EEA – DevOps Assessment

Author: Michael Sundgaard

Company: Commentor A/S

Customer: European Environment Agency (EEA)

Last Edited: 28.03.2018 16:01

Contents

Executive Summary

At the request of EEA, Commentor conducted a DevOps assessment with the following objectives:

- Understand the current DevOps situation
- Interview key team lead and PM resources
- Interview key development resources
- Interview key operations resources
- Interview key DevOps resources
- Analyse interview data
- Recommend improvements
- Generate and present a roadmap to improve the DevOps process

It is important to understand that DevOps is not just a tool or a portfolio of tools to support the development and operations staff. DevOps is first and foremost a mind-set which must have the full support from IT management and financial sponsors.

The cornerstone in a good DevOps environment is good and efficient communication and collaboration between business, development, tester and operations staff, for that reason it could be called BizDevOps or even BizDevTestOps. In Commentor we often refer to DevOps as BizDevOps, with the assumption that test is a natural interweaved part of development.

To achieve good collaboration and time to market, it is crucial that DevOps is as automated as possible, always keeping human interaction to a minimum. This will significantly speed up the development and execution of any new ideas that are identified by the business or product owners. This is where the tools to support DevOps comes in to play, and thus good tools that fits well with the technologies and mind-set of the organisation adopting DevOps is extremely important.

The interviews

Some interviews were performed on-site at EEA, Copenhagen, while others were performed remotely via Skype due to geographical constraints, as some key persons are situated in Romania and Greece.

The interviews were timeboxed with a duration of 30 minutes. As part of each interview a series of questions were asked, the questions were basically divided in to two sections shown below.

Basic questions

- What do you do on a daily basis?
- What is your involvement with other departments?
- What is your involvement with Biz, Dev, Ops?
- What pains do you have if any?
- Where do you spend the most of your time?

Additional questions

- Would you say that you are using an Agile approach?
- Do you have a frequent rate of production releases from application and business stakeholders?
- Are you using configuration management tools such as chef/puppet?
- Are you developing and testing in an environment similar to production?

- Do you have frequent builds and deployments?
- Are you continuously validating the quality of the operations process?
- Do you have strong feedback loops?
- Do you have a close collaboration between developers and operations that allows them to understand the impact of code changes?
- Are developers focused on the metrics required by the Ops team?
- Do you have a high level of automation on deployments?
- Is the pipeline of Dev–Test–Prod closely monitored for each deployment with immediate feedback?
- Do you have good collaboration and communication?

Apart from the list of questions above, individual questions were asked based on the qualifications of the interviewed party, as well as based on the flow of the interview. Though everyone were asked the same questions, not all were able to provide a clear answer to the above questions.

The main purpose of the interviews was to get an understanding of the perception of DevOps for each individual involved and identified as a key DevOps resource within EEA. Based on that, information was consolidated in to a common understanding and state of DevOps within EEA.

Obviously, many responses often lead to some level of interpretation, and thus the following sections of the document as well as the conclusion does have some level of author interpretation.

Constraints

Though not the key aspect of this analysis, no major constraints were identified, that could act as a blocker for improving or implementing DevOps in EEA. In general EEA seems to be very open minded and ready for adopting new tools and processes, if these are beneficial for EEA and the automated DevOps process, as long as these tool does not compromise the guidelines which EEA must adhere to.

Current Situation

EEA has development decentralized but it managed out of Copenhagen, Denmark. The main teams are in Bucharest, Romania and Athens, Greece. There is a great need for DevOps to ensure high quality of deliveries and to ensure that new resources adhere to the same rules and guidelines as the existing staff.

EEA has already been implementing and adopting the DevOps principles for some time, and are well on their way, to achieving a fully automated and fully integrated DevOps environment. The staff is in general very happy with how things are run, and the collaboration between business, development and operations.

Common challenges

The following issues with the current delivery capability were articulated at the start of the assessment.

Guidance:

- Limited human and/or technical resources for testing
- Manual, ad-hoc, or inconsistent deployments
- Resolving production issues is slow and/or highly disruptive
- Deployments are large and take a substantial periods of time to deliver.

- High infrastructure costs and/or budget constraints
- Challenges in testing
- Lack of collaboration between IT and Development organizations

The following business priorities were articulated at the start of the assessment.

Guidance:

Application Lifecycle Management with Visual Studio and Team Foundation Server

- Team Automony and Enterprise alignment
- Rigorous management of technical debt
- Focus on flow of customer value
- Hypothesis drive development
- Evidence gathered in production
- Live site culture
- Manage infrastructure as a flexible resource

The following tools and processes (strategy) were articulated at the start of the assessment

Guidance: **Application Lifecycle Management with Visual Studio and Team Foundation Server**

Application Lifecycle Management with Visual Studio and Team Foundation Server

- Usage of current DevOps tools
- Challenges that exist in the DevOps tools
- Knowledge of the available solutions and practices
  - Configuration management
  - Infrastructure as code
  - Release management
  - Continuous integration
  - Continuous deployment
  - Application performance monitoring
  - Test automation
  - Source Control Management

Common concerns

During the interviews, we uncovered the following common concerns, where some of the concerns have already been addressed. Yet we find it relevant to note the concerns as they are still present in the mind of the staff.

