

Chapter 1 : 10 ways to build highly effective DevOps teams | TechBeacon

Some companies think that adopting devops means bringing in specialists or a host of new tools. With this practical guide, you'll learn why devops is a professional and cultural movement that calls for change from inside your organization. Authors Ryn Daniels and Jennifer Davis provide several.

Oct 06, Terry rated it really liked it Decent book. Think some folks were put off by many of the "soft skill" items mentioned at the beginning of the book instead of the presumptive battery of tools to "Do Devops" they may have expected. The valley is the valley for a reason so I feel like the boo Decent book. The valley is the valley for a reason so I feel like the book has done a great job in broaching topics that those in managerial roles may not have brought to the front of their minds. Now, you might argue, "but wait, they do define DevOps, right in Part 1! It is a way of thinking and a way of working that enables individuals and organizations to develop and maintain sustainable work practices. It is a cultural framework for sharing stories and developing empathy, enabling people and teams to practice their crafts in effective and lasting ways. If I remove the word "Devops" from that paragraph above, it could be about anything. They even acknowledge this fact in the book itself: Their definition is too broad, and as a result, the message of this book is very diluted. Some parts are targeted at developers; some at operations people; some for managers; some for HR; some for the CEO; some for the legal team. For example, one is the odd stereotype that a "10x engineer" is always a "10x asshole" that no one wants to work with. Another one is the assumption that the differences between tools e. All that matters is how you use the tools. But the opposite is not true: All of this is a shame, as there is some excellent content lurking in these pages. There are great discussions of diversity and minorities in tech organizations. The authors could extract those discussions, go into more detail, and turn them into an great standalone book focused on that one topic. The case studies scattered throughout this book are intriguing too, as they contain real-world stories, with concrete details! Again, the authors could extract those stories, go into more detail, and create yet another interesting, standalone book on "DevOps stories. But as it is, all of this intriguing content is crammed into a single book, organized poorly, and only explored at a surface level. As always, I jot down interesting quotes as I read. Here are some of the best ones from this book: It is a fundamental shift of perspective in the design, construction, and operation of software in a world where almost every successful organization recognizes that software is not something you simply build and launch—it is something you operate. They will likely go through this time thinking they are doing well, leading to a nasty surprise come review time. The psychology of getting feedback shows that people generally react to these sorts of negative surprises emotionally rather than intellectually, a phenomenon known as amygdala hijacking. As a result, people are less likely to fully understand and be able to act on the feedback they are being given. For common problems use Opensource. For rare problems use Buy. For unique problems use Build. The stories helped to explain presented ideas. I would maybe enjoy it even more if it was shorter: Jul 11, Mark Honomichl rated it did not like it It took me a while to get through this book. Rather than laying out some good information on how to effectively build out a Devops organization, it seemed to be more like the annual workplace sensitivity training that one takes at most large organizations. Mar 04, Sebastiaan Bekker rated it it was amazing Good and broad primer for anyone wanting to start. Wide array of topics that are important for a successful implementation. Sep 24, Tomasz Gawlik rated it it was ok Seems like the books has a few points and ideas which are being covered over and over again. May 14, Brian Palmer rated it liked it The subtitle "Building a culture of collaboration, affinity, and tooling at scale" is important here, because the book spends a lot more time there than on what I was expecting. Automation is briefly discussed, and even scaling is almost all about scaling hiring practices and team growth rather than technical scaling. The book ends up wandering quite a bit The subtitle "Building a culture of collaboration, affinity, and tooling at scale" is important here, because the book spends a lot more time there than on what I was expecting. The book ends up wandering quite a bit through what the authors believe are best practices in collaboration and affinity very little discussion of tooling even at a high level. Moreover, they include some different types of companies "tech" companies to government to retailers. One brief passage in the book, however, felt completely out of

place, and knocked me into a far more critical read of the entire rest of the book. In a discussion about "group membership," the authors discuss "real names" in social networks spoiler: A lot of the book is spent on fostering diversity; I felt again in places that they stacked the deck for example, on remote work, , and floating above issues where affinity groups conflict with inclusivity.

