine Associate in Nursing surroundings wherever emotional a lot of reliable applications quicker and a lot of often will occur. unleash managers square measure setting out to utilize tools like application unleash automation and continuous integration tools to assist advance this goal, doing therefore through the continual delivery approach.

Continuous delivery and DevOps square measure similar in their meanings and square measure typically conflated, however they're 2 totally different ideas. DevOps includes a broader scope, and centers round the cultural amendment, specifically the collaboration of the assorted groups concerned in software system delivery (developers, operations, quality assurance, management, etc.), still as automating the processes in software system delivery. Continuous Delivery, on the opposite hand, is Associate in Nursing approach to alter the delivery facet, and focuses on delivery along totally different processes and capital punishment them a lot of quickly and a lot of often. they need common finish goals and square measure typically utilized in conjunction to attain them. DevOps and Continuous Delivery share a background in agile strategies

and lean thinking: little and fast changes with targeted worth to the top client. They are well communicated and collaborated internally, so serving to accomplish pace to promote, with reduced risk

Previously IT infrastructure was a service revealed to the remainder of the organization. The systems were organized and maintained by operations. If there was a replacement request, the request was created, so dead by the operations team.

In the nowadays of users and development the method may be a bit a lot of subjective and cooperative. This is often a decent issue, however shifts the dynamic and responsibilities. Development users demand a lot of, and are adopting tools at an almost un-manageable pace. Business users are a lot of technically savvy, understand what they require, and sometimes can try and address challenges on their own to boot their productivity atmosphere has created a significant move to mobile devices, that are self-contained and hungry for enterprise-wide cloud services.

A. Breaking down language barriers

Development and operations read the globe from totally different views. Though they're each involved with equivalent outcomes, their approaches are totally different, like 2 sides of an equivalent coin (in my opinion this same categorization exists between finance and accounting).

These totally different views have diode to refined however necessary variations in nomenclature, just like the definition of a "full stack." Take the time to tease out these variations and establish a typical lexicon for all of IT.

B. Overcoming the developer vs. operations mentality

Selling DevOps to developers isn't too troublesome – for development it's simple to envision the advantages to creative thinking and innovation created potential through DevOps. Operations, on the opposite hand, views DevOps with suspicion, basic cognitive process it's Associate in Nursing tried development coup to change developers to continually dump buggy, untested software package into production. This is often a minimum of somewhat secure, as Ops are the oldsters World Health Organization have to be compelled to trot out outages and repair disruptions on an everyday basis. They see amendment because the enemy, however in keeping with New Relic, "The trick is to indicate operations professionals that quicker preparation cycles, with considerably fewer changes per cycle, means issues are fewer and so abundant easier to spot and rollback if necessary." Fewer changes per unleash suggests that additional management for operations, though those releases happen additional of times. Fewer changes conjointly build it immensely less advanced to perform root cause analysis and troubleshooting once problems do occur.

C. Learning new skills

There is a preferred sports byword that goes, "There isn't any comfort within the growth zone, and there's no growth within the temperature." DevOps needs amendment.

Amendment needs adaptation. Adaptation needs learning new skills.

whether or not it's a sys admin learning to create new tools, or developers breaking out of a falls existence, rest assured that DevOps can firmly place your organization into the "growth zone."

D. Work disruption

DevOps would require new workflows, that if enforced can cause a trickle-down result throughout the organization. Current tools can become obsolete, and investments in new tools are going to be necessary to support new workflows. New and additional advanced workflows can have implications for regulative compliance and security problems. It's necessary that the DevOps implementation team incorporates abroad and deep ability set, which all neutral interests ar portrayed throughout designing. it'll need aholistic read of the complete price chain for software package delivery, from conception to observation in production. An honest exercise for the implementation team is to perform value-stream mapping for current and projected processes, and to overlap those maps to spot areas which can cause friction.

E. Management responsibilities

DevOps can shift the facility balance within the organization, in a way or another. It's inevitable, however not essentially insidious. What's necessary is knowing World Health Organization can have final authority over selections, and putting in the checks-and-balances necessary to make sure that the voices of every neutral body are portrayed and detected within the method} process. Trust is of crucial importance among the management team, borne out of a shared belief that DevOps can improve the lives of all concerned. Solution

F. Collaborate

Most additional will get done once these problems are mentioned in Associate in Nursing open forum. The fashionable testified organizations follow a oldster based mostly communication strategy that may now and then be a trifle tense, however opens the doors for all groups to be told from one another and counsel solutions, or raise problems. Break the method chain for deciding and switch it into a flow of thought and ideas. The sole trick is keeping the concepts to specific sessions, and not every which way throughout work activities.

G. Empower

Let groups choose their own tools, however request that they report them to you. Conjointly think about the library of tools approach, wherever operations vets tools before adoption, and gift to any or all groups "the library of tools you'll be able to choose from." they will be at liberty to counsel new tools, however will solely choose from the list. The sole downside is that the new burden on that Operations to stay current.