The old cloud vendor from Italy (ECS) has left its mark amongst many people in the staff. Problems with power outage and memory issues were some of the common issues. In general, the staff is happy that the old cloud vendor is now being swapped with AWS and to some extend Azure. Since the general move to AWS the staff is in general very happy with Cloud.

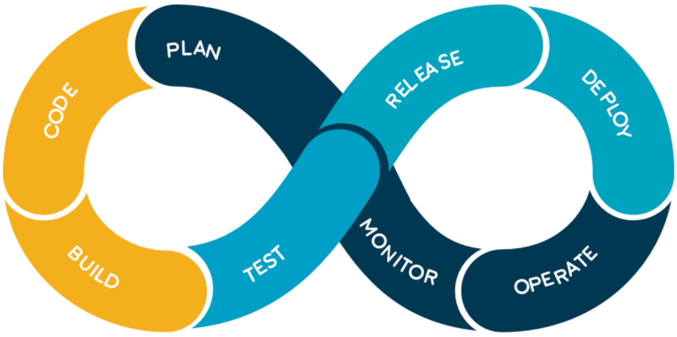
Implementing Rancher in EEA has not been without issues, many of the people involved raise this as their main issue when questioned about DevOps. Even though the implementation of Rancher is now somewhat in the past, it has still left its mark and scars on part of the staff. It is important to note that everyone who has expressed concerns about Rancher now seems quite happy with using Rancher.

Testing is somewhat automated, but in general the staff is somewhat defensive when it comes to testing, especially when it comes to anything else than unit testing. For some it is a general concern, and for others it is an area that should be concerned.

DevOps disciplines

Below is a diagram showing the core disciplines in a DevOps process. We will go through each of the disciplines and describe how the disciplines is currently being handled in EEA, as well as touch upon things that could be improved for that disciplines.

These findings will be summarized and commented further in the final chapters of this document.



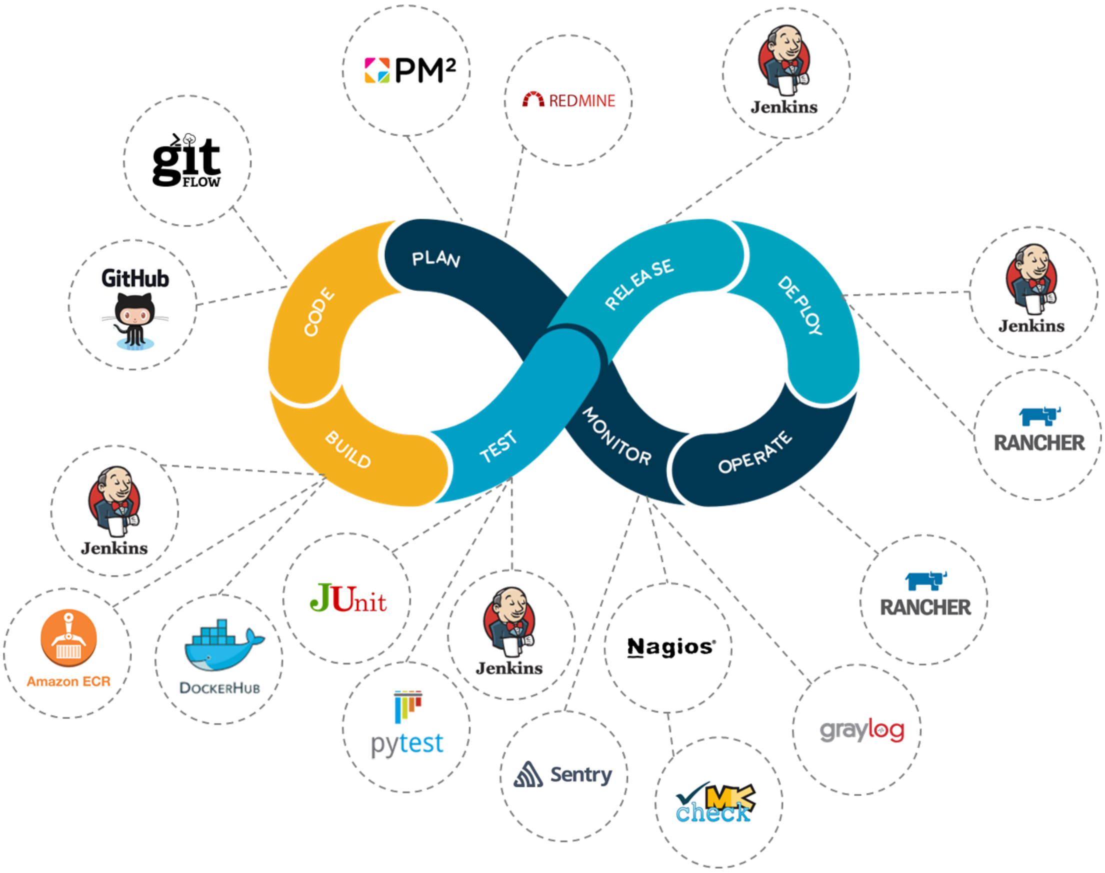
As such we will not explain what DevOps is and what each discipline consists of, but instead we assume (also based on the impression from the interviews) that EEA is already knowledgeable of what DevOps is.

DevOps tools

The figure below is depicting the most important elements in the DevOps process used at EEA. Some elements are used more intensively than others, and most are major main stream technologies.

