

DevOps Best Practices 2020



Table of Contents

ABOUT THE GUIDE	3
COMPANIES THAT TRUST INEDO	4
AXIAN	5
DEVOPS GROUP	10
HAPPIEST MINDS TECHNOLOGIES	18
MBCTG	22
QBURST	27
RELEASETEAM	32
ROCKETMAKERS	36
SCIENCESOFT	39
LEARN MORE	41



About the Guide

We assembled this guide because implementing DevOps at your company can be overwhelming. Whether you're thinking about hiring a consultancy for help, are just starting to learn about taking on DevOps at your organization, or need advice for improving your current processes, the expert insights in this guide can help development team leaders who are looking for guidance and evaluating DevOps solutions.

We interviewed DevOps Best Practices that help companies either make the transition to DevOps or to optimize their DevOps processes. They were all asked individualized questions, and the participants spent a lot of time answering them to provide high quality guidance to readers.

In this guide, these companies provide actionable advice on how to make the move to DevOps and some of the best practices for important DevOps topics like hybrid cloud implementation, selecting the best DevOps tools for your business, assessing progress with the right metrics, and adopting the cultural aspects of DevOps.

We did the preliminary research for you, because we want to see more companies adopt DevOps and provide more value to their customers and clients, all with the people and tools you already have. Through this guide, you'll get to know the consultancies and how to contact each if you see the potential for successful collaboration. We're confident that this guide will be a valuable resource as you start your DevOps adoption journey.





Companies That Trust Inedo





Axian

www.axian.com/



Organizations are increasingly moving to the cloud, but struggling to know how to integrate it into their DevOps pipelines. What are some best practices for how to get those efforts off the ground?

Any time a business decides to make the move to a cloud-based solution, they should always have a roadmap and milestones. Do not make your milestones too lofty, and focus on getting the smallest part of your product that provides good value. Every cloud resource should be able to be deployed using automation. Make sure that you can deploy multiple environments (i.e., CI, QA, staging, production) and identify any manual steps, document them, and finally automate them. You will want to also implement an automated build, test, and deployment pipeline so that all pull requests merged to master, are deployed automatically to a continuous integration environment (CI).

What are some of the ways poor cloud software can harm companies? How should they go about evaluating if their current software is adequate to meet their needs?

Poor cloud software can incur hidden costs, one of which is paralyzing development due to the fear of breaking something in production. This fear happens due to several factors: lack of automated CI/CD, insufficient unit tests and integrations tests, tightly coupled services, inability to easily provision new cloud environments, lack of clear architectural vision, and product feature backlog churn. Before making the jump to the cloud, businesses should take a look at their current processes, and identify anything in their DevOps practices that is missing or needs to be improved.



What are some of the key things for organizations to keep in mind when they are creating a plan for cloud migration?

When migrating to the cloud, organizations need to consider several key factors.

1. The organization should look at their current technical expertise, and determine if a particular cloud provider best aligns with that expertise.
2. They also should determine what, if any, on-premises infrastructure is required for a hybrid cloud configuration.
3. The migration plan should identify all of the costs associated with migrating to the cloud.
 - a. Development costs
 - b. Average monthly cost to run the cloud infrastructure
 - c. Ongoing maintenance personal costs
4. Create an inventory of all hardware and software systems that will be affected by the cloud migration.

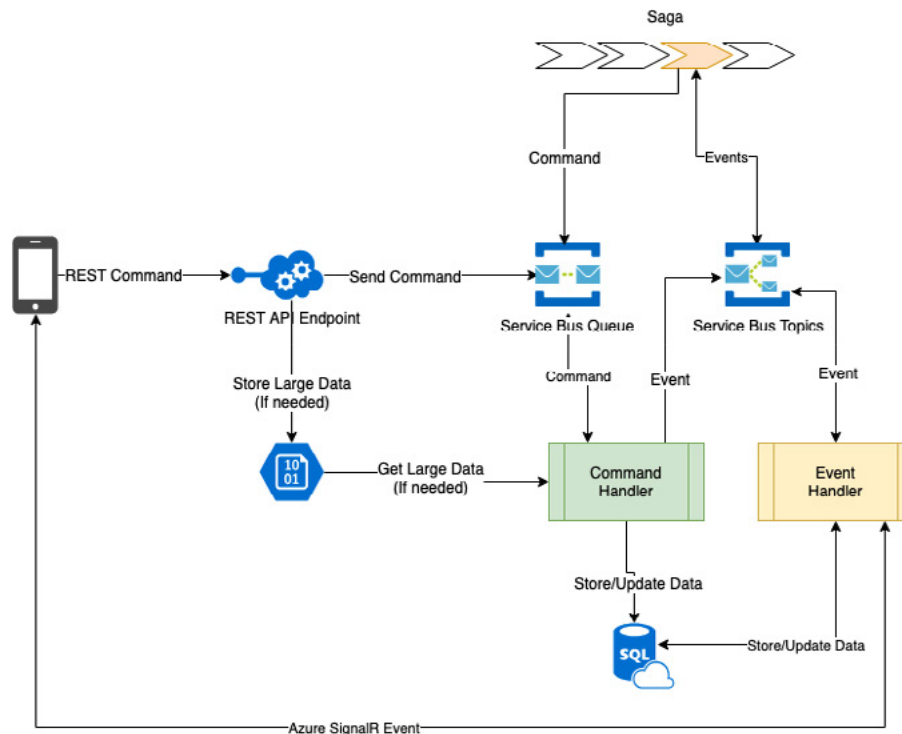
When it comes to a hybrid cloud setup, what are some of the key things organizations need to get right in order to get the most utility out of their new infrastructure?

When an organization decides to go with a hybrid cloud strategy, they should consider several factors:

1. The reliability and simplicity of the interconnections between on-premises and cloud resources. Every in-bound and out-bound connection should be very reliable. A robust monitoring and alerting system should be setup to notify when an issue arises.
2. Prefer asynchronous commands and events over synchronous connections.
3. Measure the performance of resources before and after migrating resources into the cloud

In terms of customization, can you give me an example of a custom setup or feature that solved an organization's unique problem?

When asked to implement a platform for a fast-growing identity provider startup, it was decided the that the platform should support a decoupled approach from the ground up. In order to support an asynchronous communication layer, a cloud-based service bus was used. Using SignalR, we could send notifications to clients when particular commands or events were completed (or failed) on the backend. The Saga concept ([watch the YouTube video here](#)) was also implemented, this allowed the business logic to be centralized, and easily changed as business needs changed. The development process was trunk-based, and every commit on every pull request was built and tested. We made sure that the cloud-based environments were simple to provision and configure, and were isolated for each other. We did spend several weeks to refine the CI/CD pipelines, but by the end of the project, it was time well spent. There were about 10 different environments by the end of the project CI, Mobile-CI, Demo, Demo-Sales, Staging, Production, and several developer environments. Having dedicated demo environments allowed for reliably performing sprint demos and in parallel performing demos for the client's prospective customers as the application matured.



From a security perspective, what are the best practices organizations moving to the cloud need to keep in mind to avoid any headaches down the road?

When moving into the cloud, try to avoid custom security solutions. Every major cloud provider has solutions for user management, authentication, authorization, encryption at rest, transport encryption, and API security. Use the security offerings available from your cloud provider. Also, you should keep the security concerns separate from your source code, security should primarily be configured and controlled by the Operations team. All stored passwords, certificates, and keys should be rotated on a regular basis.

You mention the need for sustainable results. Are there any red flags organizations can watch out for that will indicate something about their approach to DevOps needs to change?

Organizations should always track the amount of time it takes for a feature to make it into production, this is known as cycle time. When you start to notice an increase in cycle time, you should try to identify what area(s) of the development process are slowing down. It could be how long it is taking QA to validate features, or it could be due to inadequate backlog story definitions, or it could be due to developer turn over. Once you have identified what areas of the development process are causing the slowdown, you should come together as a united team to try and fix the issues. If it is due to poor quality stories, then you should be more intentional about spending more time during backlog grooming and refinements sessions. If it is due to QA slowdown, work with QA to find out how the team can come together to help QA validate features more efficiently (perhaps additional automation could fix the issues).



