

ENSUCKLOPEDIA OF DevUps

A improvised glossary to help us for understanding of What DevOps should not be or become!!!



Jorge Luis Castro Toribio

DevOps Program Manager and Agile Coach

@ jorge.castro@pucp.pe

in [jorgeluiscastrotoribio](#)

This paper makes irreverent fun of some of our working experiences and shares lessons learned to avoid being a DevOps Zombie.

Please do not take it too seriously and be open to real life similarities.

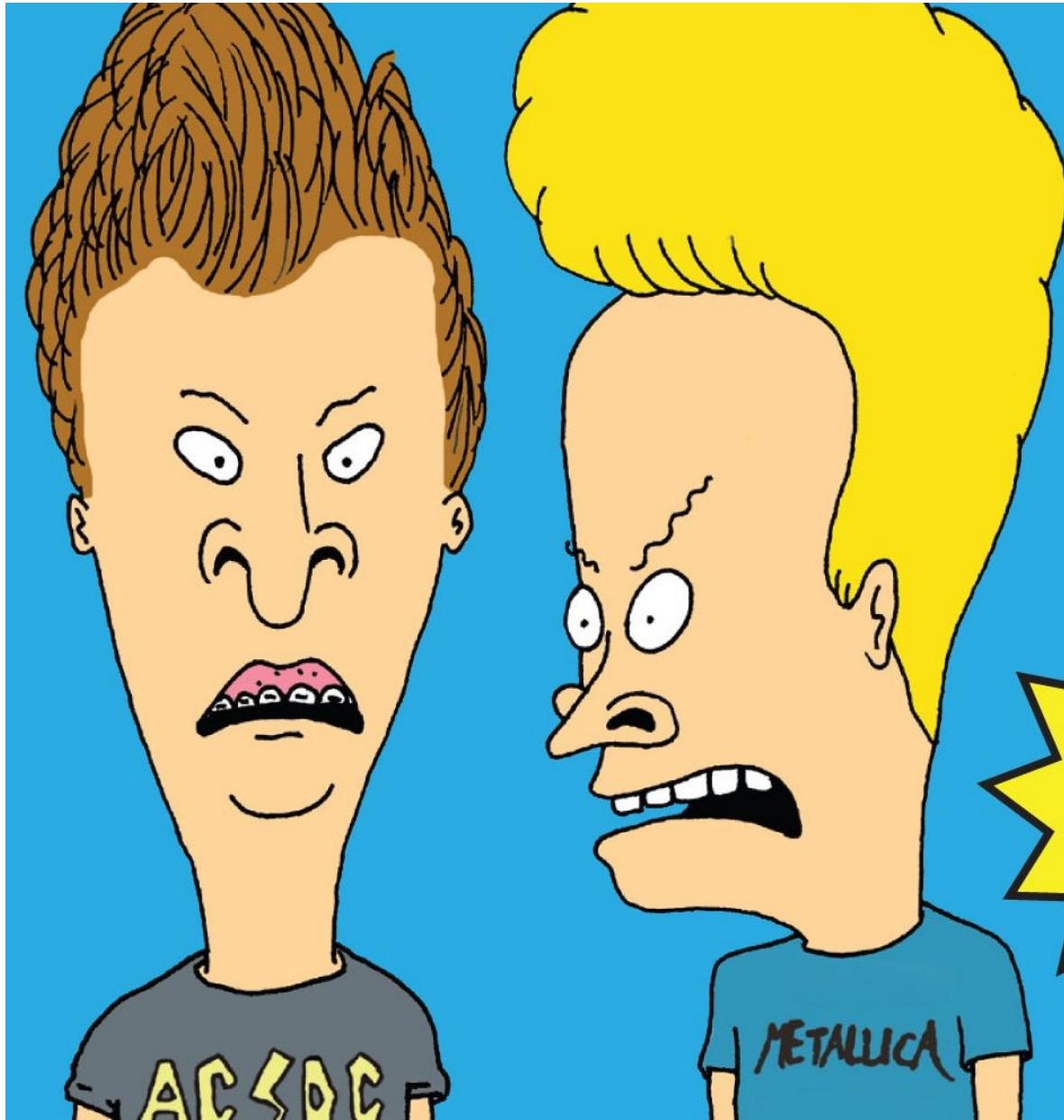


COME TO DevOps...
HUH, HUH, HUH.