Chapter 2 : Effective DevOps - Optimizing for Efficient Development - Agile®

Devops is a way of thinking and working that enables individuals and organizations to develop and maintain sustainable work practices and a cultural framework for sharing stories and developing empathy, enabling people and teams to practice their crafts in effective and lasting ways.

Determine future business goals with customer and user participation. Determine resources needed to complete goals. Build Develop and test on a continuous basis and collaborate with users as the project progresses. Deploy Release to customers and users value added work soon after it is completed and ask for feedback. Operate Continuously monitor completed goals while looking to incorporate improvements using customer feedback. The plan should be to use Agile development and DevOp principles, but only doing the ones needed to achieve the goals. The resources necessary to meet the objectives then have to be determined along with the metrics that will be used to monitor the progress of the development teams. To ensure continuous improvement and clear up bottlenecks it is important the metrics be available to the leadership team so they can intervene and provide assistance to teams having problems. It is able to do this by incorporating both Agile and lean development principles of small teams, small batch sizes, and smaller but much more frequent software releases. DevOps also builds on infrastructure as code along with continuous integration and continuous deployment. Note, however, that Agile is not a prerequisite to have before DevOps can be adopted. DevOps relies on an incremental approach of doing automatic testing frequently in order to catch and correct defects early on. Automated Testing Since coding is set up to be released in small increments, test automation must be implemented to deal with up to thousands of tests on a daily basis. With automated testing, you want the ability to localize errors and identify where they are coming from. Via automated testing, DevOps also integrates working code on a constant basis across multiple teams to check that the code received from the various teams all integrates together without causing error conditions. This allows for continuous user feedback while the project is still in a process of being completed and minimizes the time, effort, and cost of incorporating changes. Another key element of Agile development and DevOps is that the entire software life cycle of plan, design, develop, test, document, and release is repeatedly done in a timely way over a series of sprints. With DevOps, code written for a new functionality is already in production. This allows a new feature to be run and tested by company personnel prior to its release to the customer. When it is decided to go live with the new procedure, it can then be done on a piecemeal basis to one or a small group of customers. Based on customer feedback it can then be released full blown to all customers; or if there is a problem, rolled back automatically. The company can also perform experiments in production to test new business features. With DevOps, the development, test, and operations departments of IT all work together as one team to help each other. The benefits that will result if DevOps methods are used will be faster implementation time for a project with higher quality and the ability to absorb changes as they occur. Take your organization to another level Contact us to schedule a consultation with our Agile bred DevOps professionals. Without the right level of communications, and more importantly, cooperation between your IT development and operations departments, the underlying structure that supports your corporate applications will either: Fail to keep pace making your organization sustainable to outages. Will likely be over-built thereby costing your company more than is necessary. A typical scenario is when the load from development has outpaced the supporting hardware. As more and more applications developed by your business are added to your current system, your user data grows, and your systems become more and more fragile. Due to either a lack of system upgrades or the wrong upgrades having been implemented. Both situations usually occur because of ineffective communications of requirements between IT development and operations. The result is that the primary duty of IT becomes firefighting just to keep your existing systems up and running; however poorly. If continuous firefighting is the modus operandi for your company, you need to consider moving to communications integration optimized for DevOps can revolutionize the focus of your business. Correctly applied, you can move your focus from outages and blame to strategy and innovation. Ensure that in spite of growing systems complexity; your costs will be kept in check by making sure that

DOWNLOAD PDF EFFECTIVE DEVOPS

hardware is not oversized, or undersized for the current and upcoming needs of the system. If you choose to work with Agilest, you will discover that internal software development you can get right, and right-sizing systems to support an evolving application structure does not need to be ambiguous and stressful. Become a Mastermind Group Insider Get the latest industry updates, invitation to online events, subscription to the Agilest IT Leader magazine, training promotions and much more

Chapter 3 : Effective DevOps - PDF eBook Free Download

Effective DevOps Book Description: Some companies think that adopting DevOps means bringing in specialists or a host of new tools. With this practical guide, you'll learn why devops is a professional and cultural movement that calls for change from inside your organization.