Also, always be aware of relationship between the “Dev” and the “Ops” teams. They should always have a good relationship and be wary of the us and them mentality. Optimally both teams are under the same management hierarchy.

The Axian website talks about “cultural adoption”. What are some of the common cultural attitudes you see across organizations implementing DevOps effectively?

Organizations need to have a top-down DevOps buy-in to successfully implement a DevOps culture that consistently delivers value to the customer with low cycle times. Oftentimes, if one or more people or departments do not buy-in to DevOps, there will be friction, and an inability to ship features with low cycle times. If management or executives are not able to understand the DevOps philosophy, then it is up to middle-management to “train up” to the executives to teach them the why’s, how’s, and how comes of the DevOps approach. This train up approach can also be facilitated by developers and other individual contributors, who can relay to executives the DevOps justification for why adherence to certain DevOps practices is necessary for the overall health and success of the organization.

How do you suggest companies that don’t have a DevOps culture being building one?

If your organization lacks a proper DevOps culture, do not try to change the world over night. Start with baby step goals. Create a personal vision of where think the culture needs to get to, and the required milestones to get there. It is helpful to formalize this vision into a document. Create goals that adhere to S.M.A.R.T. (Specific, Measurable, Attainable, Relevant, Time-bound). For developers, I would recommend working on the practices listed below (roughly in this order):

1. Kill your ego and esteem others greater than yourself
2. Implement trunk-based development, avoiding long-lived branches (more than 1 day old)
3. Protect your main branch and require pull-requests to facilitate discussion and code reviews
4. Start writing good backlog stories, that have clear acceptance criteria
5. Prioritize your backlog
6. Identify your product owner (if no has this role, then talk to management to assign this role to someone)
7. Identify your *service boundaries*
8. Protect your backlog from feature churn. All feature requests must go through the product owner and prioritized
9. Automate your builds for the trunk branch (build and unit test every commit)
10. Automate the deployment of the trunk branch every commit into a dedicated Continuous Integration (CI) environment
11. Automate the Execution of a suite of smoke tests against the CI environment
12. Automate builds for branches and pull requests (build and unit test every commit)
13. Commit to the process of decoupling your services



For managers, I would recommend facilitating the development practices listed above. That includes prioritizing the backlog to include this work in sprints. Managers and architects should also provide an overall DevOps vision that includes the following:

1. "Train up" to the executive teams regarding DevOps
2. Ensure that the architecture is loosely coupled
3. Empowering teams to chart their own course to success
4. Monitoring should be a high priority to facilitate more automation
5. Ensure that proactive notifications from production environments is implemented
6. *Build your teams* to look like small software organizations

If you could give companies one piece of advice about DevOps adoption, what would it be?

In my estimation, all companies so embark on the DevOps journey. It will reap manifold rewards, such as:

1. Greater employee job satisfaction
2. Less fear to make significant changes
3. Deliver value to customers in less time
4. Increase developer productivity
5. Improved inter-team communication
6. Greater employee cohesion
7. Corporate alignment, focus, and vision



Mark Johnson is a Solutions Architect that works with a variety of businesses to build custom software solutions, to solve challenging business problems. He believes that a holistic DevOps approach is necessary to build software solutions that are scalable and can evolve with the business needs.

Contact Us

Axian Inc.

9600 SW Nimbus Ave. Suite 200
Beaverton, OR 97008 USA

salesinfo@axian.com

503-644-6106 (phone)
503.643.8425 (fax)

Decoupling, Architecture and Teams
Udi Dahan - If (domain logic) then CQRS, or Saga?



DevOps Group

www.devopsgroup.com/



In your keynote describing your work with BAE systems, you stated that the first step towards DevOps adoption was to move to a product-centric delivery model for just a few products. Can you take us through the beginning stages of execution on that strategy looked like? How can organizations replicate that process?

The first step is to identify what the product is. Roman Pilcher, a well-known figure in the Product Management space states that a product is “something that creates specific value for a group of people, the customers and users, and to the organisation that develops and provides it.” In BAE Systems, I started with this definition being around applications which soon re-orientated to value chains. The final phase would have been to align to customer outcomes which we started experimenting with.

The next step is to establish a long-lived team around the product with all the roles needed to do the work. This team operated the “you build it, you run it” principle and used Amazon’s “two pizza team” rule of thumb. We assigned roles such as product owner, product delivery lead, product analyst, product technical lead and product engineers to the team. For many people this needed their role redefining and as a result we needed to unlearn what we knew and learn what we now needed to know.

As part of the re-organisation, I took my organisation (and myself as their leader) through a training and coaching programme on Agile and DevOps principles and practices. This included training followed by implementation supported by coaching and then a review with more training, implementation, coaching and so on. This enabled my teams to learn in small batches, apply and review - it was almost an Agile approach to training and coaching. I use this pattern now when engaging with new customer. DevOps requires a mindset shift and it takes consistency and perseverance to change and then to stop the rubber band from snapping back to the old ways.

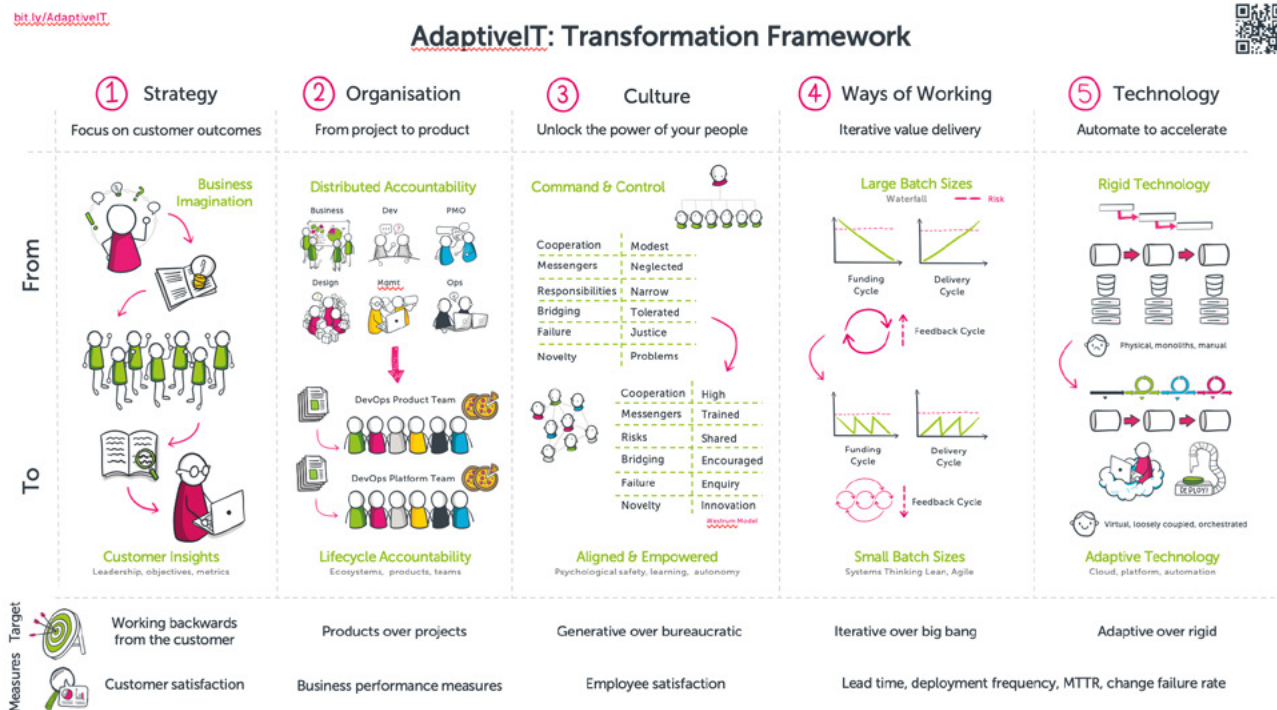
The above led to a new language forming with words like backlog and story entering the vocab and we saw the performance metrics change - initially this got worse as the teams address technical and process debt but then started to accelerate as the business and technology started working as one.