H. Speak Accounting

the problems of chargeback and OPEX have to be compelled to be self-addressed thus it doesn't hinder tool adoption. There are such big amounts of ways that to try to to this and most means on the far side my ability set. however if IT helps finance perceive what's happening within the world of technology licensing vs. cloud service they will means the shift, and facilitate them intern with budget reportage, and bottlenecks.

I. Break the immutable Infrastructure

whereas there may be a giant direct effort, it's just once, you'll be able to break the shackles that previous infrastructure has created for you. The trick is to leverage advanced practicality of virtualization tools like changing bequest systems to VMs and keeping them on-prem or unifying them with the IaaS servers. On-Prem virtualization will enjoy snapshots of VMs in their running states and connected clones to permit development work against giant business systems ,however on precise duplicates instead to eliminate the priority of breaking one thing. indeed during this situation you're inspired to hie thus quick that you just can break one thing. It won't impact anyone else. And for those nasty networking layers, leverage the powers of recent SDN technology, versus significant vLAN and physical devices.

And Finally, Analytics first: Build analytics and observation into everything, and force your developers to try to to an equivalent. By adopting a sturdy analytics platform, wherever all knowledge is keep for applications and new tooling, operations will be aware and have visibility into all systems, though it's out of their management. And today's log analysis platforms will alert operations of anomalies within the entire system with none extra effort on their half. Keeping their visibility while not preventive adoption.

Challenges aren't unaccustomed operations groups, however the pressure behind the increasing rate of amendment may be. And currently that users have alternatives with shadow IT, it's even an even bigger downside. but if you scrutinize shadow IT as a list of tools your organization wishes, and a scream fostering, you'll be able to facilitate. And if you leverage analytics to guide and manage the trouble, you'll be able to become a helper, at an equivalent time maintaining all needed operations practices.

III. CONCLUSION

It can be observed that few challenges are being faced in devops which can be overcome by suggested methods. Furthermore operational challenges will be addressed in further research papers.

ACKNOWLEDGMENT

Firstly we would like to thank Honorable Sri. C. N. Ravikiran sir Chairman , to achieve academic excellence on par with Global Standards and management committee members for their moral and financial support
We would also like to thank our beloved principal Honorable Dr.B.R.Ramesh sir, for his continuous support

and guidance for carrying out research related activities in our college.

We would also extend our thanks to all faculties of East west college of Engineering , Bengaluru for their kind support.

REFERENCES

- [1] Loukides, Mike (2012-06-07). "What is DevOps?". Floris, Erich; Chintan, Amrit; Maya, Daneva (2014-12-10). "A Mapping Study on Cooperation between Information System Development and Operations".
- [2] Samovskiy, Dmitriy (2010-03-02). "The Rise of DevOps". Fubaredness Is Contagious.
- [3] Kim, Gene. "DevOps Culture Part 1". [b Lyman](#), Jay. "DevOps mixing dev, ops, agile, cloud, open source and business". 451 CAOS Theory. Turnbull, James (Feb 2010). "What DevOps means to me...". Kartar.
- [4] Debois, Patrick. "Agile 2008 Toronto". Just Enough Documented Information. Retrieved 12 March 2015.
- [5] Debois, Patrick (2009). "DevOpsDays Ghent". DevopsDays. Retrieved 31 March 2011.
- [6] Debois, Patrick. "DevOps Days". DevOps Days. Retrieved 31 March 2011.
- [7] Gartner Market Trends: DevOps – Not a Market, but Tool-Centric Philosophy That supports a Continuous Delivery Value Chain (Report). Gartner. 18 February 2015.
- [8] Edwards, Damon. "Integrating DevOps tools into a Service Delivery Platform". dev2ops.org.
- [9] Seroter, Richard. "Exploring the ENTIRE DevOps Toolchain for (Cloud) Teams". infoq.com.
- [10] Theakanath, Thomas. "DevOps Stack on a Shoestring Budget". devops.com. "Stronger DevOps Culture with Puppet and Vagrant". Puppet Labs. Retrieved 2015-10-22.
- [11] Ambler, Scott W. (12 February 2014). "We need more Agile IT Now!". Dr. Dobb's The world of software Development. San Francisco: UBM.
- [12] Best Practices in Change, Configuration and Release Management (Report). Gartner. 14 July 2010.
- [13] Hammond, Jeffrey (9 September 2011). "The Relationship between DevOps and Continuous Deliver". Forrester Research. Forester.
- [14] Humble, Jez; Farley, David (2011). Continuous Delivery: reliable software releases through build, test, and deployment automation. Pearson Education Inc. ISBN 978-0-321-60191-9.
- [15] "What is DevOps?". NewRelic.com. Retrieved 2014-10-21.
- [16] Nasrat, Paul. "Agile Infrastructure". InfoQ. Retrieved 31 March 2011.
- [17] Wadhera, Anila (June 6, 2016). "10 Key DevOps Practices to Improve IT Efficiency". TO THE NEW. Retrieved September 2, 2016.