Help teams play Sec. A good solution has good communication and marketing. Nothing happens if people do not know how to play Sec; they must know what is right and what is wrong when they develop software or provision and configure infrastructure. Everyone knows how to play Sec and they are prepared to implement the policies and follow the rules. If no human task is able to answer the demand, then we must implement enforcement in our pipeline, either for software and infrastructure. I will not show a proprietary tool, but you will find paid alternatives in sites like stackshare. No Secret Exposures Secrets are passwords, passphrases, private keys, db account, or tokens. To have no secret exposures, first, we must not have secrets saved within flat files, hardcoded, or within the Git repository, and no perpetual passwords. To solve almost all of these problems, we must employ a vault tool to manage every single secret. A vault tool is obligatory to manage the secret life cycle, mitigating the vulnerability in case of a leak. A mature tool can manage many different types of credentials, like databases, operational systems, cloud providers, etc. HashiCorp Vault to manage secrets Gitrob to scan Git repositories for exposures If you use secrets in your pipeline, SSH key, or consume it directly from your automation runner e. Employ a tool to monitor each certificate, inside private domain and outside public domain of our company: Sensu to check certificate TTL HashiCorp Vault to manage the certificates of our private domain Sensu does not have this functionality out-of-the-box, but there is a good plugin. With Sensu, we can do a lot of things related to monitoring, and to help other targets, check this out. No Outdated Libraries A big company has thousands of projects in development running in parallel. This is prone to using outdated libraries when a team starts to develop a new microservice, for example. Never apply enforcement in legacy just to follow the crowd. Work with the teams and understand the impact of the new rules and avoid the chaos. The tools available to perform this job are specific to the platform and language that we have in our projects, but these tools are a must-have to help us to identify outdated and bugged library dependencies: We must create a step in our pipeline to enforce this target in every build. This guarantees that we never pass an unauthorized, outdated, or bugged dependency. To enforce this target in the server farm, we can employ tools like Chef and Inspect to manage the configuration and maintain stability. These tools act at the OS level to maintain things right; normally they require an installed agent in the host. Static Application Security Testing. This target hunts bugs, bad practices, potential memory leaks, infinite loops, and anything that can cause a vulnerability. A well-known tool that handles this job is SonarQube , which provides several ways to scan the source files, hunting bugs and bad practices, and is used for many languages. Many tools follow this guide and apply their own checks; for now, I will propose two good options. Measurement Know the numbers to understand what happens. The measurements are critical to understanding how much is effective with abDevSecOps initiative. Without them, we are in a darkened room. There are good open-source options for metrics and for presenting them. Grafana has a good integration with InfluxDB, too. What can we measure? This is my view of effective DevSecOps. If you like it, give some claps and share your opinion! See you in the next articles. Read More From DZone.

Chapter 4 : effective devops – Building a Culture of Collaboration, Affinity, and Tooling at Scale

Stay ahead with the world's most comprehensive technology and business learning platform. With Safari, you learn the way you learn best. Get unlimited access to videos, live online training, learning paths, books, tutorials, and more.