Reflecting on my journey and experience with a number of customers, I have now identified 5 areas that an organisation needs to address:

- Strategy: Working backwards from the customer
- Organisation: From project to Product
- Culture: Establishing a generative environment focused on psychological safety, learning and autonomy
- Ways of Working: Reducing risks through smaller batches and faster feedback cycles
- Technology: utilisation of modern technologies to build continuous delivery pipelines and cloud platforms to accelerate the delivery of new features whilst improving the stability of the product.

The transition from the current state to the future state is illustrated below:



Your keynote also mentioned the need for engineers to be “at the front” of organizations. Can you describe what you mean by that, why it’s important, and how DevOps principles enable companies to make it happen?

If you were to map out the information flows between users and engineers in a traditional enterprise, there would be many handoffs and dependencies. Software engineers rarely get to see how their code is working in production. Operations engineer run production but feel powerless because you cannot change or improve the service.



In Agile and DevOps ways of working the business and all IT functions work closely together for a common goal in small teams. If the team is building and supporting the product, engineers will be able to see how users are using their code. They will be regularly speaking to customers and will respond regularly to feedback, failures and learning.

War and Peace in IT by Schwartz paints this picture really well and IT people have been treated like they are not part of the business. "Business and IT" are separate entities and that technology people speak a different language and business should be protected from them. This is no longer true; IT is increasingly becoming the business and engineers can use the technical capabilities of cloud and automation to translate the language of the business into language of code almost instantaneously.

The role of the engineer is to translate the business model into code, hence the need to ensure they are engaged, aligned and empowered to achieve the company's goal.

Your consultancy places a big emphasis on DevOps culture as a way to "modernize ways of working". What does an effective DevOps culture look like to you, what are the similarities and differences across organizations, and how do companies go about creating one that works for them?

This is a big question and I think you are tapping into our secret sauce.

In Kung Fu Panda the "one" is able to open the Dragon Scroll to identify what the true power is and there is nothing in it - yet there is... a reflection. You are the Dragon King, you are the power, you are the secret ingredient.

I had a number of my own teams, I now work with a number of customers and I host DevOps Manchester and work with leading edge companies like Auto Trader and ao.com. Each organisation, each team has its own culture. But the culture has a common characteristic - the people trust each other. Trust is a key component.

How is trust created? Through relationship. How are good relationships formed? Through spending time together, sharing information, failing together and learning together.

Westrum in his topology of organisational culture shares how different organisational cultures share information in a safety environment. A generative culture will embrace bad news and learn from it. It will encourage open flow of information and sharing, risks are shared, and novelty is embraced.

We try to take the team and the organisation (including the leadership) through a journey where they understand and align to common goals and measures underpinned by a deeper understanding of the "why" they exist. We then show them what their work is, the complexity of their environment and how



work flows across the team(s) in order to deliver value to the customer. By this time, mindsets and hearts are changing - they can see the hamster wheel they are running in and recognise the need to change to move forward. We then roll up our sleeves and come alongside them to training and coach them in new ways of working, whilst engaging with leadership to create the right environment for their teams to flourish.

After the initial pain and a few setbacks, the teams start forming a culture of accountability, alignment, sharing and customer-focus - but each one is different. A true leader will know how to grow different plants in the same garden.

Your website cites statistics from the State of Agile Report that 44% of respondents stated “Inadequate management support and sponsorship” as an issue that hinders DevOps. What are some key things management can do to ensure they don’t become the constraint to adoption?

“Where’s the plan”, “Show me the Gantt Chart”, “Where is the business case for the project”, “Have you been through the lifecycle governance process?”, “I want to see the people organised in plan, build and operate functions”, “Don’t show me the MVP, when will I get all the features”, “I don’t want to see people learning on the job, they can go on a training course if they need to”, “Does the operational readiness review identify all the checks and balances a team needs to fulfil to deploy to production?”, “How many more resources do you need to get this thing live?”, “Where is the one throat to choke (or belly button to press)?”

We have all heard some or all of these statements before. Leadership need to embrace the change almost before the rest of the organisation. A team may be able to start small and come together to deliver value in a small way but without leadership sponsorship. That’s like starting a fire in a building fitted with smoke detectors and a sprinkler system – the initiative will get put out.

A senior leader and a business sponsor need to switch the sprinkler system off - even if it’s just for one floor or one room to start with. I mentioned this in my recent articles, the DevOps Playbook where I stated, “how could I be part of the solution, when I was part of the problem?”. I needed to change. My boss needed to change.

What can management do? They need to start reading, they need a leadership coach and they need training and go on the journey with their teams. It’s like Alcoholics Anonymous though: you need to admit you have a problem and then don’t be afraid to ask for help. This is what I did. The list of books I read is on Goodreads. For leaders I would specifically recommend; Phoenix Project (Kim), Lean Start-up (Ries), Radical Candor



(Scott), War and Peace in IT (Schwartz), Team of Teams (McChrystal), The Goal (Goldratt), A Seat at the Table (Schwartz), Turn the Ship Around (Marquet), Start with Why (Sinek), Powerful (McCord), Project to Product (Kirsten), Good to Great (Collins), Legacy (Kerr), Scrum (Sutherland).

The initial challenge for many enterprise organizations is breaking up the legacy monolith system. What are some of the recurring issues you see at the start of this process? How do you tell your clients to overcome them?

We see this a lot and sometimes it's in two parts - monoliths and technical debt. The monoliths tend also to be connected to an eco-system of other systems (including other monoliths). This makes change very hard and the ops team are striving just to keep these systems running and they resist any change. Meanwhile the business will have spun up multiple projects all heading to the same monolith. There are better words for this, but it can be a bit of a train wreck!

What do we advise? Like with anything, start small. Can the internal components of the monolith be treated like a sub-system or a module, can that be isolated at all? Can you align the work, the team and the customers to these modules, essentially creating mini value streams? Can some of the work then be delivered in smaller batches without affecting other sub-systems and external interfaces?

At some point the answers to these will be no. Then we need to move towards a technical strategy and associated capabilities to help us. One of these strategies is to develop a comprehensive automation test framework that conducts a full regression and integration test - this will allow work to be broken down into smaller batches and accelerated. Another strategy is to establish an API layer that forms between the different systems in the enterprise and this creates a capability that allows you to change the monolith without impacting the wider eco-system. It also enables you to start moving workloads/modules/sub-systems onto smaller applications or micro-services outside of the monolith - in time this will either remove the monolith altogether or reduce the monolith to a much smaller legacy footprint. This is also known as a strangler pattern.

On the IT side of things, technical debt is something a lot of enterprise organizations struggle with when breaking up a monolith. What are some best practices for clearing that debt while making the move to microservices?

The best way is to give the teams time. Once they are aligned to a common goal, integrated as a multidisciplinary team and are visualising and measuring the work, they will be able to see their constraints. Too many incidents, a backlog of problems, change is hard, break other components of the service, etc. will impact their flow. Allow the team to focus on the Four Types of Work identified in the Phoenix Project by Gene Kim and then prioritise "internal projects" just like any other work. The team will go slower before they go faster, but that's ok - leadership need to provide that environment



Are there any challenges you see your clients constantly overlooking or minimizing when adopting a full DevOps approach? How can companies work to discover their own blind spots, so they aren't surprised with a problem during the transition process?

I would say the key blind spots include:

- people firstly start with the tools and technology over the people and culture. These need to be done together not separately.
- leadership will want the organisation to change and not consider themselves
- management will try and change the layout of the house without changing the foundations - the governance processes, financial constructs, management structures, KPIs, employee incentives, etc. need to change alongside the ways of working and technology capabilities.
- they will try and do a big bang change over starting small, learning and growing.

Changing the organisation is now more like Candy Crush than CMMi - it's about continuous improvement with no apparent end. We work with customer who are trying to start the game and with customers who need help going from Level 2 to 3 (or 200 to 300).

Talk to us about DevOps tools. What role do you believe they play in the broader cultural DevOps landscape? What are some best practices organizations can follow to get the most out of them?

I love the book Good to Great, Collins, in which we state that technology is a momentum accelerator not a momentum generator - build the flywheel and then use technology to accelerate. We get this wrong all the time.