**Probably most of us joining
IT help desk teams will be
familiar to this situation...**

Video

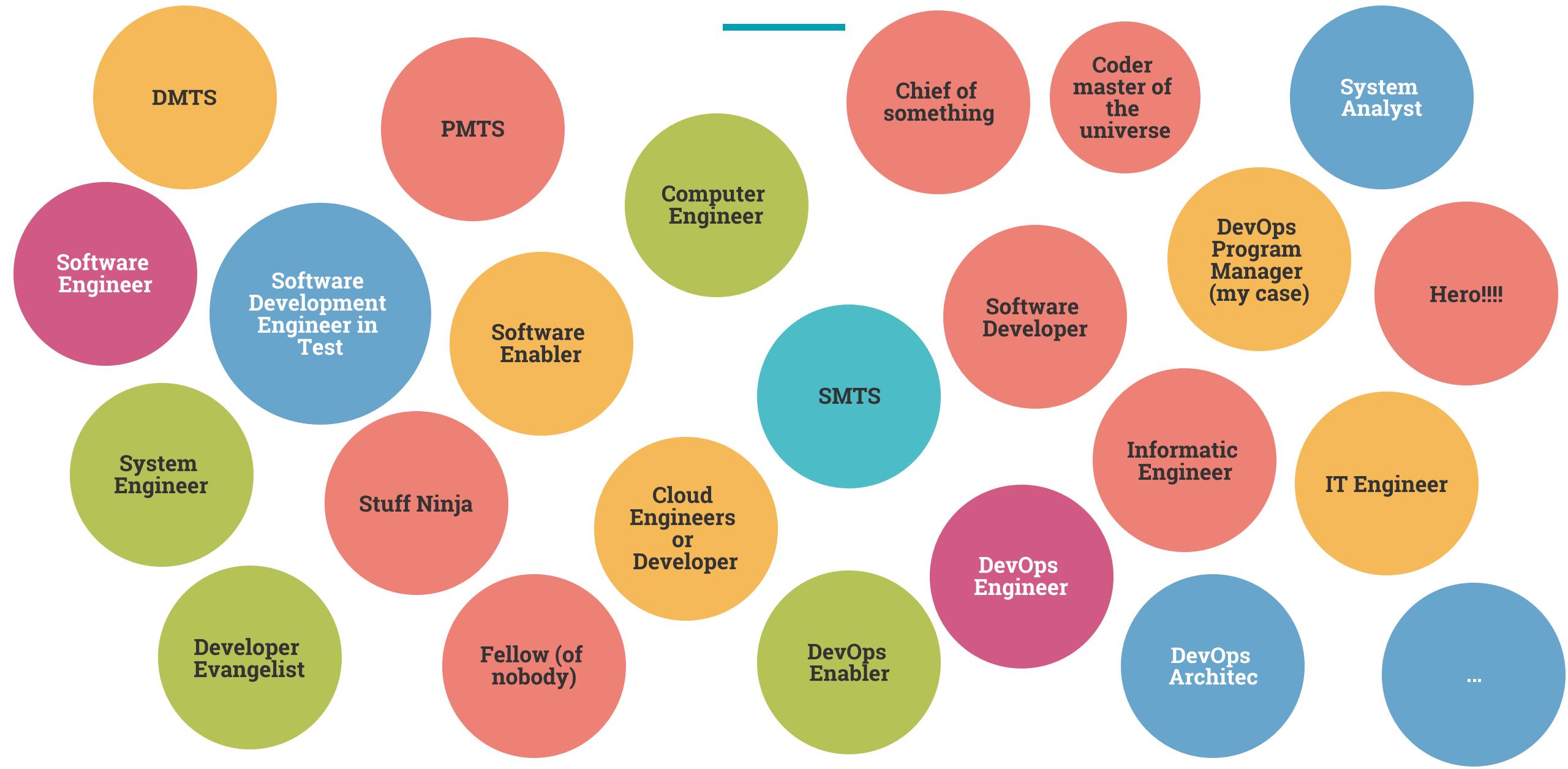
**Have you ever heard of
something like this?**



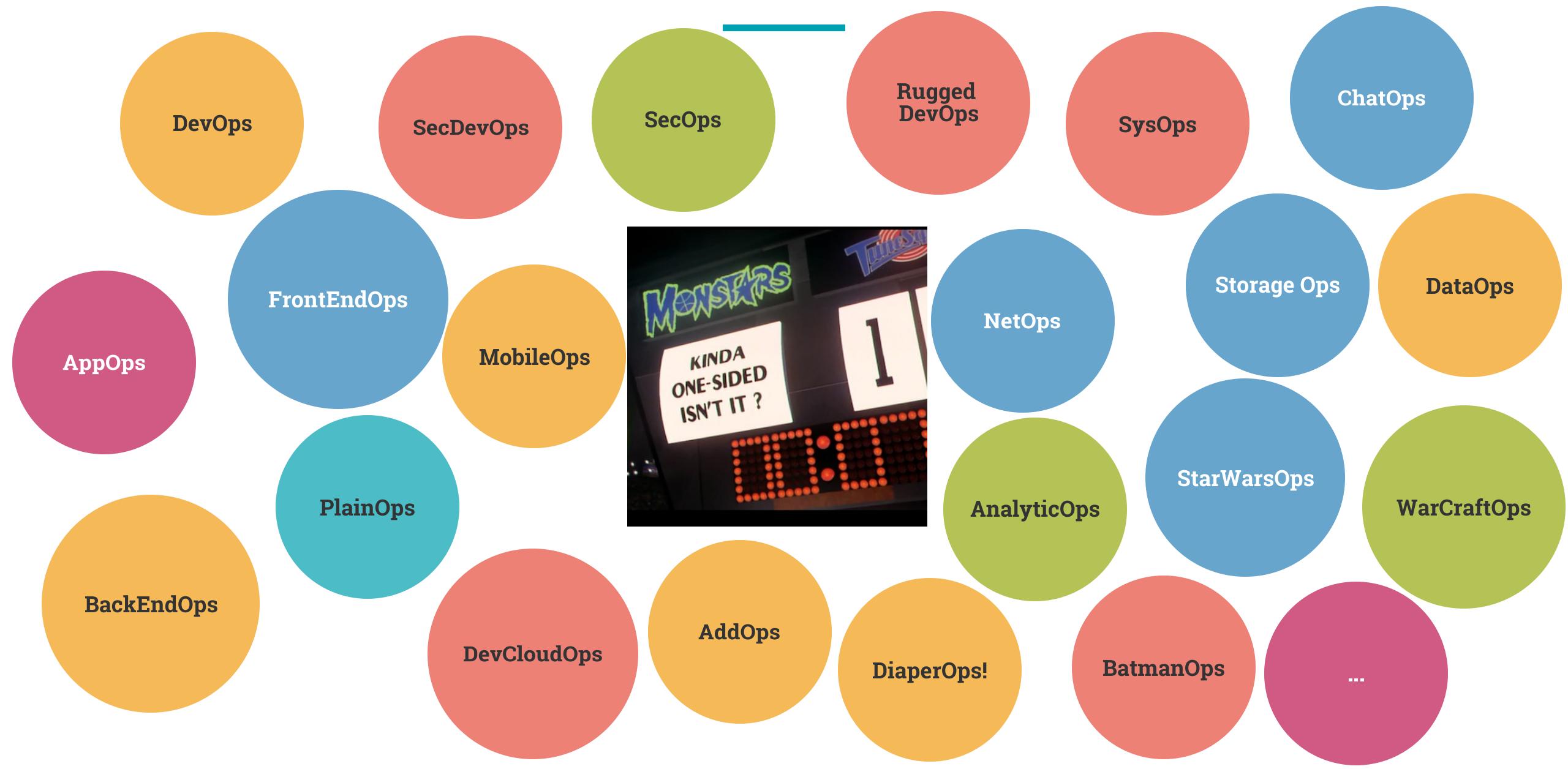
- "So you are a software engineer ... could you check my printer, it has a paper jam?"
- "Hey, you're a systems engineer, right? Can you help me upgrade my Windows?"
- Oh so you work as security engineer... can you help me to hack my ex's Facebook account?
- Cool, you are a mobile developer...so what is the best app to make memes?