The concepts of blended or shared responsibilities, blameless postmortems, and speed vs. They realize that the future of their business will be highly impacted by their ability to deliver software faster via new features, new products, and new routes-to-market , but they struggle to find a language or framework to communicate to their teams and peers about how to make this happen. Far too often, we ask them to learn the language of lean manufacturing or the principles taught by W. If you started 20 years ago, Client-Server was just beginning to gain momentum. If you started 10 years ago, the iPhone had barely been introduced and AWS had 2 services. Be the change-agent that the business requires. From there, it works backwards to determine how frequently this can happen, which leads to an examination of existing skills people , ability to manage deployment frequency culture and if the right tools and platforms are in place technology. And they need to adapt their applications to become mobile-native in order to address emerging customer expectations. Putting first things first, these core elements need to be in place before the business can expect to be successful at rapidly deploying software into production. Automation – It is critical to build core competency in the skills and tools e. Ansible needed to automate repetitive tasks, both for applications and infrastructure. Building skills to manage pipelines and the process around frequent software updates is critical towards have the framework in place to manage frequently updated software applications. Managing the day-to-day activities of applications is the role of an application platform e. For many years, companies tried to built and maintain their own application platforms, but that approach is rapidly changing as companies realize that their value-add is in the applications, not the platform. Once these elements are in place, many IT teams are ready to start containerizing their existing and modern applications. Far too often, the DevOps discussion is framed as the tension and disconnect between Development and Operations teams. From the State of DevOps report , we see that Gene Kim and team measure the speed at the point when developers push code into source control e. Even in a microservices environment, it can take several weeks or months to actually develop the software features. So how do teams potentially get to a Win-Win scenario? Here are a few suggestions: For Operations teams, adopting automation tools e. Ansible and Infrastructure-as-Code principles e. For Development teams, insist that Security people are embedded within the development process and code review. Security should not be an end-of-process step, but instead embedded in day-to-day development and testing. For both teams, require that automated testing are becoming part of normal updates. Six or seven years ago, nearly every CIO said that they wanted to try and emulate the output of web scale giants like Google in terms of operational efficiency servers per 1 engineer and be more responsive to their developers and business. Unfortunately, at the time, it was difficult to find examples of companies outside of Silicon Valley that could actually execute at a similar level. And the technology Google used was not publicly available to these CIOs. But times have changed significantly over the last few years. Not only is Google and other web scale technologies readily available via open source projects , but the examples of Enterprise companies implementing similar success are plentiful examples: So before a CIO sends a few architects out to Silicon Valley to study with the masters, it might be more valuable to study similar companies to their own. It will allow them to bring back experience that is industry-specific, region-specific, and use-case-similar. Sometimes the right answer is to leverage the broad set of engineers that work on popular open source projects. The challenge with this is that org charts and financial incentives e. If the goal is a specific output e. To achieve this, it is critical for cross-pollination of ideas. Teams need to share their goals, challenges and resource availability with other teams. It would be easy to say that IT organizations need to make sure that their teams are keeping up-to-date on training and new skills. But all too often, this becomes a budget line-item that sometimes get ignored. Encourage employees to seek the best way for them to learn e. Make it a KPI to improve the skill levels and skill diversity of the entire team, with incentives for individuals and the overall team to get better. The Importance of Storytelling The 7 Habits framework is one that has proven to be successful in helping for individuals and groups to improve

their interpersonal skills. Those skills are at the core of any cultural transformation. One of the most important skills that IT leaders can leverage as they drive transformation within their DevOps journey is the ability to tell stories of success and change. The storytelling skill is important because it can inspire emotions and it can help spread successes from group to group. Storytelling allow groups to personalize their challenges, and align their solutions to the cultural nuances within their organization or group.

Chapter 5 : The Seven Habits Of Highly Effective DevOps

Read this e-book from O'Reilly, Effective DevOps, for ideas to improve collaboration, affinity, and tool usage across your organization. DevOps is a culture and practice that unifies people, processes, and tools across development and operations to help you deliver software faster and more reliably.