I started with a wall, some magic whiteboards and some post-it's to create a Kanban board with WIP limits supported by some basic Agile practices and monitored through a scorecard and a cumulative flow diagram. In any other craft, we start small, we start manual - we learn before we use more industrial tools and advanced techniques. We get this wrong in business all the time - People and Interactions over processes and tools - Agile Manifesto.

The tools come later and there may be some tools that the company should mandate but there are a lot they should leave to the teams to figure out and use. Getting the balance is sometimes hard but the principle is start slow, start manual, build understanding, practice, master, automate.

That said, there are a number of organisations that I have worked with who may need to alleviate some of their biggest constraints using technology. Capabilities such as APIs, cloud platforms, automated testing and continuous delivery pipelines can become essential in making working that would take months to minutes.

You've done some work with the insurance sector, specifically. How does DevOps help legacy



companies fend off upstart disruptors? And what are some of the DevOps practices you have found are particularly important to achieving that goal?

The insurance sector is facing some big challenges and I have spoken with leaders in the field, attended roundtables and been to specific conferences. The traditional organisations need to be more responsive with the ability to deliver products to their customers, who now demand more, want more personalisation and have greater expectations of what will bring them “peace of mind”. This is not restricted to the B2C firms but applies to the B2B firms as well (although less aggressively).

A key part to all this is data and being able to use their data as they engage with customer but also use this data to form new products and services. The ability to manage data requires the right digital platforms and should exercise the use of Machine Learning and Artificial Intelligence. Firms know they have a long way to go and we are seen traditional organisations bring IT closer to the business (as part of the business), re-structure their IT sourcing contracts and in-source in some areas and are create small multi-functional teams who deliver business outcomes whilst harnessing the power of technology - not in a “big bang” sort of way but more in an evolution of technical capability over time and in response to the needs of the user/customer.

This is where product-aligned teams, Agile ways of working, Cloud platforms and Automation combined bring the capabilities needed to compete and win in this vibrant marketplace

If you could give one piece of advice to an enterprise manager or team about to embark on a DevOps adoption journey, what would it be?

Ok, I tend to see a version of me in most of the leaders I meet as I was starting my journey - “Where do I start?” So, if I was speaking to myself 4 years ago:

1. This is going to take a few years, buckle up!
2. It's going to get worse before it gets better, have faith!
3. Get exec sponsorship, else you are going nowhere fast.
4. You can't do this alone, find Yoda! (or Erik if you have read the Phoenix Project) – DevOpsGroup were my Yoda...
5. Get the right people on the right seats on the bus and then figure out where to go, build great teams - priceless!
6. Become the evangelist in the organisation, read, learn, share, apply, repeat
7. Face the hard problems head on, (wo)man up.
8. Don't be afraid, have courage - be bold
9. Believe in yourself and surround yourself with the right ‘generals’ - create a safe space
10. Give yourself a break - clear the trash out of your diary, spend time with the teams and customer,
11. Go home earlier, do other stuff outside of work, be the whole you!



Raj Fowler

Director of Enterprise Services

Years at DOG: **1 year 6 months**

Raj Fowler is a leading authority on how organisations re-orientate themselves to compete and win in the digital economy. Raj has a track record of organisational change in a global Aerospace and Defence business with a deep understanding of the challenges faced by large enterprises. Employing the principles of DevOps and Agile, Raj has successfully established products that have enabled competitive advantage through harmonising both speed and stability, harnessing both the power of people and the power of technology.

Raj, as the Director of Enterprise Services at DevOpsGroup, works with enterprise customers to fulfil the need for speed. Speeding up the work is achieved by embedding DevOps practices supported by Cloud Platforms and Continuous Delivery Pipelines, resulting in the transformation of “months” to “minutes”

Contact Us

Cardiff

Floor 22, Capital Tower
Greyfriars Road Cardiff
Wales
CF10 3AG

London

Rise,
41 Luke Street,
Shoreditch
EC2A 4DP

0800 368 7378

Team@devopsgroup.com *for general enquiries*
Academy@devopsgroup.com *for training course enquiries*
Talent@devopsgroup.com *for recruitment enquiries*



Happiest Minds Technologies

www.happiestminds.com



One of Happiest Minds' specialties is hybrid cloud setup. What are some best practices for companies to consider when running a hybrid cloud?

We believe that there are many practices that companies need to adopt based on our experience as an SI and Managed Services Provider in this space. We see that organizations are at various levels of maturity in hybrid cloud adoption and management. Many of the best practices are highlighted below:

Unified IT Monitoring

- Deep Discovery for Application Dependency Mapping
- Intelligent Monitoring of Infra and Apps
- Advanced Analytics for Operational Insights
- Self-healing & Data Driven Automation

Self-Service Provisioning

- Service Catalogs
- Self Service Portals
- Automated Provisioning & De-provisioning of Resources

Cost Management

- Visibility into Costs
- Billing & Metering
- Charge-Back & Show-Back

Enabling Developers

- DevSecOps Integration
- Code and Infrastructure Deployment Automation
- Secure SDLC

Security and Compliance

- Application, API, OS & Network Security
- Configuration & Compliance Governance
- Identity & Access Management
- SIEM Integration

Reporting & Analytics

- Dashboards for CxOs, IT Admins and End Users
- Availability, Performance, Capacity, Costs, SLA and Compliance Reports
- Analytics Driven Insights



When it comes to securing a hybrid cloud, what are some of the big challenges you see your clients facing, and how do you help them to overcome them?

Many of our clients face challenges with respect to regulatory compliance needs (e.g., GDPR), establishing the right level of security controls and compliance for both private, public and hybrid cloud infrastructure and applications, integration of multiple tools (e.g., existing tools for on-premise and cloud based/SaaS tools for cloud infrastructure), gaps in skills/processes/tools when it comes to security monitoring and operations. Happiest Minds takes a consulting led approach to address security challenges. We have a Governance, Risk and Compliance (GRC) practice to understand AS-IS landscape and maturity and provide a road-map for the TO-BE state. To address any gaps in security the customer may have, we offer full spectrum of services such as Cybersecurity Services, Identity and Access Management, Data Security, Advanced Threat Management.

As companies are increasingly adopting “cloud native” model, we see a great opportunity for DevSecOps integration for those on hybrid cloud journey.

What kind of metrics do you recommend your clients use, and what kind of actionable insights will they yield?

From a monitoring perspective, we recommend that application availability, performance, capacity, and cloud consumption cost are monitored with thresholds for triggering alerts. Increasingly, we are recommending the use of monitoring tools with analytics to generate insights to address the dynamic nature of cloud infrastructure and consumption patterns. The insights can be related to application performance, anomaly detection, costs, compliance, and capacity planning. We leverage our ELLIPSE platform for monitoring + analytics when we provide managed services to clients.

How can companies go about evaluating their own incident management procedures? What are some best practices they should be looking to implement?

We take a holistic approach at IT operational setup. Incident Management (IM) is one part of IT operations. Companies can start evaluating IM effectiveness by creating a framework involving process, tools, technologies, and people. It typically starts by ensuring an accurate infrastructure inventory is available followed by a monitoring tool with good coverage of the infrastructure. Next comes the Service Desk tool with IM workflows defined to suit the business and operational needs of the company. With such basics in place, companies can evaluate their need to automate further. Typically, it involves integration between tools for auto-ticketing, alert noise reduction, event correlation, automating tasks etc. We provide an IT operational maturity model to customers that can help define short and long term road-maps towards achieving operational excellence.



In a *white paper* you did on managed infrastructure services, you wrote that freeing internal staff from operational work should be a goal of organizations. Can you give an example of a task you often help your clients to eliminate, and how you go about doing it?

We offer managed infrastructure services which enables customer's teams to focus on more strategic objectives (for example cloud transformation) and reduce the time spent on daily operational tasks such as monitoring, incident remediation, patching, system upgrades, etc.

Where do you see the role of DevOps tools in a hybrid cloud setup? What kinds of functionality should companies be looking for when they decide to adopt a tool solution?