Why Does this happen?

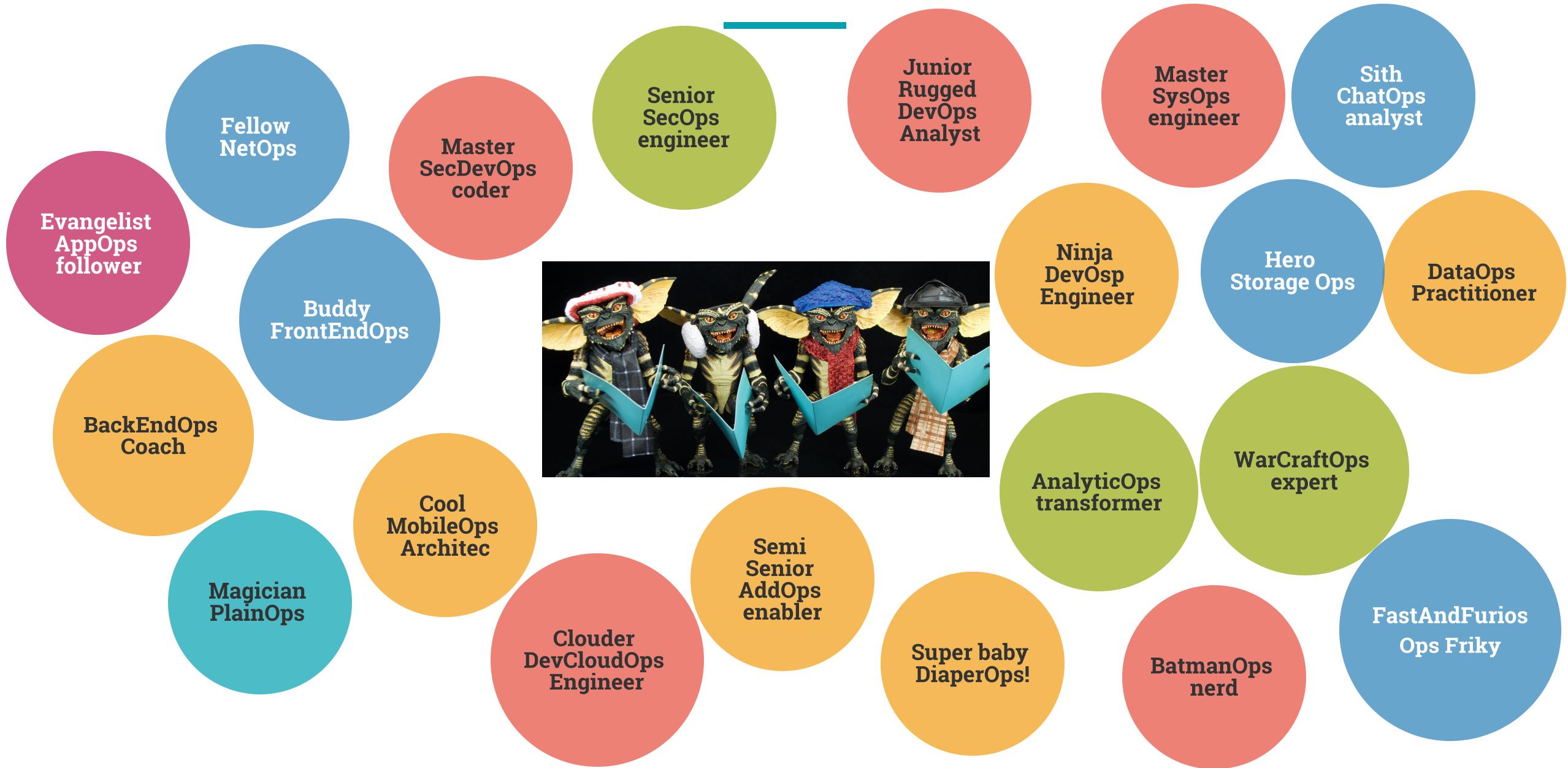


What about DevOps terms around there?



**So now imagine that you
add words like Ninja and
engineer to each term...**

This starts to be exponential like a pandemic!!!

