

Performance Engineering in air-gapped environments and teams



Carlos Ortega
Sr. Tech Scrum Master



Agenda

- Personal background
- About this talk
- Broad Overview
 - The company
 - The ecosystem and the environment
 - My teams
- Our own team agile framework
- The Challenges
- What we did
- Our mistakes
- Organizational Dynamics
- Our response
- Lessons Learned
- Questions - Comments

Personal Background



About me

- My name: Carlos Ortega
- Studies: Computer Engineering in Mexico City
- Roles and Experience: > 20 years
 - Business Analyst
 - Developer
 - Software Architect (6 years)
 - Project Manager
 - Scrum Master (14 years)
 - Agile Coach (1.5 years)
- Different industries:



Biotech



Airlines



Banking



Insurance



Oil

About this talk





Objectives of the talk

- Share personal experiences of the infrastructure teams in which I was involved.
- Provide a glimpse of the “day to day” activities, the products we build, the environment, the challenges, the heuristics we applied, our misses and hits, and our lessons learned.
- By no means this talk try to be a set of recipes or practices that should be applied to any kind of situation =>

Our decisions were framed within the context of our own environment, conditions and history and what may have been successful for us may not apply to your organization or team.

Broad Overview





The Company

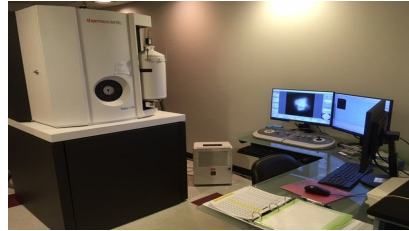
- The largest and most important biotechnology company in the world.
- Provides solutions to research and development organizations and companies associated to life sciences, biochemicals & pharmas.
- The strongest “right-arm” of the life science and biochemical/pharma scientists.
- Presence in 29 Countries.
- Earnings (2021): 39.21 Billions USD & > 80,000 Employees world wide.
- **All Major COVID Vaccine Producers are customers of the company.**





The Ecosystem

- Solutions compounded by instruments (hardware), consumables, software, installation, training.
 - Example: Electronic Microscope used to define the COVID 19 Virus profile in China.



- Software:
 - > 2000 IT persons (BioInformatics, Physicist, Chemical Eng, Electronic Engineers, Computer Engineers,)
 - 2 CoE - Tijuana and Bangalore
 - Aprox 800 developers Areas ranging: Tech Support, Security, SecOpsMobile, IoT, Data Science
 - Hundreds of clusters (AWS, MS Azure), Salesforce, Oracle.
 - OS: Linux, Windows
 - Clients: Android, IOS,Windows, MacOS.



The Teams

- Initially 1 compounded by 12-13, years later several drops.
- Today: 2 teams: 2 x (2 senior dev + 1 jr dev) + 1 PO + 1 SM geographically dispersed in Tijuana and CDMX
- Platform developed (DECP + DEAM) consumed by > 90 global teams

Our own agile framework





Agile Frameworks

- CoE adopted SCRUM for teams
- DE adopted SAFe for agile enterprise escalation
- Level of agile framework adoption varied between all product teams from Classical Waterfall to Agile-Hybrid up to Agile (SCRUM, SCRUMBAN, KANBAN).

My teams

- Initially used a tailored SCRUM flavor.
- Later started to migrate to **SCRUMBAN**



SCRUM Tailoring

The Scrum Framework





SCRUM Tailoring

- **Sprint:** 2 weeks length (sometimes 1 week or 3 weeks)
- **Daily:** Tried < 20 min but it may vary.
 - *Focus changed from classical 3 questions to objective for today and issues happened the day before.*
- **Design Meetings** adhoc
- **Grooming** always happened to validate, clarify, add details to US and Tasks for next sprint (Risk identification, Dependencies to be Solved) => Next Sprint Planning very quick!
- **Dry Run** was adopted as a security measure before Sprint Review (Demo)
- **Retrospective** changed a lot =>
 - *From the Classical 3 questions to Sprint Content Review plus lesson learned and 6 hats perspective analysis.*
- **New Artifacts generation** - *Root Cause Analysis, Disaster Recovery Plan, Tech Support SLAs, Security Exceptions,, Tech Support Measurement Plan, etc.*



The Platform

- Platform consisted of:
 - Products to facilitate, extend and control the consumption of AWS (EKS, ECS, S3, RDS, Dynamo, Lambda, etc.)
 - Several hundred thousand USD of AWS Services consumed by the product teams.
 - Capabilities:
 - Orchestration
 - Automation request
 - Monitoring / Alarms
 - Pipeline build automation
 - CLI Improvement
 - Base Technology/Products
 - Python, Java, Javascript, Bash,
 - Jenkins, Git, Github
 - Jira, Confluence, MS Office, Sharepoint, MS Teams
 - Grafana, Watchdog, , Docker, Kubernetes, GithubAction (to be adopted)

The Challenges





Success may bring hurdles & make you feel miserable in your venture.