In our experience, we often see DevOps or DevSecOps being adopted in environments that are going cloud native with micro-services architecture being deployed. We see significant barriers to DevOps adoption in legacy or traditional environments where the infrastructure has not been standardized and virtualized to make it easier for automated provisioning/de-provisioning activities. From an "Ops" perspective, most of the customers are looking for tools that help automate deployment, configuration management and monitoring automation.

What are the key performance metrics you recommend companies track, and how should they use them to increase performance?

From an infrastructure perspective, measuring CPU, memory and storage I/O, network latency and database query performance are key to ensure how the underlying infrastructure impacts application performance. For applications themselves, measuring application uptime, response times, number of active sessions and application error rates can help provide insights into user experience levels. Real User Monitoring and Synthetic Monitoring techniques can also be useful in certain use cases.

Your website states that your *ELLIPSE platform* can reduce IT infrastructure operational costs by as much as 30%. What are companies doing wrong that allows you to offer such a big improvement?

The platform alone cannot bring in such improvements in productivity. It is a combination of process streamlining/simplification, monitoring coverage, reducing efforts through self-service/self-heal/self-fix/task automation and actionable insights leading to better decisions.

The ELLIPSE platform also does "policy-based automation". Can you give an example of what that looks like?

A simple example can be automated log rotation – if the size of the log file exceeds X GB, then the automation engine can spawn an action handler to initiate log rotation.



If you had to give companies just one piece of advice about IT infrastructure, what would it be?

The complexity of managing private, public and hybrid infrastructure has increased due to multitude of technologies, tools and skills required over the past few years. To simplify infrastructure management, a holistic view together with application is relevant wherein process, technology and configuration standardization is a must before automation can be introduced.



Girish Chandangoudar:
General Manager
Head of Infrastructure Management Services

Girish Chandangoudar has over 25 years of experience in Telecom and Enterprise IT services spanning Internet Services, Network Operations, System Integration, Managed Services, Infrastructure Design & Architecture and Technical Pre-sales. He is currently a part of Infrastructure Management and Security Services business unit in Happiest Minds Technologies Pvt Ltd. He is responsible for infrastructure practice strategy, solution and offering development, pre-sales and enabling delivery teams. Girish has wide ranging interests in cloud, data center, networking, and emerging technologies.



MBCTG

www.mbctg.com



MBCTG MBCTG specializes in IT modernization. We design, develop and implement state-of-the-art products in the fields of software infrastructure (middleware, data stores, storage, and analytics) and internet-facing IaaS (Infrastructure as Service), SaaS (Software as Service) and PaaS (Platform as Service) applications.

Take me through some of the biggest mistakes you see companies make in how they approach the problem of bringing IT into a modern DevOps approach?

DevOps on a budget

Companies try to stand out against the competition with fewer resources and less-skilled people, which leads to the poor quality delivered. DevOps is good only with adequate resources and experience. Also, DevOps binds Developers, QA, and Security together, so inadequate management is not affordable.

Misaligned focus

Tools are the elements enabling DevOps, taking off some manual work, and eliminating process repetition and effectively managing the available resources. But the real value of the tools is in the goals people achieve using them.

Separate DevOps teams

The creation of an independent DevOps team usually adds processes and complexity. It's also hard to force IT and operation teams for collaboration. Companies should better focus on the DevOps process rather than a new unit because DevOps is just a new approach to developing products faster.

Expecting magic

DevOps is not going to manage itself. It's about continuous learning and SMALL CONTINUOUS improvements. Companies need professionals to manage resources, budgets, goals, and progress, prioritize and break down projects into shorter timelines, and milestones are the key to obtaining results in the initial phase.

Not preparing for the culture change

Change is a part of the DevOps process, and it's inevitable. Most people fail to adapt to change and continue with familiar patterns, especially when change is difficult.



The Cloud is a hot topic right now, but many organizations lack the expertise to know how to approach it. What are some of the questions that organizations should be asking as they start to build a strategy for integrating a solution that works for them?

How does Cloud computing factor into our application strategy and architecture?

How will we access, secure, manage, integrate, and govern across cloud environments?

How should our existing data center, infrastructure approaches, and technologies change?

How should companies go about assessing a baseline for what they need from their cloud infrastructure?

The managers must understand the business role cloud will take in their business. Cloud is no deferent from any other outsourced services such as reception or bookkeeping. Sometimes the Cloud just not the right solution for the company. The baseline would be to define the business process or a role that business needs. Then to figure out how to implement it. For example, a business looking to apply test-driven development will have a deferent baseline than a company looking for application performance or cloud accounting and billing solutions.

You talk about the idea of a “Cloud transformation.” What is the most successful approach you’ve seen organizations take to ensure the move to the Cloud provides all the functionality and benefits anticipated while minimizing disruption to business operations during the transition?

Identify a specific business reason for moving to the Cloud – understand the motivation, business outcomes, business changes, and project scope.

Develop a business case for moving to the Cloud.

Architect, plan and create a baseline cloud account structure (how to manage cloud accounts) and control environments (backup, networking, IAM, and DR) that support the core business, security, and compliance needs.

1. Discuss your background and goals. Review the data sources and current workloads.
2. Review current monitoring. Uncover trends, outlying behaviors, and overall performance.

What are some of the problems companies run into when they try to set up a hybrid cloud, and how can they solve them?

Companies must solve many issues to make their hybrid IT strategies work. Legacy systems, multiple workloads, finding and retraining the right people, security, and compliance issues are among the



top challenges that IT must mitigate. Today, most of the businesses ending up with hybrid Cloud by accident. Companies are integrating public Cloud with on-premises tech without standardizing. We see more shadow IT cloud “experiments” that suddenly become production, and outdated governance practices that puts a business at high risk. To design a good hybrid strategy, companies should look into standardizing infrastructure as code practice. The effort toward being a hybrid is much cartulary as technological. Companies must adapt both local support and CI/CD automation. Skill the workforce around developing infrastructure as code. And build functional teams focused on product delivery.

Let’s talk scaling. How can companies effectively scale their public, private, or hybrid cloud infrastructure while making sure that they find the optimal balance between cost and revenue?

Vertical scaling or scaling up is done by resizing your server with no changes in your code. Horizontal scaling or scaling out is adding additional nodes to the existing infrastructure. You can combine both implementing essential scaling strategies: timing and threshold. Timing is a simple strategy that scales up or down vertically and horizontally depending on the scheduled time, such as removing an instance at 19:00, which is efficient for periodic workloads. A threshold is the most popular strategy when scaling decided by the given threshold: memory, CPU utilization, bandwidth, and other application metrics. Of course, you can also use a mixed strategy.

One of the primary reasons for scaling is to increase performance. Scalable architecture can handle bursts of traffic and heavy workloads. However, sometimes, in case of moving from a monolith to microservices, scaling may increase response time, which is also one of the primary metrics of performance. Scalability efficiency usually depends on resource granularity and different types and patterns of peak workload.

Companies sometimes know they should be scaling but fear the amount of time that takes. What’s the best way for organizations to scale quickly?

Scalability is the new norm. Auto-scaling enables you to build consistent infrastructure and application-driven scaling to control cost. It involves combining monitoring and integrating action plans. Most cloud vendors provide (because it’s in their interest amplifying resource consumption) built-in tools and make it relatively easy to achieve scaling. A more effective approach would be using cloud brokerage service platforms like Abiquo. Abiquo product will allow companies to quickly and efficiently implement multi-cloud scaling and stay in control of the project budget. Multi-cloud scaling or cloud bursting is a better approach and will benefit your company of staying out of the vendor lock trap.

Correctly configured Multi-cloud scaling can sense and inform when scaling is required, be prescriptive about what to scale, and orchestrate all the underlying processes through automation. And again, look out for vendor lockin.



Besides scale, infrastructure as a Service (IaaS) offers a lot of promise and much peril in terms of security. What vulnerabilities do you see companies overlooking? What process can they follow to minimize the number of oversights?

In general, IaaS is vulnerable to all of the threats that we know from on-premise environments. Like SQL injections, cross-site scripting, and other well-known attacks listed in the Open Web Applications Security Project (OWASP). So applications need to be developed with secure coding practices in mind.