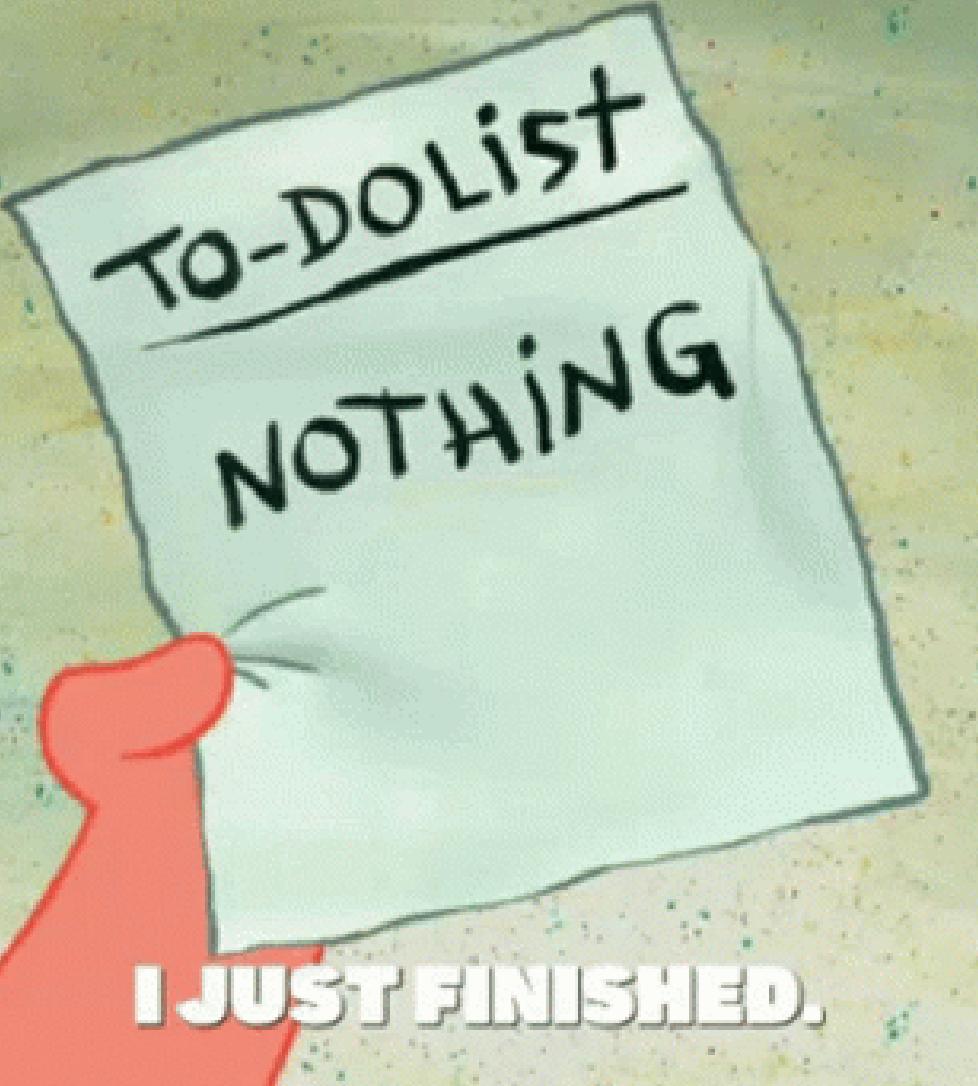




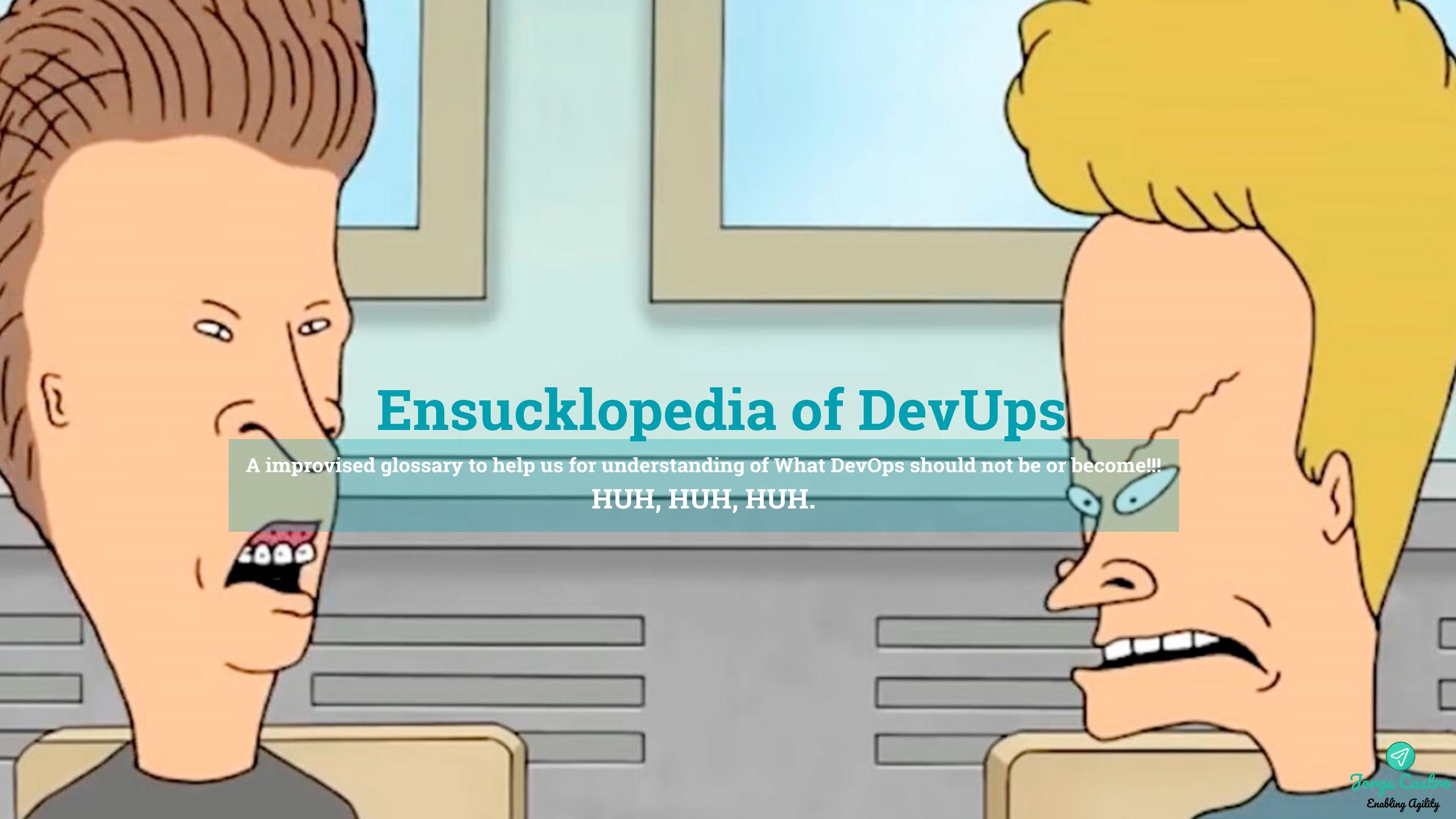
That stops there?...Well no, the Ops suffix makes things sounds cool so probably we will have more gremlis