As we can see the tools vary from source code management tools such as GitHub to project and task management tools such as Redmine Taskman and PM2. For managing build and release pipelines EEA uses Jenkins, and for operating Rancher is used for managing all containers, and containers are stored in Amazon Elastic Container Registry.

For monitoring a combination of Nagios, Sentry, GrayLog and Check_MK (a plug-in for Nagios) is used. Junit and Pytest is used for implementing unit tests for Java and Python based applications



Each discipline will get a score from 0 to 10 and at the end an overall score will be given from the same scale. This score is an expression of the current maturity level and quality of the DevOps process, in this case for EEA.

Plan

For overall project planning EEA is using PM² which is a project management tool with elements from Prince2 and RUP. As such this project model does not collide with DevOps and agile, but it is not highly relevant for the DevOps analysis and thus we will not touch further upon this method in this document. The actual task break-down and daily stand ups are handled via a tool from Redmine called Task Manager or in short Taskman. Taskman is used for registering anything from a new feature to enhancements, general tasks and bugs. Team leads or project managers are usually the roles that register tasks in Taskman at EEA.

For some projects or solutions, the PO (though not always the official PO, but acting PO) can also register tasks in Taskman. Developers themselves will also register tasks in Taskman when needed.

Finally, operations also use Taskman to inform development of issues, and development also use Taskman to inform the operations department of issues. Especially the communication between development and operations is key to a good DevOps process. When tasks are minor and to speed up resolution time instant messaging technologies such as Skype and Jabber are used. Although they don’t offer a good audit trail of what has been agreed, they are great tools for troubleshooting.

Apart from using Taskman for all of the above, PMs and team leads also use Taskman for generating overview and monitoring progress in the form of burndown charts, SCRUM boards and Kanban boards.

Score: 8.5

Code

The primary programming languages used in EEA are Python, Java, HTML and JavaScript. The programming languages as such are not highly relevant for a DevOps analysis, though it is good to know the scope in order to get an idea of whether it is testable, deployable, buildable and packageable.

More importantly all code is stored in GitHub which is a hosted repository (SaaS solution) for storing code and more. GitHub is used primarily using the CLI and through Jenkins. GitHub is furthermore used using the guidelines dictated by GitFlow, though it was not clarified whether or not the GitFlow CLI was used, or if it was merely simulated using the normal Git CLI.

One of the very important steps of a successful DevOps process is a proper branching strategy. EEA has adopted the best practices dictated by GitFlow, which is a great way of handling feature and release branching. With this approach EEA has ensured that they are able to do cherry-picking (the art of deciding which features to include in a release at the end of a sprint), and that developers can work in parallel without major conflicts.

Furthermore, as part of the GitFlow practise, EEA are using pull-requests with code review. This is again an extremely important part of DevOps and an important step to achieve a high quality of code prior to testing and releasing an application or service.

Score: 9

Build

Builds are triggered via Jenkins both nightly (for some projects) and when a push is made to GitHub. As part of the build process Jenkins builds a corresponding Docker image, which is then stored in DockerHub. It seems that this part is not clear to all, and thus it is unclear whether or not DockerHub.com is used or whether it actually is Amazon Elastic Container Registry (ECR). Amazon ECR is a fully-managed Docker container registry just like DockherHub.com.

As such it is not that relevant for this analysis whether Amazon ECR or DockerHub.com is used, but for documentation and information purposes, this should be clarified. Also, there is a plan to better scale the Docker container registry used, and for that both DockerHub.com, Amazon ECR and other options such as ACR (Azure Container Registry) should be considered. The plans to scale should be seen as a result of poor performance with the currently used container registry.

For a successful DevOps process having an automated build process is crucial. This is the case in EEA, as both nightly scheduled builds as well as triggered build is already configured and running. Also, the involved parties are actively monitoring the result of the build, and taking the necessary actions when a build error occur.

Score: 9

Test

As part of the build tests are run in an automated fashion. It has not been analysed in detail for which project and to what extent unit tests are run, but the common understanding and feedback is that automated unit tests are developed and run for most projects. Apart from unit tests some functional tests and integration tests are run, but this seems to be much more sporadic and much less extensive.

All automated tests are triggered by Jenkins as part of the CI pipeline, and thus the DevOps process around testing is somewhat automated in place, though it should have a bit more depth and content. This is further described under “Recommendations”.

Score: 6.5

Release

For creating releases Jenkins is used. This means that all release pipelines are defined in Jenkins, and Jenkins will keep track of a given release and the related documentation and features in the release. It is uncertain what level of release documentation is being created with a given release. Though Taskman’s Wiki functionality is used for documentation purposes to some extent, it was not clear whether release documentation is done automatically or on an ad-hoc basis.

Score: 8

Deploy

Deploying a given release is done with a combination of Jenkins and Rancher. Jenkins will pick up the release and deploy it to Rancher. This process is true for the vast majority of the solutions in EEA. Some solutions are not containerized and follow a slightly different process, but as these applications are reaching end of life, they will not be discussed further in this document.

Score: 9

Operate

Rancher is the primary tool for operating and managing the running Docker based applications. Though there have been some starting difficulties and some level of frustration around Rancher, there seem to be a broad and general acceptance of using Rancher and the qualities of Rancher as a container management platform as such.