According to Gartner, the most common IaaS cloud security risks are still misconfiguration and so-called “shadow IT” workloads like deployed and not updated test servers, orphan storage, and network resources. Most of these issues addressed by building correct IT procedures, the use of automation, and monitoring mechanisms.

Can you share a creative custom management solution you’ve seen that other organizations might be able to take inspiration?

Many customers we worked with have been looking to bring a new product or service to the market. To achieve that goal, you are going to need a solution that has a simple architecture and implemented without impacting your existing investments in infrastructure. You will also need a solution that uses familiar technology stacks, so you don’t need to spend time training your technical team and has the flexibility to integrate to your existing technology and processes. Overcomplicating any one of these will only add to the time it will take to launch your new service, and ultimately cost you dearly. Choosing a solution like Abiquo as your Cloud Management platform is going to enable you to meet your goals quickly. Abiquo built on a familiar technology stack with a simple modular architecture that will allow your business to scale when you need to. We have a track record of implementing quickly and can work with you to make your project a success

If you could only share one IT cloud infrastructure best practice, what would it be?

It will be a complete waste of time building a new service if you are unable to sell it. But all too often little thought has gone into answering this question until the solution has been built and it is sat there with no customers using it. It is essential to engage with your existing customers as early in this process as possible. After all, you won’t be able to answer some of the earlier questions if you have no understanding of what your customers need. It is equally important to engage with your sales team. They are the ones regularly in front of the customer and will need to understand what they need to sell. It is also essential to make sure that you build something that they can sell.



Maxim Borovkov

CEO

Entrepreneurial executive and investor with over ten years of experience in managing cloud-based solutions and IT security companies. MBCTG established in the year 2009 as a middleware consulting and professional services firm. Today we provide excellent professional services focused on application performance improvement, architecture, and stability. MBCTG adds development and integration services and resells innovative software and hardware products.

Our mission is to accelerate the success of our valuable customers throughout cloud transformation and modernization processes.

Our team consists of Talented, Experienced, and highly skilled individuals with Honesty, Integrity, Professionalism, and proven capabilities in the execution of advanced technologies projects.

Contact Us



QBurst
www.qburst.com



A global product development and consulting company with a strong focus on new-generation technology platforms. We provide a broad range of services around Analytics, AI/ML, Web and Mobile Development, DevOps, User Experience Design, Blockchain, IoT, and Testing.

Your website states part of your core mission is to “help clients maximize the effectiveness of their business”. What are some of the common ways you see a lack of DevOps adoption harming companies? Are there any recurring drains on productivity that companies may not even be aware of?

Some common issues that we have encountered are:

- Slow deployment cycles, often holding off multiple, small changes by weeks for a window.
- Issues take longer to get fixed.
- Lot of uncertainty and duress prior to release of new features.
- The implementation/deployment phase is often ignored while estimating to get a new solution in place.
- Minor configuration changes are often missed and don't get applied on servers.
- Operations teams often roll back changes during an incident, without understanding if that change actually caused the issue.

In non-DevOps environments, teams spend a lot of time solving problems over and over again. This not only leads to dissatisfaction, but also creates a significant drain on productivity.

How would you suggest companies go about assessing their own DevOps needs?

A good way to evaluate DevOps needs is to have a bottom-up approach where teams assess themselves and find areas to improve based on company goals. Consider whether teams are unified, extensive, or work in silos. Is there a climate of transparency?

- Assess current state of Dev and Ops
- Check whether teams can collaborate and equip themselves with the technologies and practices of each other's processes
- Gap analysis - Identify opportunities for automation
- Identify challenges in build, deployment, and release management



- Assess your organization's ability to deliver and maintain stable applications to serve business requirements as quickly as they are needed
- Examine whether formal IT Service Management processes are in place and whether these are mature enough

One of the big barriers to DevOps adoption is fear stemming from the large number of divergent platforms used by Development and Operations teams. What are some of the questions companies should be answering internally when creating a plan to ensure fast, seamless handoffs between teams using divergent tools?

- What is the skill and knowledge level of the current team?
- Is the team open to learn and adapt to new technology?
- How much training is required to ensure seamless cross-functional collaboration?
- What is the feasibility of using a common set of tools across the organization?
- How can individual handoffs be reduced?
- What safe-to-fail experiments can be conducted to identify areas that need attention?

Where should companies be starting with automating their iterative process? How do they ensure their new automation will not break their current manual processes so much that productivity suffers in the short-term?

Ideally any repetitive tasks should be automated. And as such, all installations, configuration changes, and deployments should be automated and where possible, provisioning of servers should be automated.

Start by automating tasks that don't take up a lot of time but are repetitive. It can be a simple task such as installing a web server on a machine or deploying a war file to a server.

In order to ensure the new automation does not break existing changes, start off by testing it in lower environments such as Dev/QA/Stage. Initially there will be a lot of rewriting involved, but as the teams adapt and start creating automation scripts, a lot of time can be saved with the automation of repetitive tasks.

On the theme of cross-team organization, talk about some of the version control problems you see companies with manual processes facing. Can you give an example of how DevOps can streamline change requests and disaster recovery?

One of the key problems with some of the version control systems is the inefficiency in serving geographically dispersed teams. Scalability of VC systems is also a pain point that is impacting DevOps success.



Versioning tools and (more importantly) a good Version Control Strategy, gives many advantages, but the major ones to this scenario are:

- A simple way of viewing what has changed and when
- A single source of truth

With a good branching strategy, each pull request, provides a simple and easy way to understand what is going into the merge. By attaching a PR to each change request, it becomes simple to understand what is being released with the change.

An example of how DevOps can streamline change requests and disaster recovery: For a simple DR plan such as backup and restore, using a VC tool for Infrastructure as Code offers a single source of truth that defines what each instance should look like. The same can be used to carry out a DR, provided relevant data is backed up, so that it can be restored. Furthermore, the use of Configuration Management will further help by ensuring required packages and dependencies are implemented across the board.

Let's talk Containers. What are some of the common issues you see companies facing with technologies like Docker, and how can they be overcome?

- a. Hesitation to adopt something new and not properly understanding how the container works or using the wrong values where it is not needed. For example, using ARGs instead of ENV, or using links where they are being deprecated. All of these can be easily addressed through training programs.
- b. Not thinking about container orchestration or how containers should be deployed in production and deployment policy for the same. Moving towards a defined CD pipeline helps, and so does defining an application lifecycle.

Companies making the move to microservices usually need to keep some of their legacy monolith untouched. What are some best practices you can offer for how to use DevOps to get the most out of a monolithic/micro service application hybrid?

The recommended solution is to have a dual speed for DevOps in hybrid setups, where the practice is to take things slow with monoliths, and move fast with microservices — this although is not always possible. It is often recommended to implement as much DevOps practices across all environments uniformly. Yes, this can be avoided to keep the environment stable, but it is only going to slow down the pace as a whole.

Where possible, it is also suggested to start looking at migrating your legacy applications to your new environment, or at least to start rearchitecting your code. Which solution would suit the client will depend on the client's needs.



Server security is a big issue right now. What are some of the strategies you suggest companies adopt to ensure applications remain stable for end users?

When we talk about security in the DevOps world, we're usually referring to either Application Security (and the entire shift left practice) or Server Security. For all our clients, we suggest a standard policy when it comes to securing servers. For clients who have no preference, we suggest CIS (<https://www.cisecurity.org/cis-benchmarks/>) as a bare minimum. We usually do this by setting up playbooks or roles within the configuration management tool to set up all the standards in the environment. We have this playbook/role executed as the first (and compulsory) step against all servers. Furthermore, we suggest running the same at routine intervals to:

- a. Detect configuration drifts and resolve them.
- b. Add new standards into policies when they are released

Server Configuration Drift is very often considered an inevitable reality of complex enterprise organizations. What best practices can you offer companies for how to maintain their application performance?

The ideal scenario is to use an automated Configuration management tool, for example, Puppet, which has an agent that runs always and ensures there is zero configuration drift. However, that is not always possible. We usually suggest one of two things with regard to Configuration Management (CM):

- a. Conduct regular audits on all devices to capture and fix configuration drifts
- b. Set up an automated monitor to routinely run the same, based off an agent (similar to how Puppet does it), and automatically correct drifts.