It is ok to extend DevOps to other field but we should be careful about it because we could dilute the importance of it, lose our focus on software delivery, (which is the area that DevOps benefits most) and reproduce more gremlins in the spirit of EverythingOps



so besides
“complains”,
lets do stuff...



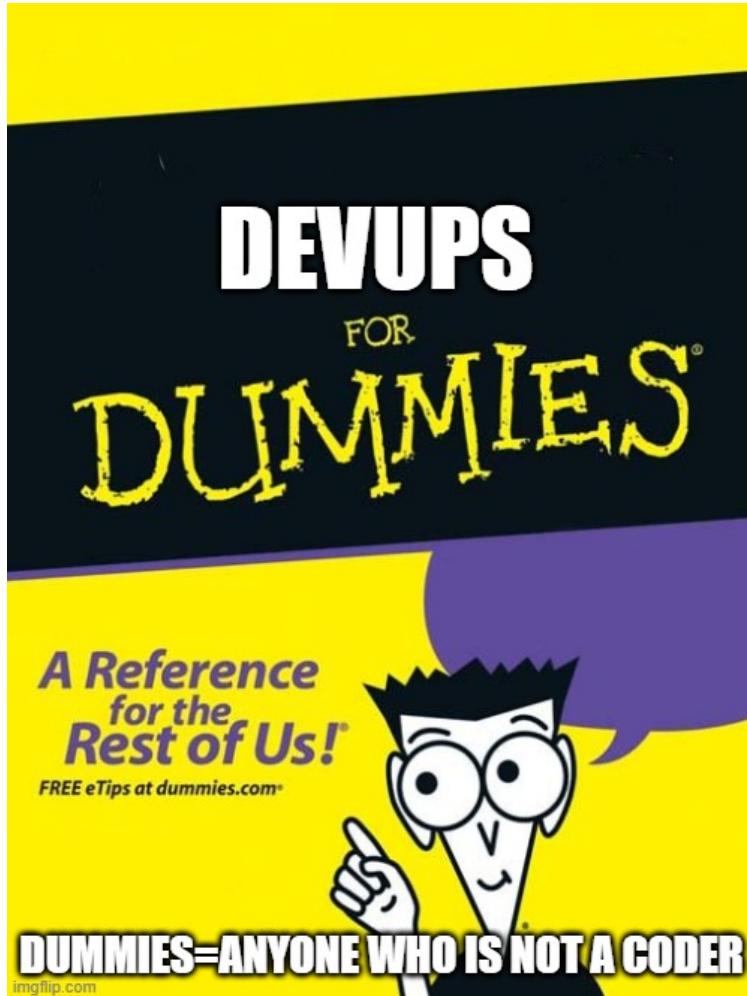
Ensucklopedia of DevUps

A improvised glossary to help us for understanding of What DevOps should not be or become!!!

HUH, HUH, HUH.



Content #1: DevUps from the scrotch



- 1 DevUps is the best way that managers have found out to push Dev and Ops to stop the crap to blame each other for crappy user experience and make them both responsible and do something about it
- 2 DevUps is about speed up stuff, due we have not clear about what we want, it helps us to automate crappy processes and make team "faster" in a non sense way but in an automation fashion
- 3 Flow: It is enabled by buying more non sense tooling and pressure teams to overwork and become unicorn developers with superpowers
- 4 There are only one kind of IT people, the ones who code, DevUps is for them, the rest are just jumping around to take pictures, present work to clients or managers and write papers like this