- Recognition of great work does not always provide the key to be supported.
 - Various requests over the years to replace or add new members went unheeded
 - Revenue and cost reduction were key metrics
 - Situations of continuous fatigue appeared after several months
- The benefits that brought to the teams the usage of our products were welcome.
 - When a successful product is used very frequently is accompanied by tech support requests.
 - => What is more important? support the users or continue the development of our products?
 - => Our velocity was impacted!
 - => A Tech Support Request cannot be estimated!!!
 - => We didn't know how many support request would be in a sprint => Planning / Estimation turned to be unreliable.



Profit and cost reduction are always sought by companies

- Usually profit and cost reduction are common goal within a lot of companies.
- A goal for our teams was to look for ways to reduce costs.
- We tried to build features and capacities to extend our products that allowed the measurement of services consumption without success.



What we did





Our solutions

- **For unplanned activities (including tech support):**
 - Reserved a percentage of the sprint for tech support.
 - Our capacity was impacted as well our velocity.
 - Start to replace SCRUM as purt base framework to Kanban that included considerations for not planned activities like tech support request.
 - Kanban was not adopted due to lack of several needed features of our Jira installation.
- **For cost reduction:**
 - We changed our products base technology:
 - We figured out to change ECS to EKS - This and other combination of products would allow teams to avoid the consumption of services if those are not used.
 - Migra
 - A Migration large sustained effort begun.
 - Explicit strategies to migrate many services for many products of the team happened.
 - Strategies were focused on risk reduction and quick feedback
 - Sprint included lots of SPIKES and tasks (not US).

Our mistakes





Issues and Consequences

- We didn't know that base technology migration would need a lot of time of effort.
 - => Took almost 1 year and several attempts at the beginning.
- We didn't know that inviting and organizing several dozen teams might take a lot of time and effort,
- We discovered that is important to setup expectation management as early as possible!!!
- We didn't anticipated situations of constant fatigues for several members of the teams.
- We didn't anticipated that functionality of teams products would be affected and that many various defects of their applications could be attributable to us.

Organizational Dynamics





An Organization is a living entity

- **At first our director allowed us to have great flexibility in terms of the activities, tasks and metrics for the day to day activities.**

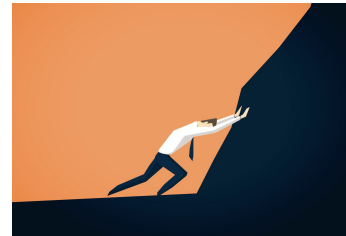
=> Although we needed to be ISO 9000 and ISO 27015 Compliant, some agile metrics and measurements were not requested or required by him. ***So we didn't capture those ones for several months.***

- **However after some time the director was replaced by a new one!!!**
- *New director brought new ideas, dynamics and objectives.*



An Organization is a living entity

- The new director were aware of the things we needed.
- He agreed to hire new staff members (3) but needed to justify with data that decision.
- **That data was all those measurements and metrics we didn't capture during several months!!!**
- Me and my manager were in big problems,



Our Response





What we did

- We tried to rebuild and regenerate the measurements and old data with the few existing data through extrapolation, tendencies and averages.
- The problem with that approach is that under pressure define and invent measures that never were obtained, becomes easily questionable, because it needs to be congruent from different perspectives at the same time.
- **The only solution =====> Say the truth!!!! The truth is always the way to go no matter the consequences, it is always better**
- Our professional image was seriously questioned.
- Several decisions were made and some required data were never obtained. Technical Deficit emerged.

Lessons Learned





Good and not so good experiences

Although we confronted and lived different situations, today the infrastructure team is highly recognized because of its innovative ideas and courageous approach to confront difficult situations with few resources.

The new team has grown with 3 new members located in India.

The team keeps its research and exploratory approach to provide new innovative solutions.

The team now knows that no matter if a high level role requests it or not, agile practices, measurement and correct and complete PM activities must be done.

No real defense or approach is provided by agile frameworks in situations of losing highly skilled members, quite the contrary in small teams the drop of a highly skilled member impacts several metrics and even the morality of a team.



Good and not so good experiences

- The lack of resources motivated us to generate approaches aimed to help us:
 - Better documentation for self service.
 - SLA's and Tech Support attention better defined.
 - Monitor and observability better suited for our own use:
 - Alarms integration to MS Teams
 - Chatbots
- The rapid growth and entrance of several new members at the same time in other time zones motivated us to:
 - Define strategies to manage part of the onboarding activities in other regions as well in our own region
 - Define better and more effective approaches for KT.
 - Discover the need for a managing counterpart in the remote location to facilitate the resolution of common problems that new members may find
 - Discover and figure out specific ours for syncing different team members.

Questions - Comments