From a process perspective, we suggest that all changes be committed to the repository that hosts your configuration management tool. So even if you are making a critical patch change in production, you must have that change added to your CM. If you cannot commit the changes beforehand, due to the criticality of the issue, it must be at least added as soon as possible afterwards.

If you could offer one best practice to companies who currently use some DevOps principles but would like to increase adoption, what would it be?

Adopt Iteratively: Identify one pilot application, form a cross-functional DevOps team that includes developers, testers, and operations. Examine your possible constraints and bottlenecks and create a deployment pipeline that addresses these process constraints. Measure progress, success, and repeat.

Moving to DevOps involves changes in team structure, attitude, and tools. The focus should be on enabling organizations to adapt culturally and automate key processes in development, testing, integration, delivery, and monitoring.



Please summarize the mission and methods of QBurst. Is there a client story that represents the value you provide your DevOps clients?

Our client is a pioneer in cloud-based analytics. Their product enables companies to easily integrate Internet-enabled devices with business systems. Through a series of processes, data is organized into consumable formats that downstream systems can use. QBurst worked with the client in the role of DevOps—enabling them to control changes to product master repositories, deploy to a wider range of environments and gain broader access to help operations diagnose production issues. You can check out the case study [here](#).



Praven John is a DevOps Architect at QBurst with 11 years of experience as a DevOps, Cloud Engineer, and Systems Admin. A huge advocate of Open Source systems, Praven spends his free time with family, books, and keenly pursues his interest in photography.

Contact Us

United States

14150 Newbrook Drive
Suite 115, Chantilly, VA 20151
+1-571-281-2720
info@qburst.com
www.qburst.com

Fast Facts

- 120+ Active Clients worldwide who rely on our consulting and business solutions across industries
- 1400+ projects rolled out successfully
- Key Alliances & Partnerships: Microsoft Gold Partner, Salesforce Registered Consulting Partner, Oracle Gold Partner, Adobe Enterprise Solutions Partner, Amazon Web Services, Salesforce Registered Partner, Drupal Organization Member
- Major Clients: NYU, Dell, Bajaj Allianz, Bozutto, Omron, AddThis, Petrofac, Adani





ReleaseTEAM

www.releaseteam.com



ReleaseTEAM is a veteran-owned, full-service DevOps consulting firm, servicing both the public and commercial sectors. We focus on helping our clients adopt and mature their software development lifecycle methodology, with solutions tailored to their objectives and culture. We offer the best solutions in the industry, delivered with over 20 years of expertise. We offer DevOps consulting, support, staff augmentation, mentoring, and DevOps software solutions. Our staff will work with you to understand your unique needs, audit your current development process, and recommend adoption paths with the highest return. Our consultants then evaluate the tools and own the implementation.

You wrote a blog post about the importance of DevOps tools. What advice can you give companies about how to effectively evaluate their needs and select the proper tool(s) to meet them?

Tool selection is essential, but as our blog has often pointed out, not more so than people and processes. The right set of tools is unique to every team or organization; they should fit your processes and people. Many organizations make the mistake of trying to change their people and processes to fit the tools. Consider the whole picture, do your research, and the right choices should emerge. Of course, engaging an experienced consultant to help you assess your environment and select the best tools is never a bad idea.

You use the potential for “integration hell” as one reason to ensure quality Continuous Integration processes. What objective benchmarks can companies use to identify when they are in integration hell? What’s the first step they take to try and get out of it?

It may be obvious to many when they are having integration issues. Still, many organizations get so used to allowing time in a schedule for integrations that they become numb to it -- they are already in integration hell. The objective benchmark is that integrations are taking time during the product schedule. The first best step is to implement CI.



You wrote an entire white paper on how to implement DevOps in a change-resistant environment. What would you say is the biggest takeaway?

Change makes most people uncomfortable to some degree, especially when it comes to doing their jobs. To ease the process for your team, keep them informed and feeling that they are a part of the process and don't rush to implement all the changes at once. Too much change all at once can be overwhelming and can make some folks feel they are being left behind. Left out and left behind – nobody likes those feelings.

You offer organizations the opportunity to “train your [DevOps] trainers”. What do you think the number one mistake managers charged with instituting DevOps principles into their team make?

Training for a new process or tool should start before the new tech is in-production. Making sure that everyone is on the same page and knows what they need to do to excel at their jobs is critical. Having team members work side-by-side with a seasoned pro during the implementation of your new environment, adds an extra layer of knowledge that can then be shared with the rest of your group.

You also offer customized DevOps training for teams. That underscores an emphasis on culture as much as technical prowess. How should cultural considerations factor into the strategic decisions made by management when implementing DevOps.

When an organization ignores the culture of their environment, new concepts don't tend to stick. For example, you can teach the Zero Blame policy to everyone in your organization, but the concept will fail to gain traction if the teams or individuals continue to pin the root causes for errors on an individual or another group.

What are some of the biggest DevOps security challenges you see companies facing today, and where should they turn to start tackling them?

Security has always been a concern for software development; DevOps allows teams to address issues quicker and earlier in the process. There are many great security tools that organizations should explore to address their specific security needs; however, security starts with people and best practices. Adequate training and access to security policies and procedures are critical.

On the IT side, how should companies be balancing the benefits of Infrastructure as Code for scalability with the increasing potential of configuration drift that comes with scale?

IaC is a fantastic technology, and organizations utilizing DevOps should also embrace it. However, IaC falls in with many other great technologies that can, along with DevOps principles, enhance repeatability, time



to market, and overall productivity. It's like peanut butter and chocolate – they both stand on their own but are also great together.

You mention the need for working rollback plans as a part of release management. What kinds of things should companies be doing to make sure they can quickly rollback when needed?

Disaster recovery has been a critical part of systems since their inception. Having access to previous (and hopefully stable) versions of software and files ensures you can, well...recover from a disaster. Of course, DevOps gives us automation. Having automated rollback plans in place is a great start. Add in a solid strategy for managing rollouts – such as blue and green rollout paths, and canary deployments, to eliminate downtime and minimize the impact of issues, so they don't become disasters.

You also wrote a blog post on the DevOps security challenges the government sector faces. What are some best practices for government entities to balance lead time speed with security needs?

Leverage automation everywhere and bring security into the product pipelines. Including the entire operational scope within a pipeline, security becomes an integral part. Using this approach, government agencies can utilize DevOps to address security concerns at the code build and release cycle.

If you could give organizations a single piece of DevOps advice that will have an immediate impact on their organization, what would it be?

If we are limited to a single piece, then understand your value. Knowledge is Power. Assess, learn, teach, and document.



Shawn D. Doyle
Co-Founder, CEO

Shawn D. Doyle is a founder and CEO of ReleaseTEAM, Inc., a DevOps Consulting Firm, established in 1999. After serving in the US Army and Desert Storm, he has amassed nearly 30 years of software delivery experience, having worked with companies from mid-sized to Fortune 100 to solve their business-critical challenges. Shawn lives with his wife and three dogs in Colorado, where he is working on his first book, DevOps Overture: What you need to know when starting a DevOps Journey.

Contact Us

Corporate HQ

1400 W 122nd Ave.
Suite 202
Denver, CO 80234

(866) 887-0489
info@releaseteam.com

Massachusetts

1257 Worcester Rd.
Suite 108
Framingham, MA 01701

Canada

PMB# 604
1-110 Cumberland St.
Toronto, ON M5R 3V5





RocketMakers

www.rocketmakers.com



RocketMakers accomplishes a lot as a small team. What advice can you give to small DevOps teams on how to punch above their weight and make the biggest impact possible with just a few people?

Replace documentation with scripts (which are self-describing as much as possible).

Modularise components as much as possible - keep each part of the workflow as simple and repeatable as possible. Focus on a few good tools and keep it consistent.

Your website mentions the idea of “fail fast agile development”. What is the biggest misconception companies have about what that means, and how to achieve it?

Fail fast is about the business more than the tech. For us, it's how we can prove that customers need the product to solve a genuine problem as quickly as possible - often called product/market fit these days. Having built lots of startups, the hardest part is shutting them down when they run out of cash/time - so we like to fail quickly (if they are going to fail) so we can get on with applying the learnings from that failure to the next opportunity.