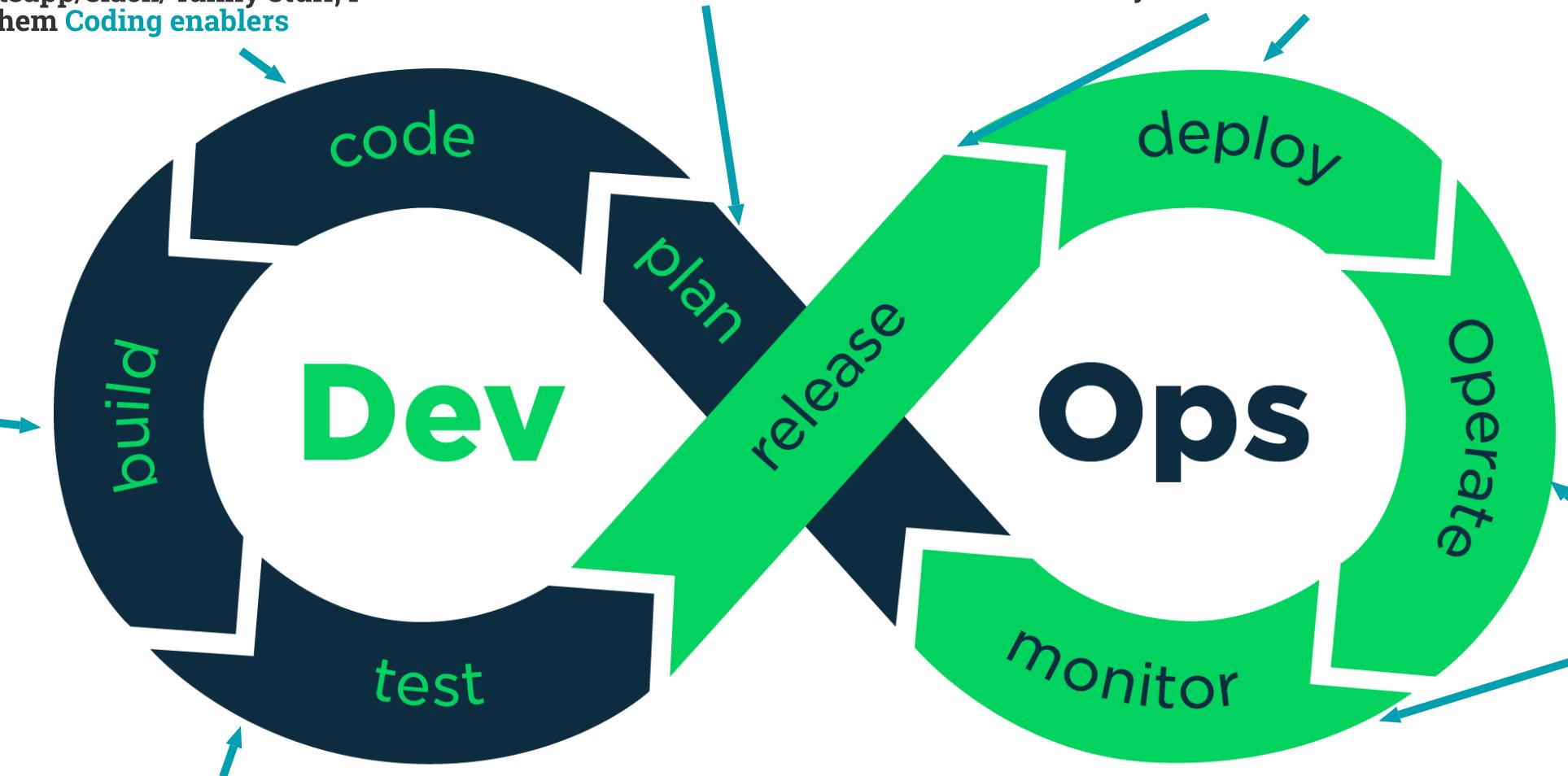


Before real coding starts, a bunch of real important and lean stuff happens like we must recruit new coders due to no body has the skills to do what we planned. DOTA, Netflix, FCB, funny squirrels videos on tube or whatsapp/slack/ funny stuff, I like call them **Coding enablers**

Plan stuff no matter if nobody has it clear, just plan it and be positive about the future, anyway remember that **Continuous ScrewUp** is also one of those continuous stuff

Here the team work as a real one for the second time ever and cross finger hoping the production does not collapse and the minimum of upper managers users get affected (they do not care about the other users). Here can occur **Continuous Fixing** (tons of hot bugs) or **Continuous RollBack** (tons of shitty manual stuff that enables more hot fixing)

Here we have to work as a team for the first time ever and pray together so our **crappy code** can build a stuff



We come to normal and we hate each other and we started to point fingers to whoever looks like a the responsible of such a shitty software quality, the developers become hardcore and blame testers, the tester become hardcore and blame developers, and the POC remains the same blaming the team as usual. All This creates **Continuous Retesting**

Here we have more of **Continuous Fixing** and less sleep, both turn more aggressive each time due upper managers are pissing bosses out





Law of Murphy for DevUps: if stuff can able go wrong, is mean is already wrong so we are losing money, our people is unmotivated and each time we lose more clients but you do not have and alert or feedback mechanism of it yet because who cares honestly, we care about coding what the feature says and marking our stories as done