Score: 8.5

Monitor

Monitoring is done via Nagios, Check_MK (Nagios plugin), Sentry and Graylog. In general, it seems to be the case in EEA that Nagios, Check_MK and Sentry are used mostly for server and infrastructure logging, whereas Graylog is used for application logging. This was the general response from the interviewed parties, but the picture of which logging system is used for what is a bit unclear.

Apart from the above also Monit is used for monitoring Linux processes and servers, and to troubleshoot hardware issues HP Insights is used.

Most of the above logging systems have enterprise capabilities and are great tools that support the DevOps process well. At first glance though it may seem like there are one or two logging systems too many considering the needs and size of EEA.

Score: 8.5

Recommendations

Though not many, there are a few areas of the DevOps process which should be improved in EEA. Some of the issues are already being addressed, though others have not yet been planned.

Key areas for Improvement

After analysing the input from the interviews, it is clear that especially testing, and more precisely the types of tests should be improved and intensified. Unit tests are being developed to some extent, but the test coverage varies. It was hard for the developers to put precise test coverage percentages on the various solutions when I asked, and this most likely means that tools and data are missing to report the test coverage percentages.

Other areas to improve

Although the interviewed parties seem to be happy with Taskman and the capabilities of Taskman, it is my concern that the community and the future of Taskman is uncertain. The documentation and information for Taskman is almost non-existing in comparison with the major main-stream alternatives.

Though not an urgent matter it is my recommendation that Taskman is replaced with one of the major alternatives, such as Jira or VSTS. Jira seems to be the best fit for EEA, as EEA is using the open source technology stack, which means that there is already seamless integration to many of the chosen technologies. Also without knowing Taskman in details, it is my impression that Jira will be able to handle all the scenarios that Taskman handles today, and more. A transition from Taskman to Jira could be done in steps, and thus a big bang migration is not needed.

It is our recommendation to start with a PoC and after and well ended PoC all new projects should be handled in Jira going forward. The risk of continuing with Taskman would be that documentation and resources with knowledge of Taskman will fade over time.

As Taskman is also used for documentation purposes it is our recommendation to introduce Confluence for documentation purposes, as Jira does not handle this part. Confluence and Jira are both provided by Atlassian and the most widely used tools for task management and documentation in companies using open source technologies for CI/CD.

As an alternative to Jira and Confluence similar features exists in GitHub which could be a light weight alternative, though it is uncertain whether or not this would suffice for EEA.

It was somewhat unclear at the end of the interview rounds whether or not tasks or bugs gets created automatically in the case of failed builds, tests and releases. If this is not automated the Jenkins pipelines should be configured to automate this process.

There are quite a few logging systems involved, which is described under the “Monitor” section of this document, it may be beneficial to reduce the number of monitoring systems to one or two systems. There is a risk that sustaining knowledge of four or more monitoring systems will be difficult or time consuming over time, and thus the investment in understanding and using the a given monitoring system may outweigh the benefit of the monitoring system itself.

Market trends

One of the things to consider when we want to ensure a good and smooth DevOps process, is the popularity and communities around the technologies we choose to support the various disciplines in the DevOps process.

Planning tools

As mentioned, the community and popularity of Redmine Taskman, which plays a relatively central role in EEA is very narrow and popularity is very sparse. Looking at the Gartner Magic Quadrant for Agile Planning Tools it clearly shows that Redmine is not at all considered to be within the quadrant at all. It should be noted that many other alternatives to the ones depicted in the quadrant are not part of the quadrant either.