Companies sometime feel they need to compromise between security and speed. What is the true nature of the relationship between these two goals? How can management minimize risk exposure without increasing lead times?

We typically see security and user experience being more at odds more than security vs. speed. Usually security gets in the way of a smooth user journey and we have to compromise somewhere.

Having said that, some key things for us are:

- Having repeatable code/scripts which have been tested and are re-used can help
- Doing things well and fast aren't always mutually exclusive - components and micro-services that have been tried, tested and can be re-used are the way we accelerate start-ups and keep costs low.
- Education is key - making sure developers and DevOps understand the likely security vulnerabilities and how they can be mitigated
- Consistency in architecture is essential - we insist on having tested scripts in place which bring up the whole environment in a fully tested way. That way you get repeatable and predictable environments that save time in the long run and also help with disaster recovery too.



Keeping on the security theme, DevSecOps is another buzzword we hear a lot now. What's a concrete way for companies to think about the concepts packaged into that term, and how can they work them into their DevOps strategies?

Getting the basics right, such as making there are no secrets in source, ensuring keys are managed effectively outside of source code control in vaults, etc.

Consciously minimize the exposure/surface area of the application and the tools you use to build and deploy

Remove direct access - Restricting access to those who really need it, locking down VMs and containers to controlled connections and consider using a Bastion approach for connecting.

Run regular penetration tests ideally via a 3rd party and act on their recommendations

Always use the principle of least privilege with all apps, containers, machines running with the very minimum of permissions by default and loosening them only for specific documented needs

You work with startups. Do these smaller organizations need DevOps? What does implementation look like versus what is needed in a larger, more mature corporate structure?

Definitely! Unfortunately, they often don't see the value (when they often want every dollar spent of features), but they do need it. The good news is that we do it for them and educate them about the real return on investment in the long term.

Without fail, they all see the value later - when we can create an environment at the touch of a button to demonstrate a new feature and when finding issues with performance, availability, scalability, etc. For us, they need the same as a more mature organization if they are going to scale, so we address it from sprint zero.

As companies grow, maintaining stability while scaling is a huge challenge. How can DevOps help this process, and what are some best practices for scaling DevOps operations and culture itself?

Maybe we've been lucky but building in a core DevOps understanding and capability from day one means this is not an issue in our experience.

Having repeatable, consistent processes are essential and this is where working with a development shop that has been through it all before can make all the difference - it took us at least 18 months to get our initial processes humming and a decade of tweaking them ever since.



While many people are looking for documentation, our view is that self-documenting avoids misunderstandings, old docs, etc. Not suggesting that having documentation is bad in any way, just that out of date documentation can be worse than nothing at all.

We also think that a common toolset = common language = common understanding and efficiencies.

If you had to pick the one pitfall you most frequently see scaling organizations have with DevOps, what would it be? What can companies do to avoid the same problem?

Not setting everything up consistently at the start with a view to it scaling from day one would be the key thing from our experience. To help avoid it make sure you schedule in laying the DevOps foundations in sprint zero and leverage existing tool sets rather than writing your own.

You work with all of the most popular DevOps integrations, but of course every organization has its own needs. How should companies evaluate DevOps tools so they can hit on the stack that best meets their needs?

That's very individual, so set your own goals/outputs and focus on ensuring they are met rather than worrying about the detail. In our case we care about making developers and operations teams life easy, vendor agnostic, self-describing, consistency using proven tools and tech, and using a buy rather than build approach where possible but with our own code to orchestrate as needed.

If you could give one best practice to companies looking to either scale or overhaul their DevOps operations, what would it be?

Come to RocketMasters :-)

More seriously, focus on containerization through Docker or similar and management via Kubernetes, etc. and ensure you have environments that can be created and run locally and remotely in a consistent way.

If you could give one practice to companies getting ready to implement DevOps for the first time, what would it be?

Learn Docker and Kubernetes fast!



ScienceSoft

www.scnsoft.com

scnsoft.com/services/devops-consulting



One of the pain-points you try to alleviate is a “lack of efficient horizontal collaboration”. What are some of the collaboration mistakes you see companies making over and over again? What best practices can you offer them for alleviating them?

The one collaboration mistake that, in our opinion, can have the largest and most negative impact on the work processes is unclear role and responsibility distribution. When even one person on the team doesn't know what they should do and what their coworkers are doing, the entire team is at a risk of facing unpleasant consequences: same tasks can turn out to be performed by several people, while others aren't performed by anyone at all. Making sure that everyone is on the same page in terms of their own and collective work scope is paramount.

You talk about helping clients to develop a “new technology agenda”. What do you mean by that?

Instead of simply listing the possible technological improvements, we create a detailed and balanced ‘agenda’ – a plan that helps a company to clearly see the gradual steps that should be taken in order to improve their current IT strategy. We also employ change management in order to make the transformation as smooth for the customer's businesses as possible. As a result, our approach helps the customer to conveniently allocate their investments and reduces any possible financial losses related to the disruption of business process during the strategy change.

Your website states IT departments should be thinking about what makes a business “customer-focused”. What are the big strategic goals companies should be looking to accomplish in order to achieve a customer-focused IT approach?

We believe that business-customer interaction should be organized in a way that would minimize the efforts on the customer's side. IT businesses should strive to facilitate communication by catering to the customer's industry background and depth of their understanding of IT specifics. Ability to explain everything in layman's terms is one of our core principles that makes our services customer-focused.

You list “reduced IT spending” as a goal of your IT consulting. How can companies achieve that, while still maintaining/increasing the quality of their services to end-users?

Before launching any cooperation, we discuss the consulting scope with our customers in order to set clear goals and define expectations from our services. In our practice, this not only leads to customer's



satisfaction with the quality of our services, but also prevents their budget exceedance.

You work with hybrid cloud setups. What are some of the key mistakes you see companies making over and over again when setting these hybrid systems up?

Common mistakes are related to the choice and combination of on-site and public clouds. Customers can focus too much on the costs and go for a public cloud with lower performance speed while having a very fast on-site cloud. This difference often results in overall staggering performance.

When it comes to data and metrics, your website states you help companies “develop rules and policies to ensure high data quality”. What are some key considerations for companies if they want to do their own internal audit of their data policies?

During the audit, we ensure that corporate data policies regulate at least two most important aspects: data quality and data security. Data quality management policies should cover attributes to be tracked (i.e., consistency, completeness, accuracy), rules that allow checking whether a data entry corresponds to a certain attribute (i.e., ‘Customer full name must include at least one space’) and thresholds (i.e., 85% of all the data records should comply). With the right combination of data quality management tools, it’s possible to automate most of the data quality assessment process and continuously monitor the status quo per attribute. As for data security, companies should perform monthly vulnerability assessments and penetration testing at least once a year to introduce required changes and reconsider the existing policies.



Andy Lipnitski

IT Director at *ScienceSoft* with 8 years of experience in managed IT services. Andy has extensive background in solution architecture design and

deployment. He directs the effective delivery of IT infrastructure consulting and implementation, *managed infrastructure services* and support services, cloud services (AWS, Azure, etc.) and migration to the cloud, *DevOps consulting* and CI/CD pipeline design and implementation.



Learn More

Inedo Tools Facilitate DevOps

-  BuildMaster is a first-class CI/CD tool that automates builds, testing, and releases, and ensures compliance with gates and permissions, all with a unified dashboard for end-to-end visibility.
-  ProGet keeps your repository secure and deliveries reliable by self-managing packages and managing loads for maximum uptime.
-  Otter helps you avoid the pain of unanticipated, hard-to-fix bugs caused by undocumented server changes by monitoring configuration drift and reporting on necessary remediation.

Want to Learn More about Inedo's Products?

CONNECT WITH US FOR MORE DETAILS:

Mike Goulis
Chief Operations Officer
mgoulis@inedo.com

 56 Front Street., Berea, OH, USA 44017

 <https://www.inedo.com/>

 (440) 243-6737

 @inedo

 <https://www.linkedin.com/company/inedo>