Evaluate Weigh the pros and cons of technologies, products and projects you are considering. Here are seven qualities to look for when deciding among third parties. Share this item with your network: What is it exactly that makes DevOps consultants particularly qualified at bringing together development and operations tasks? It turns out that, like many other things in life, it takes seven habits. Good DevOps consultants set goals pertaining to things like reduced bugs, deployment times, frequency of deployments and software quality. But you must set goals before you start. So, before embarking upon a DevOps transition, effective DevOps consultants will baseline everything from deployment frequencies to the amount of time it takes to recover from a broken build. Anything that could change should be baselined. All one has to do is gather the metrics used to create the baseline and evaluate the results. Adopt new tools Reducing defect conversion time -- that is, the amount of time it takes to fix a defect after it has been identified -- is an often overlooked benefit of DevOps. Well-established choices, like Chef and Puppet, make it possible to automate the software development environment. New tools -- like those offered by Go2Group , a DevOps tools and solutions vendor -- are good for tasks like eliminating the need for departmental sign-offs and time-consuming interdepartmental requisition tasks. Effective DevOps consultants know that taking the time to learn and adopt new tools can save significant time in the long run. When DevOps works, goals are being achieved, and the metrics indicate that the plan is working. So, then, the culture will shift accordingly. Instead, corporate attitudes change when they see DevOps models starting to work. Embrace alternate technologies Are there ways that new technologies -- alternate data streams or other digital channels -- can be integrated into the continuous delivery pipeline to reduce deployment times or increase software quality? Maybe there are opportunities for AI to help identify potential issues in the code base? Could drones capture data streams that could reduce the reliance upon the human element in any of the largely human-based interactions that are required when the user interacts with the system? An effective DevOps consultant is always looking for ways to automate tasks, while also improving the overall user experience. Turn failures into successes Things fail. But effective DevOps consultants know that inevitable failures can be chalked up as successes so long as they are dealt with appropriately. Are there manual tasks that happen when a build breaks that can be automated? Can scripts be written to replicate the common troubleshooting tasks a systems administrator performs? If so, the entire process of getting the system back up when it inevitably goes down can be hastened. This was last published in December Related Resources.

Chapter 6 : The 7 Habits of Highly Effective DevOps â€™ Red Hat OpenShift Blog

Focusing on the Culture of DevOps, Effective DevOps is a welcome addition to the literature on modern software development practice. It is a rare book in the field that begins with people and process, then moves to technology.

Retention is every bit as important as recruitment. The same principle applies when it comes to staffing up effective teams. Instead, focus first on building cross-functional teams. Be clear with them about what level of decision-making authority you want to allow them to have, and stick to it. Integrate two-pizza teams around portfolios Amazon has made waves in the DevOps world by introducing the concept of two-pizza teams: Scaling up and "integrating these resources is much easier said than done," he said. One way to start is to build out portfolio-based organizations, he suggested. From there, this vice president oversees a number of smaller business-aligned DevOps teams that would be led by managers or directors. Some of these smaller teams might support product delivery in an application area such as revenue reporting, accounts payable, or financial reporting. And by closely aligning them within the delivery organization, they are able to have a shared sense of mission to serve a particular business area. It requires that the directors and managers see their role as building a collaborative and trusting environment for their team members. Recruit the right mix The first order of business should be to do a skills gap assessment. Be wary of the pitfall of creating miniature IT organizations with siloed expertise; in the long run that will create more issues than you have today. The best candidates for DevOps teams like working with new, shiny technology. Candidates often look up to these leaders and in some cases join a team solely based on the promise of working with someone they hope to emulate. Teach at universities One way to become a teacher at work is to literally teach elsewhere. Simply look for a connection with your local university to start. Organizing and promoting technical challenges, competitions, or events among engineering students can be a nice starting point. By enabling teams to self-select, they figure out for themselves who they want to work withâ€™and it turns out that a good group of people who like to work together almost could work on anything. The engineers get more curious and engaged around how the customer is using their part of the system. Much of the advice dispensed above is of the common sense variety. A lot of it requires good-old-fashioned hard work and a grounding in reality. And while recruiting is important, retention and training of existing staff are just as important.

Chapter 7 : Seven habits of highly effective DevOps consultants

Effective DevOps is about optimizing the factors that support efficient development, and ensuring that the architecture needed to support any development efforts are sufficient.

Chapter 8 : DevOps - Effective Project Management Consultancy

Book Description: Some companies think that adopting devops means bringing in specialists or a host of new tools. With this practical guide, you'll learn why devops is a professional and cultural movement that calls for change from inside your organization.

Chapter 9 : effective devops | Download eBook pdf, epub, tuebl, mobi

DevOps involves integrating development, testing, deployment and release cycles into a collaborative process. Learn more about the 4 steps to an effective DevSecOps infrastructure. Security isn.