Gartner Magic Quadrant

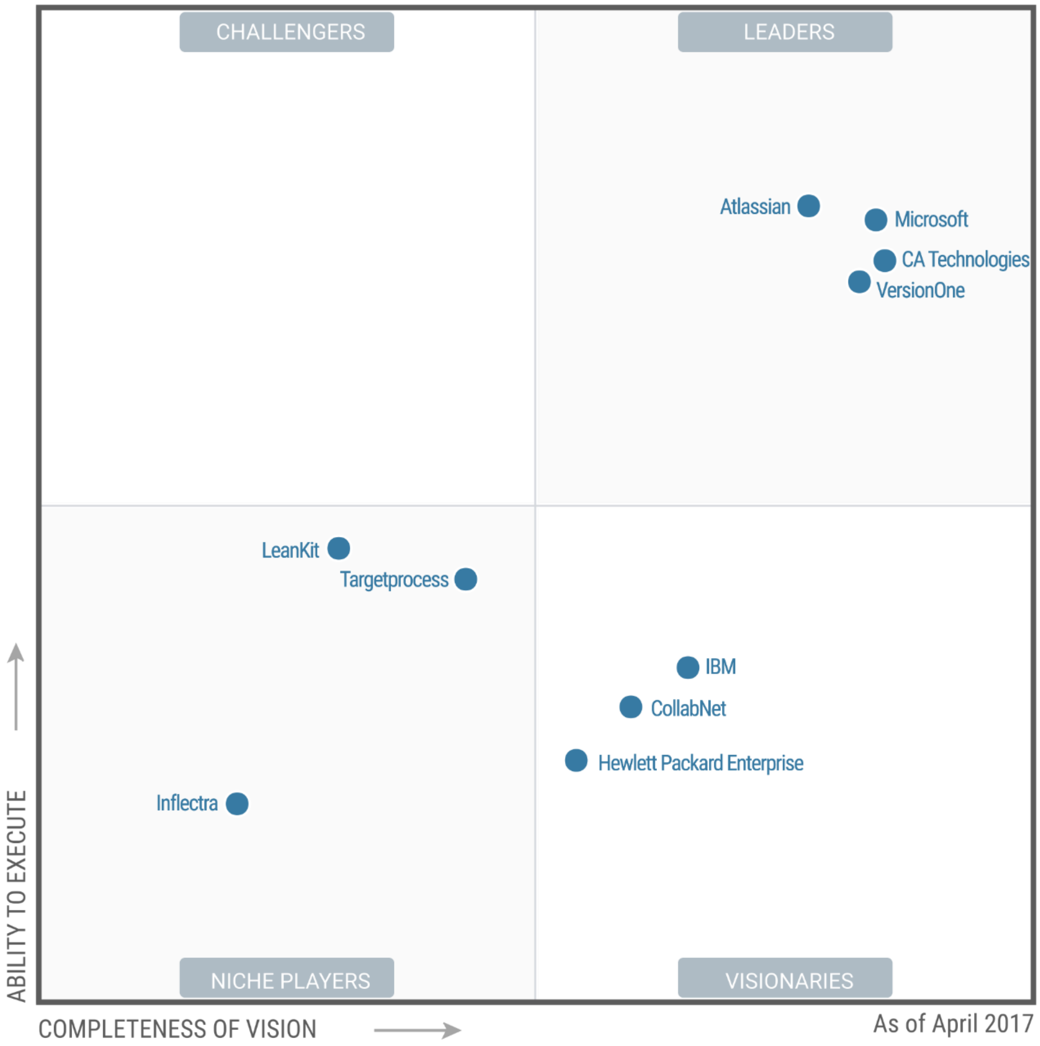


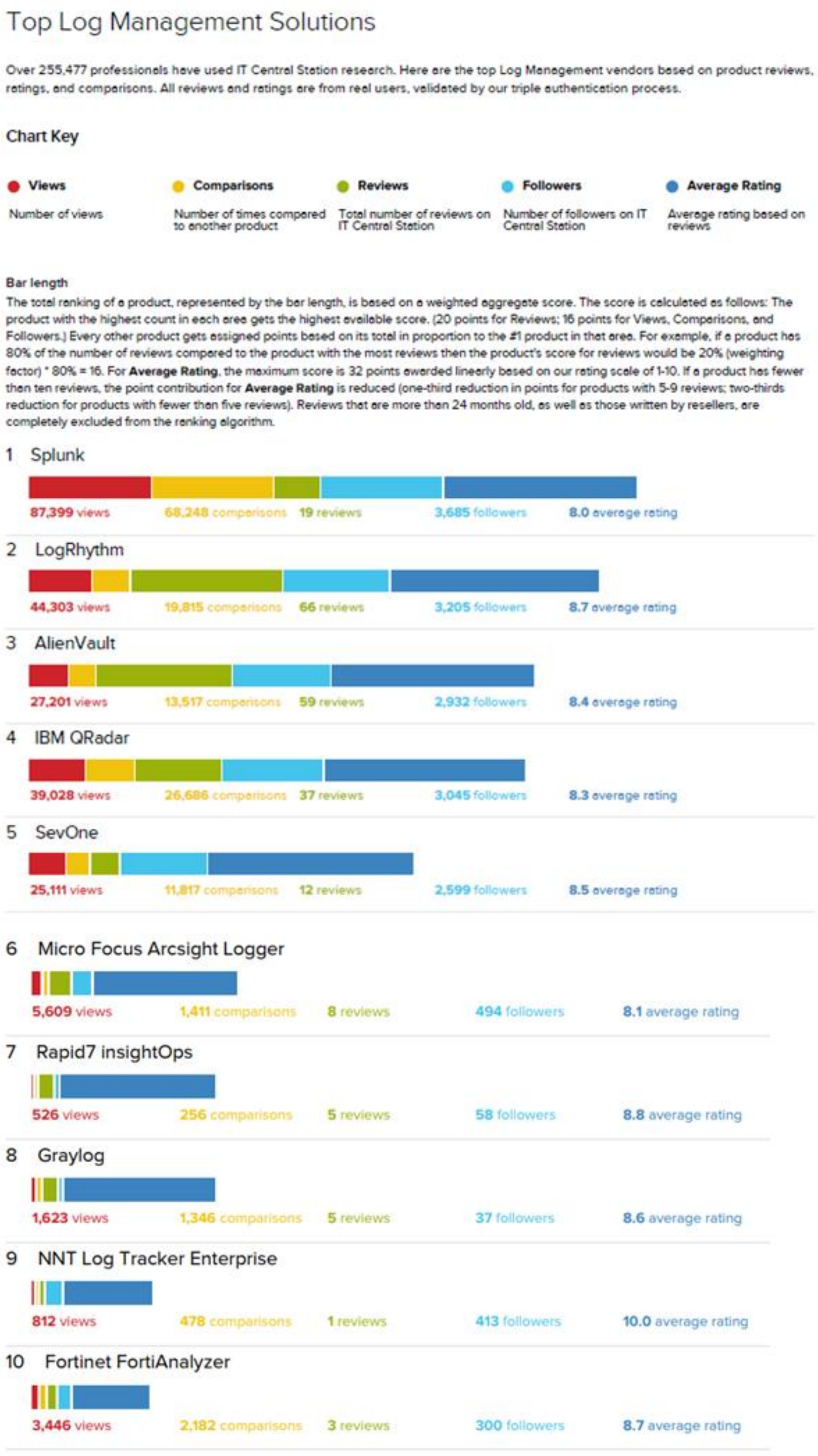
Figure 1. Magic Quadrant for Enterprise Agile Planning Tools

Having said that, not being in the quadrant is a risk, and with many new tools surfacing and disappearing again on a monthly basis, the safer choice is one of those in the quadrant. The preferred picks are the ones that are in the Leaders part of the quadrant and in the case of planning tools clearly Atlassian, Microsoft, CA and VersionOne are taking the lead.

