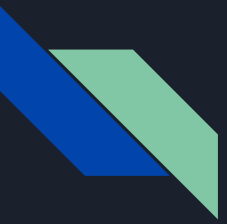
The Dawn of Developer Experience

@marioploria



Mario Loria

SRE, CartaX

marioloria.dev

Certified Kubernetes A/AD

CNCF Ambassador

K8s Office Hours Panel

KubeCon Program Committee

CloudNative Consultant and Mentor

What is DevEx?

What is DevEx?

*The more time engineers spend onboarding, bootstrapping, configuring, validating, coordinating, and asking for help, the less time they spend writing **code**.*

[..] developers are forced to borrow time from writing code and redirect it towards managing the issues associated with highly complex, multi-factor developer toolchains held together in places by duct tape and baling wire.

- "The Developer Experience Gap" by Stephen O'Grady, Redmonk

...end-to-end experience to **develop**, **deploy**, and **operate** services...

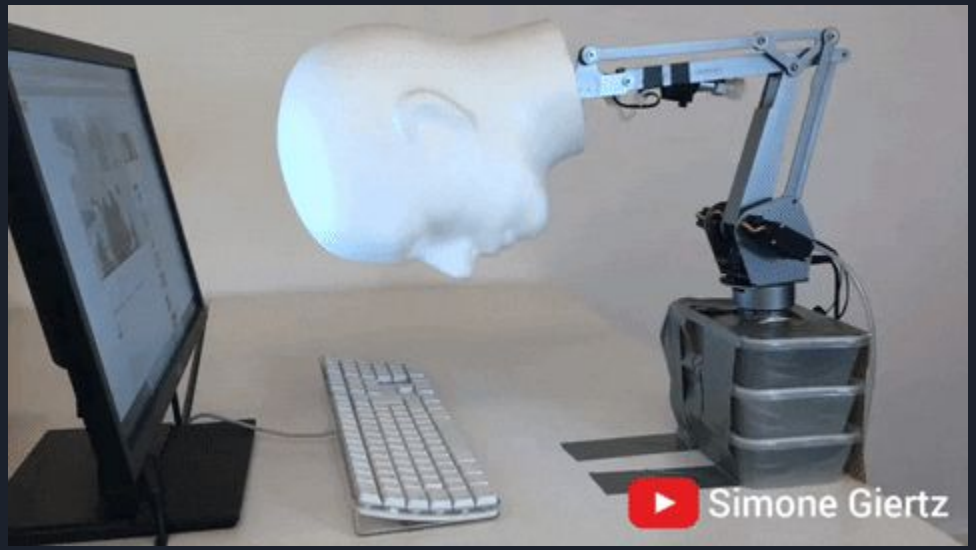
Sound Familiar?

Just some of the ways developers feel....

- Overwhelming Configuration
- Onboarding Pain
- Feeling of Isolation
- Lack of Best Practices
- Duplicated Efforts
- Who will help me? And when?
- How has this problem not been solved/documented/reported?

Causes of Poor DevEx?

- Tooling Overload
- Disjoined Communication
- Access Hurdles
- Maintenance Debt
- Lack of Platform Education
- Excessive Waiting (Support)
- Repeated Conversations
- Recurring Issues (unfixed from first encounter)
- Dependencies on other teams



Did I actually do anything today???

What about SRE/DevOps/Platform Teams?

Team learns about Developer Workflow ad-hoc from the developers themselves.

Primary focus is on cloud infrastructure, protocols, maintenance, security, compliance, and other systems-bound work.

SRE thus becomes a:

1. Constant Dependency
2. Support Group
3. Dumping Ground
4. Massively Delayed
5. Third-Party "External" Team

Their problem,
not **our** problem.

Not a **priority**

You're a **snowflake**

What about SRE/DevOps/Platform Teams?

But there's *Eventual Consistency*, right??

Developer-bound problems will eventually become solved by foundational enhancements, new tech or new tools.

Rarely the case:

- ➔ Second-hand understanding
- ➔ Thin viewpoint of issues
- ➔ Lack scale of importance
- ➔ Short-term, limited solutions
- ➔ Already underwater, split focus

Organizational Solutions?

Bigger SRE Team

- DevEx focused sub-group
- Close collab with rest of SRE
- See both sides easily
- Wavering focus, increased scope

Dedicated DevEx Team

- Exclusive focus on Devs
- Workflow understanding
- More native autonomy
- Difficult to hire for and execute

Alright, well here goes!

Start Now!

3 EASY PAYMENTS $$$
FULL REFUND!!! <<<
MUCH DELIGHT !!!

However, there is light at the end of the tunnel as *individual developers, teams and projects* start to think deeply about **design**, **style** and **taste** for tools to help us get into a flow focused on what we all love most: being creative and writing elegant code.

What all of them have in common is the desire to **delight**, the *desire* to reduce **friction** and the *desire* to leverage **automation** to solve problems and *turn duct tape into a Swiss clockwork* that works reliably below the surface.

- devxconf.org/manifesto

Every developer should feel fully **aware**, **responsible**, and **confident** in their ability to … services

Areas to Consider - *Make Life Easier*!

Technical Focus

- New Service Provisioning
- Onboarding Variance
- Heavy Infra Configs
- Environmental Sprawl
- Access Barriers

Team Dynamics

- Review Ad-Hoc Interactions
- Documentation Intuitiveness
- Engineering Workshops/Demos
- Communication/Touch Points
- Collocated SRE

Areas to Consider - *Be Innovative*!

- ★ Focus on high-value, high-impact, innovative projects
- ★ Create and utilize Working Groups to spawn increased feedback
- ★ Experimentation and Iteration are expected and encouraged!
- ★ Leverage one or more champion teams to help you greenfield and verify
- ★ Stay away from simple "fix it" or "make it better" projects
- ★ The Fine Line: *Supplemental*

Abstractions

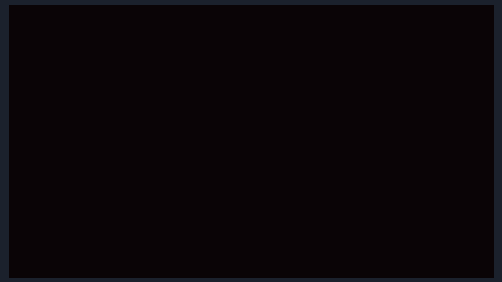
Crossplane.io

Build your own internal infrastructure abstractions on top of the CRDs Crossplane provides. Your custom APIs can include policy guardrails, hiding infrastructure complexity and making it safe for applications to consume.

Welcome to the Developer Interface!

Backstage.io

Project from Spotify focused on building developer portals for service owners!



- Centralized Service Catalog
- Uniform Introspection
- Discoverability and Accountability
- Service Management, not Cluster
- App Store for Infra
- Write your own Plugins

Welcome to the Developer Interface!

Code

Ship

Run

Measuring Progress

DORA Metrics

Model for DevOps related metrics that measure performance of Dev Teams.

1. Deployment Frequency
2. Lead Time for Changes
3. Change Failure Rate
4. Time to Restore Services

BONUS!

5. Microservice Bootstrap Time
6. Employee Net Promoter Score

Parting Principles

Infrastructure Components should be boring

Focus on Business logic and features

Functional ephemeral environments

Outward and over communication

Build Champions to your causes

Reduce the cognitive load and toil

Keep meetings short and productive

Self-Service, Self-Sufficient solutions

Service-Level Objectives

Educate Educate Educate!

Questions?

@marioploria
marioloria.dev

CartaX is Hiring! Remote US roles. PM me!