DevOps Assessment

MIT - Cloud & DevOps - Continuous Transformation

Brief description

eSales (https://esales.com.br/ - please use Google translate on it since there's no english version so far) is a company that provides solutions (digital products) for the whole supply chain, logistics and file transfers. We count with a relevant number of important brands as our customers in Brazil. A few examples are Unilever, Outback, Burger King, and Natura.

There's also another company in the group which is a leader for government solutions for file transfer in Brazil.

Participant’s bio

Guilherme Sesterheim (https://www.linkedin.com/in/gsesterheim/) is a technical leader with a deep background on open source software, agile methodologies and AWS cloud. He’s a masters degree in applied computing sciences by Unisinos and a business-oriented vision on how to apply technology to solve businesses problems.

Introduction & Company challenge

I’m new to the company and I was hired especially to lead the technical implementation of DevOps on the two main products and to lead the journey to transform them on cloud hosted and cloud native in the future.

1. Currently the main product is a PHP-written monolith working on top of a big Oracle database. There is no CI/CD pipeline, no automated testing, no infrastructure as code, no disaster-recovery plans and no high-availability practices/strategies.
   a. The main challenge for this one is to modernize the application and give: (1) more autonomy to developers, (2) speed and faster time-to-market to business owners and (3) more reliability to the supporting teams.
2. The second product will start to be developed in the upcoming months.
   a. The main challenge for this new product, which in the future will replace the other one, is to design and implement an architecture that solves several situations such as: (1) bad performance, (2) Oracle lock-in, (3) scalability, (4) cost reduction and (5) reliability.

IMPORTANT: The company leadership is already on board of the cloud-moving journey, so there is no more discussions about IF we are going, but yes HOW and WHEN we are going. My mission is to give those answers, and lead and execute the technical approach.

Solutions / Project stages / Implementation

1. For the **existing product** my current macro strategy is as follows:

Order	Timeline	Action	Expected impact
1	08/05/20	Make the developers start writing their own unit and integration tests as code so they can be automated. The existing QA professional on the team will slowly stop testing manually and start writing higher levels of tests like: contract, end-to-end and stress.	More reliability to the software
2	Aug/20	Implement a CI/CD pipeline to automatize the deploy practice (which has several problems and dependencies) and start monitoring three metrics that don't exist nowadays: (1) **% tests coverage**, (2) **deployment frequency, and** (3) **change fail rate.**	More autonomy to developers More data to take decisions
3	Oct/20 to Aug/21	Create and implement a plan to move the whole application to the cloud "as-is"	Cost reduction. More reliability. More speed to apply changes to the environment. Implement an HA and DR plan.
4	Oct/20 to Aug/21	Migrate the Oracle to an open source database	Cost reduction

Solutions / Project stages / Implementation

2. For the **new product** my current macro strategy is as follows:

Order	Timeline	Action
1	Aug/20	Understand the application big picture, requirements, possible bottlenecks, and current pain points. Also, understand the client current feelings about the application and possible ways to enhance its experience.
2	Aug/20 for first version. But will keep going on indefinitely	Design and implement the new application architecture using best practices such as: (1) cloud native development, (2) open source databases, (3) adopt NoSQL databases to key features, (4) infrastructure as code, (5) immutable infrastructure, (6) at least 4 layers of automated tests: unit, integration, end-to-end and stress.

Risks

1. Developers' resistance to the test culture delaying improvements on the legacy pipeline and generating conflicts on the team.
2. Moving the legacy application to the cloud 'as-is' too fast (not following the appropriate strategy) because of increased costs in the beginning, making it more vulnerable and degrading the cloud within the company.
3. Moving and converting the database of the legacy application to an open-source too fast due to increased costs in the beginning, making possible errors traumatizing and making the open-source vulnerable within the company.
4. Not applying UX practices to the new system, losing the opportunity to listen the client needs appropriately and make the tool even more competitive.

Mitigations

1. For: “Developers’ resistance to the test culture delaying improvements on the legacy pipeline and generating conflicts on the team”:
    a. Create a strategy led by leadership and close to the HR best practices to enroll people, show the benefits and convince them of long term benefits to they want to create their own tests instead of feeling pushed to it.
2. For: “Moving the legacy application to the cloud ‘as-is’ too fast (not following the appropriate strategy) because of increased costs in the beginning, making it more vulnerable and degrading the cloud within the company.”
    a. Create a strategy for the cloud move making small portions of clients to migrate little-by-little and watch as they grow. Show the risk to the leadership so they compromise on the small costs in the beginning and allow the strategy to be followed.
3. For: “Moving and converting the database of the legacy application to an open-source too fast due to increased costs in the beginning, making possible errors traumatizing and making the open-source vulnerable within the company.”:
    a. IF it is possible to convert the big database into a big open source database (faster solutions:)
        i. Create a strategy for the database conversion and migrate portions of clients little-by-little and increase the number of clients migrated after watching their usage.
    b. IF it is not possible to convert the big database into open source and the strategy of breaking the big application in microservices have to be adopted earlier:
        i. Create a roadmap of microservices to be created and break the big database little by little.
4. For: “Not applying UX practices to the new system, losing the opportunity to listen the client needs appropriately and make the tool even more competitive”:
    a. Build a case showing UX benefits to a digital product to the leadership and create a small pilot project.

Conclusion

There are several big challenges ahead, that probably will take something between 2 to 3 years to be finished. Some key benefits and ROI can be achieved in the very beginning with small moves, but the whole company leadership has to feel confident on the plan as they see first steps happening.

Metrics must guide the whole implementation so the benefits can be shown and followed by the whole company.

Thank you colleague for sharing your knowledge and feedback with me :)