It is for those reasons that we recommend that one of these tools should be used in EEA going forward.

Log management tools

One of the primary tools used for log management in EEA is Graylog. Though not the most popular tool currently in the market it is still placed at a seventh place in a large log management tools review performed by IT Central Station.



Top Log Management Solutions

Over 255,477 professionals have used IT Central Station research. Here are the top Log Management vendors based on product reviews, ratings, and comparisons. All reviews and ratings are from real users, validated by our triple authentication process.

Chart Key

● Views: Number of views

● Comparisons: Number of times compared to another product

● Reviews: Total number of reviews on IT Central Station

● Followers: Number of followers on IT Central Station

● Average Rating: Average rating based on reviews

Bar length

The total ranking of a product, represented by the bar length, is based on a weighted aggregate score. The score is calculated as follows: The product with the highest count in each area gets the highest available score. (20 points for Reviews; 16 points for Views, Comparisons, and Followers.) Every other product gets assigned points based on its total in proportion to the #1 product in that area. For example, if a product has 80% of the number of reviews compared to the product with the most reviews then the product's score for reviews would be 20% (weighting factor) * 80% = 16. For **Average Rating**, the maximum score is 32 points awarded linearly based on our rating scale of 1-10. If a product has fewer than ten reviews, the point contribution for **Average Rating** is reduced (one-third reduction in points for products with 5-9 reviews; two-thirds reduction for products with fewer than five reviews). Reviews that are more than 24 months old, as well as those written by resellers, are completely excluded from the ranking algorithm.

1	Splunk	87,399 views	68,248 comparisons	19 reviews	3,685 followers	8.0 average rating
2	LogRhythm	44,303 views	19,815 comparisons	66 reviews	3,205 followers	8.7 average rating
3	AlienVault	27,201 views	13,517 comparisons	59 reviews	2,932 followers	8.4 average rating
4	IBM QRadar	39,028 views	26,686 comparisons	37 reviews	3,045 followers	8.3 average rating
5	SevOne	25,111 views	11,817 comparisons	12 reviews	2,599 followers	8.5 average rating
6	Micro Focus Arcsight Logger	5,609 views	1,411 comparisons	8 reviews	494 followers	8.1 average rating
7	Rapid7 insightOps	526 views	256 comparisons	5 reviews	58 followers	8.8 average rating
8	Graylog	1,623 views	1,346 comparisons	5 reviews	37 followers	8.6 average rating
9	NNT Log Tracker Enterprise	812 views	478 comparisons	1 reviews	413 followers	10.0 average rating
10	Fortinet FortiAnalyzer	3,446 views	2,182 comparisons	3 reviews	300 followers	8.7 average rating

This placement should be seen in the light of a great number of logging tools being available in the market, and the list does not even include major log management tools such as Logstash, SumoLogic and Loggly.

Based on this and a number of other articles and our experience recommending and using modern log management systems, it should be safe to assume that GrayLog will also be in the leader's quadrant some years from now. It should be mentioned though that the replacement GrayLog2 should be preferred over GrayLog. It is a bit unclear based on the answers from the interview whether GrayLog or GrayLog2 is actually used.

Conclusion

First of all, it has been a pleasure working with EEA, and the EEA staff. I was a bit surprised to discover a very positive and homogeneous organisation. When interviewing many other organisations, it has often been a frustrated staff with many boundaries and manual procedures. This does not seem to be the case in EEA, which is great.

All interviewed parties have been asked a series of questions in regards to the DevOps process, the answers to these questions has been consolidated in this document. When asked about daily pains and challenges running DevOps, all interviewed parties unanimously replied that they have no pains or challenges as such in their daily work, and furthermore communication across teams and departments are smooth and trouble-free.

The intention of the analysis has been to pinpoint issues in the DevOps process, and not to invent issues or come up with alternative solutions to already working processes. As the DevOps process is working well in EEA, the reader of this document may get the impression that not all disciplines in the DevOps process has been accounted for, which is not the case. It is simply due to a well-oiled machine when it comes to DevOps, with exception of the few recommendations outlined in the previous section.

Commentor evaluates an overall score of 8.5 out of 10, which is a very high and extremely good score, with only some or minor improvements needed to the DevOps process. Individual scores have been given for each individual discipline in the DevOps process, and the overall score is an expression of those scores, combined with the general feeling and impressions of the reviewer.

The growth of a mature DevOps practice is not a direct one. As the team becomes more efficient, requirements become refined and testing results become more transparent, and there will be new areas of concern that will emerge.