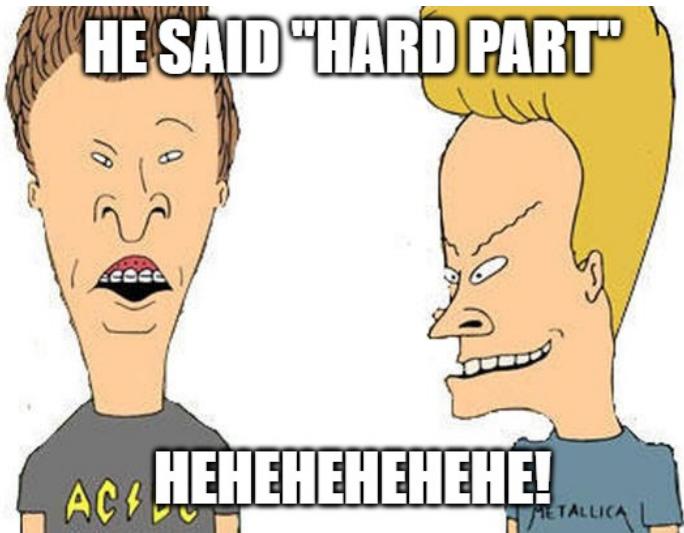
| 1 Os | | | | | | | | | | | | | | | | | | | | 2 En |
|---------------------|-------|--------------------------|------------------------|------------------------|----------------------------|-----------------------|---------------------|------------------------------|------------------------------|------------------------|-----------------------------------|-------------------------------|----------------------------|--------------------------------|------------------------------|----------------------------|-------------------------------|-------------------------------|-------------------------|-------|
| Gl GitLab | | | | | | | | | | | | | | | | | | | Sp Splunk | |
| 3 Fm | 4 En | Dt Datical | | | | | | | | | | | | | | | | | 10 Fm | |
| Gh GitHub | | | | | | | | | | | | | | | | | | | Sg Sumo Logic | |
| 11 Os | 12 En | Sv Subversion | Db DBMaestro | | | | | | | | | | | | | | | | 18 Os | |
| Cw ISPW | 20 En | Dp Delphix | Jn Jenkins | Cs Codeship | Fn FitNesse | Ju JUnit | Ka Karma | Su SoapUI | Ch Chef | Tf Terraform | XLd XebiaLabs XL Deploy | Ud UrbanCode Deploy | Ku Kubernetes | Cc CADD Director | Pr Plutora Release | Al Alibaba Cloud | Os OpenStack | Ps Prometheus | 36 Os | |
| 37 Pd | 38 Fm | At Artifactory | Rg Redgate | Ba Bamboo | Vs VSTS | Se Selenium | Jm JMeter | Ja Jasmine | Sl Sauce Labs | An Ansible | Ru Rudder | Oc Octopus Deploy | Go GoCD | Ms Mesos | Gke GKE | Om OpenMake | Cp AWS CodePipeline | Cy Cloud Foundry | 54 En | |
| 55 Pd | 56 Os | Nx Nexus | Fw Flyway | Tr Travis CI | Tc TeamCity | Ga Gatling | Tn TestNG | Tt Tricentis Tosca | Pe Perfecto | Pu Puppet | Pa Packer | Cd AWS CodeDeploy | Ec ElectricCloud | Ra Rancher | Aks AKS | Rk Rkt | Sp Spinnaker | Ir Iron.io | Mg Moogsoft | 72 Pd |
| 73 Fm | 74 En | Bb BitBucket | Pf Perforce | Cr Circle CI | Cb AWS CodeBuild | Cu Cucumber | Mc Mocha | Lo Locust.io | Mf Micro Focus UFT | Sa Salt | Ce CFEngine | Eb ElasticBox | Ca CA Automic | De Docker Enterprise | Ae AWS ECS | Cf Codefresh | Hm Helm | Aw Apache OpenWhisk | Ls Logstash | 90 Os |



Follow @xebialabs

| | | | | | | | | | | | | | | |
|-----------------------------------|---------------------|------------------------|------------------------|----------------------|-----------------------------------|----------------------------|----------------------------|-----------------------|------------------------|-----------------------------|------------------------------|------------------------|------------------------|------------------------------|
| 91 En | 92 Os | 93 Fm | 94 En | 95 En | 96 Fm | 97 Os | 98 Os | 99 Os | 100 En | 101 En | 102 En | 103 En | 104 Os | 105 Os |
| XLi XebiaLabs XL Impact | Ki Kibana | Nr New Relic | Dt Dynatrace | Dd Datadog | Ad AppDynamics | EI ElasticSearch | Ni Nagios | Zb Zabbix | Zn Zenoss | Cx Checkmarx SAST | Sg Signal Sciences | Bd BlackDuck | Sr SonarQube | Hv HashiCorp Vault |
| 106 En | 107 Pd | 108 Fm | 109 Fm | 110 Fm | 111 En | 112 En | 113 En | 114 Pd | 115 Pd | 116 Os | 117 Fm | 118 En | 119 En | 120 En |
| Sw ServiceNow | Jr Jira | Tl Trello | Sk Slack | St Stride | Cn CollabNet VersionOne | Ry Remedy | Ac Agile Central | Og OpsGenie | Pd Pagerduty | Sn Snort | Tw Tripwire | Ck CyberArk | Vc Veracode | Ff Fortify SCA |

Content #2: DevUps tool\$



- 1 A **hard part** in DevUps is make money from tools after you are reinvent it, add it a big great maximum and super feature and make your customers to buy the **freemium** version and its consultancy service despite there is an open source option that does the same
- 2 Open Source DevOps tools are only used for POC, spikes or demos, after that, if you want to "scale", you need to buy too\$
- 3 Advantage of DevOps tools: They can **make you look smarter and cool** despite of you keep the coders **doing the same stuff with the same issues** in a fashion way.



Content #3: DevUps Engineer
You hired DevUps engineers imagining on this but...

A photograph of a man dressed as Clark Kent/Superman. He is wearing a red suit over a blue t-shirt with a yellow 'S' emblem. He has dark hair, wears black-rimmed glasses, and has a confused or weary expression. His right arm is raised with a fist, and his left arm is bent with a flexed bicep. The background is a blurred outdoor setting with some red and white structures.

...the reality is a little different from your expectation



And this happens because of things like this...