As DevOps maturity levels move through the next stages (for example, Repeatable to Reliable), some common "growing pains" will begin to emerge. These common "growing pains" will be the best choices for additional assessments and guidance to help assist with the transition for the business, teams, and individuals.

We recommend that the comments outlined in this document be validated during the initial deployment and as projects and teams are brought on board the system. Teams constantly have to adapt and change their processes as the business and technical environment changes around them.

To encompass all of the recommendations in this document, a schedule for all of the relevant tasks should be created. Complete implementation and customization should be done by EEA operations staff.

Main DevOps practices in EEA

Depending on the type of organisation and the type of tasks and projects that are worked on, in a given organization, some DevOps disciplines have more focus than others. In EEA the complexity of the applications are relatively simple, and the maturity level of the resources involved in DevOps in EEA is relatively high.

The people that were interviewed were relatively technical and skilled in many agile and DevOps related disciplines. Especially the processes around continuous integration and continuous deploy were in focus in the minds of the staff. This means that creating, updating and maintaining backlogs, as well as coding and branching based on a backlog is a natural element of the daily work at EEA.

Focus is on following the guidelines dictated by GitFlow which means focus on code reviews, feature branching and automated build, test and deploy mechanisms.

Backlog

Schedule and Team

Technical Debt

Flow

Evidence

Production

Cloud

We have found that EEA is mainly performing DevOps at a <Level> level and should focus on incrementally improving.

Getting the Most from Your DevOps Plan

Our recommendations for optimizing the DevOps practices and tools in your environment are detailed within this document. Please take your time to review the findings and ask any follow-up questions necessary. Depending on the capabilities of your teams and organization, you may elect to try the DevOps improvements in-house or contract with an outside consultant. In either case, this plan should be given to the party responsible for the work and used as an implementation guide.

Existing Best Practices

Our interviews surfaced the following Best Practices that are being used by teams at EEA. These practices are:

-

We recommend that these practices continue to be employed and are continuously evaluated and improved in order to promote process optimization.

Existing Hindering Practices

Guidance: The purpose of the existing hindering practices are to point out the pain points the team is experiencing from the lack of DevOps practices. The following are some examples.

Our interviews surfaced a few top issues that are interfering with growth in DevOps maturity. Adopting a new mindset around these issues and prioritizing improvements in these areas will yield significant results.

DevOps progression Framework/Readiness Model

The model below shows the different practice areas and maturity levels of DevOps at a broad scale. Levels of DevOps maturity may vary across teams, even within a common department or division, and the same organization may have significantly more advanced practices in some categories than others. Enhancing the entire workflow to achieve the best possible DevOps process throughout the enterprise should be the end goal of the team and individual effort.

Current vs. Desired State

- BACKLOG
- SCHEDULE & TEAM
- TECHNICAL DEBT
- FLOW
- EVIDENCE
- PRODUCTION
- CLOUD
- FOUNDATIONAL
- Releases: ad-hoc
- Decisions: brute force
- Teams: siloed
- Goals: unknown
- Limited or no public cloud
- REPEATABLE
- Releases: periodic
- Decisions: anecdotal
- Teams: communicative
- Goals: inconsistent
- Individual teams working with VM's
- RELIABLE
- Releases: frequent
- Decisions: corroborated
- Teams: collaborative
- Goals: aligned
- Corporate wide with no team autonomy
- ASPIRATIONAL
- Releases: continuous
- Decisions: evidence-based
- Teams: unified
- Goals: shared
- Public Cloud fully embraced

Guidance: This section will give an overview of the current state of the organization and how it matches with the maturity model and the organization's Ideal State.

Application Lifecycle Management with Visual Studio and Team Foundation Server

The current DevOps maturity levels at EEA, on a scale of 1 (Foundational) to 4 (Aspirational), for each practice area are illustrated below. The team has some good practices in place and the enthusiasm to achieve a higher level in several key categories.

Guidance: Give key examples here for some or all of the DevOps Progression Framework practice areas, focusing on the most impactful opportunities and/or customer's priority goals.

Application Lifecycle Management with Visual Studio and Team Foundation Server

Roadmap

Based on our observations and discussions, we recommend that the following iterative roadmap be implemented in order to understand and instill DevOps best practices within the teams.

Please note that the areas for improvement mentioned in the prior section, which are marked as urgent, may not be addressed immediately. In some cases, the foundations for improving a particular service area will not be in place in the first or second iteration.

Brainstorm (to be removed)

https://github.com/eea?page=3

https://taskman.eionet.europa.eu/agile/board?query_id=551

Code Reviews

Feature Branching

Docker

DockerHub (on-site soon, scalable) / Amazon ECR

(HelpDesk / TopDesk / ????)

Rancher Alarms

Monit (Linux monitoring)

HP Insights (Hardware problems)

Guidance: This template includes guidance blocks and wording examples. Prior to handing over the document, remove the guidance blocks (like this one) and replace any highlighted sample text in <brackets> with your findings and recommendations.

Testing and ALM (to be removed)

- ALM Rangers Guide – Test Release Management: http://vsartestreleaseguide.codeplex.com/

This Visual Studio ALM Ranger project has the primary goal of delivering scenario based and hands-on guidance for managing Microsoft Test Manager Test plans.

- ALM Rangers Guide – Visual Studio Test Tooling Guidance: http://vsartesttoolingguide.codeplex.com/

This umbrella project delivers a range of practical and scenario based guidance for Visual Studio test features, such as Coded UI, Microsoft Test Manager, IntelliTrace, and Microsoft Fakes.

Automated Build-Deploy-Test (to be removed)

- Setting Up Automated Build-Deploy-Test Workflows: http://msdn.microsoft.com/en-us/library/hh191495.aspx

You can use a build-deploy-test workflow to deploy and test your application when you run a build. This lets you schedule and run the build, deployment, and testing of your application with one build process. Build-deploy-test workflows work with Lab Management to deploy your applications to a lab environment and run tests on them as part of the build process.

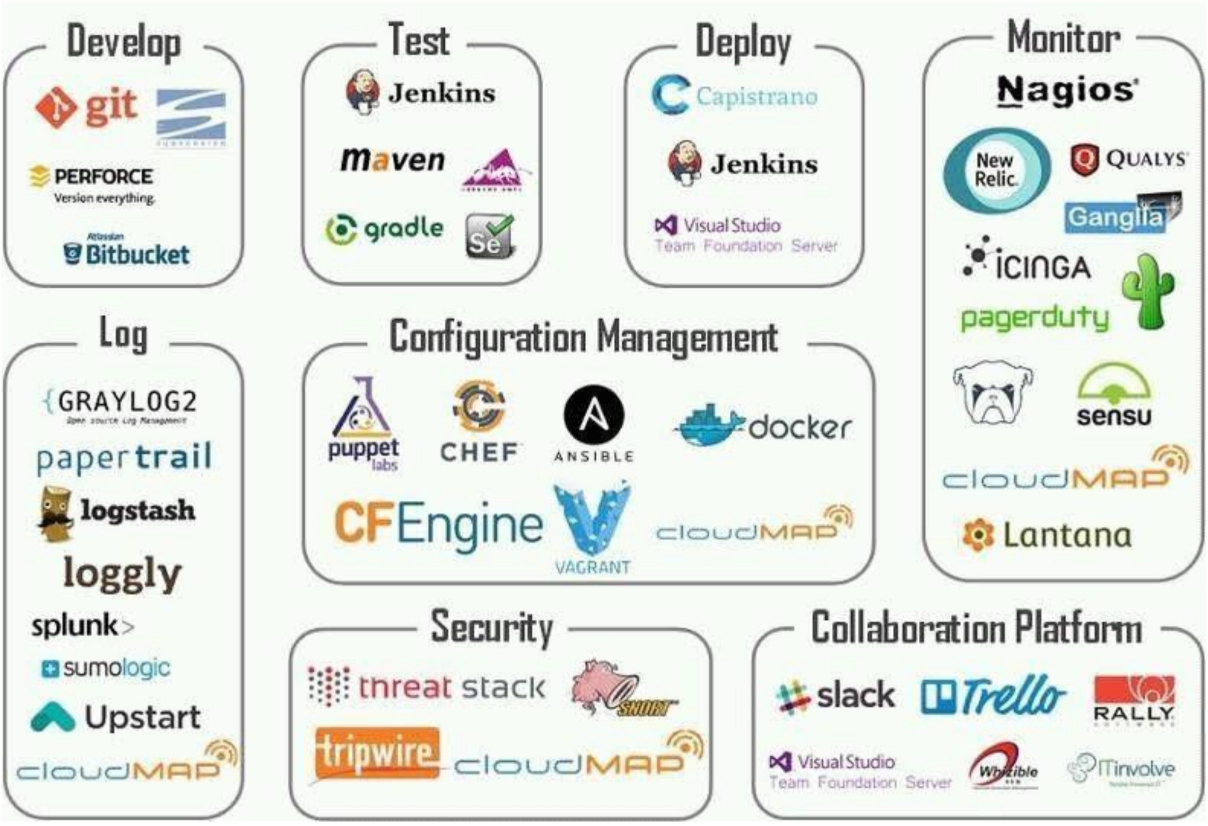
- Visual Studio Team Foundation Build Customization Guidance: http://vsarbuildguide.codeplex.com/

This Visual Studio ALM Ranger project has the primary goal of delivering scenario based and hands-on lab guidance for the customization and deployment of Team Foundation Build.

General ALM Resources

- What’s new in Visual Studio 2015: https://www.visualstudio.com/news/releasenotes/vs2015-update3-vs
- https://msdn.microsoft.com/en-us/library/bb386063.aspx

- Getting started with Application Lifecycle Management: http://msdn.microsoft.com/en-US/library/vstudio/dd286491(v=vs.110)


Develop
git
PERFORCE
Version everything.
Atlassian
Bitbucket
Test
Jenkins
maven
gradle
Se
Deploy
Capistrano
Jenkins
Visual Studio
Team Foundation Server
Monitor
Nagios
New Relic.
QUALYS
Ganglia
icinga
pagerduty
sensu
cloudMAP
Lantana
Log
GRAYLOG2
Open source Log Management
papertrail
logstash
loggly
splunk>
sumologic
Upstart
cloudMAP
Configuration Management
puppet labs
CHEF
ANSIBLE
docker
CFEngine
VAGRANT
cloudMAP
Security
threat stack
SNORT
tripwire
cloudMAP
Collaboration Platform
slack
Trello
RALLY
Visual Studio
Team Foundation Server
Whizible
ITinvolve