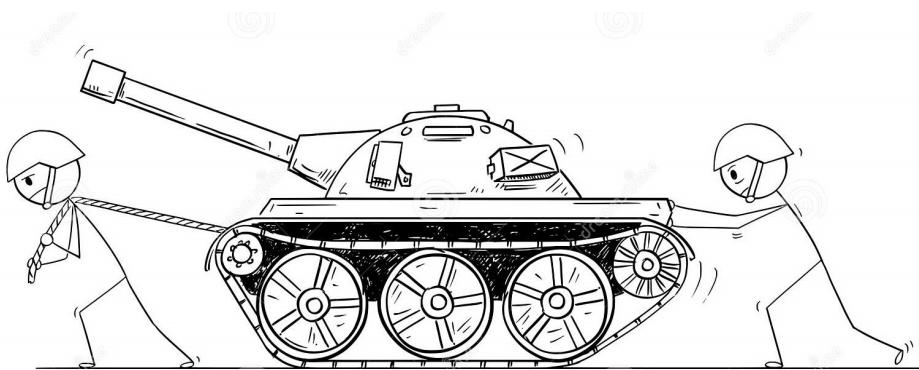
EXPERIENCE

DevOps Ninja job description

Technical:

- Minimum of 13 years' experience in IT Infrastructure engineering including at least 2 years' supporting cloud based infrastructure and working as a programmer for web application and mobile projects.
 - Strong experience working with CI/CD processes and tooling such as Git, Jenkins, Bamboo, GitLab, TeamCity, CircleCI, SolanoCI, AWS CodePipeline
 - Strong experience with configuration management tools such as Chef, Ansible, SALT Stack
 - Strong Experience with orchestration CloudFormation, Azure Resource Manager, Terraform, Ansible
 - Strong experience with HashiCorp Suite (Terraform, Packer, Nomad, Consul, Vault)
 - Strong understanding of complex enterprise network and security in cloud environments (Direct Connect, VLANS, Peering, VPN etc)
 - Strong scripting abilities (Python, Powershell and/or Bash)
 - Strong coding abilities (Java, .NET, COBOL, FORTRAN, Pascal, AS400)
 - Strong knowledge of Linux systems administration (Ubuntu, CoreOS, RHEL, Alpine)
 - Strong knowledge of Windows Server administration (2003, 2008 R2, 2012 R2, 2016 Nano)
 - Strong knowledge of containerization using Docker compose services
 - Strong knowledge of containerization platforms on Docker Swarm, Kubernetes, AWS ECS
 - Strong knowledge of serverless platforms such as Lambda, Kinesis, Batch
 - Strong knowledge of telecom, robotic, Manufacturing, mining, Aerospace business
- Strong knowledge of Testing, Project Management, Agile, SAFE, UX, accounting, lean, PMI, DevUp and so forth cool stuff

Content #4: DevUps and Agile

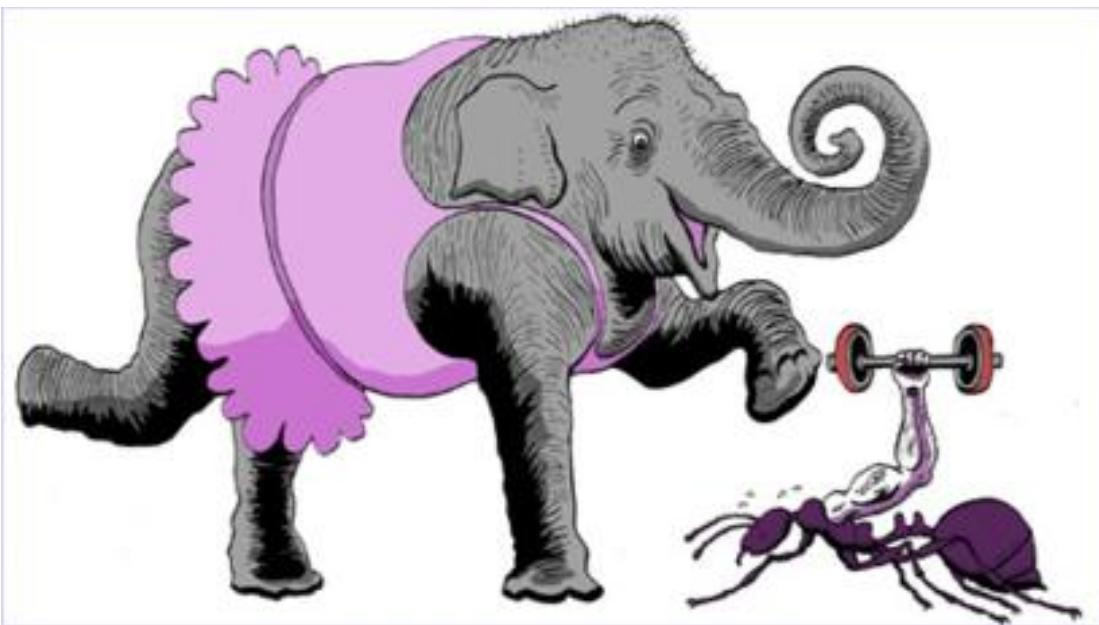


- 1 Agile DevOps can not able afford have bug in code. Is why developers test their own code and **not need QA**.
- 2 DevOps is not true agile if is not sold with an expensive agile transformation consultancy service.
- 3 Agile needs DevOps in the same way that **developers need Stack Overflow** to be really productive
- 4 In theory, DevOps and Agile are just as important, in practice they are not

Advantage of Agile is allowing DevOps gets nowhere much faster without a purpose in the middle



Content #5: Scamled DevUps



- 1 Support from the Top while let the self-organized teams do the work and trains/tribes/etc. takes the credit.
- 2 Scaling DevOps is the art of fail at scale continuously (**Continuous ScrewUp**)
- 3 Trust on your senior DevOps engineers to lead scaling DevOps, they make junior developers be in charge of scale up while they are in charge to scale down.

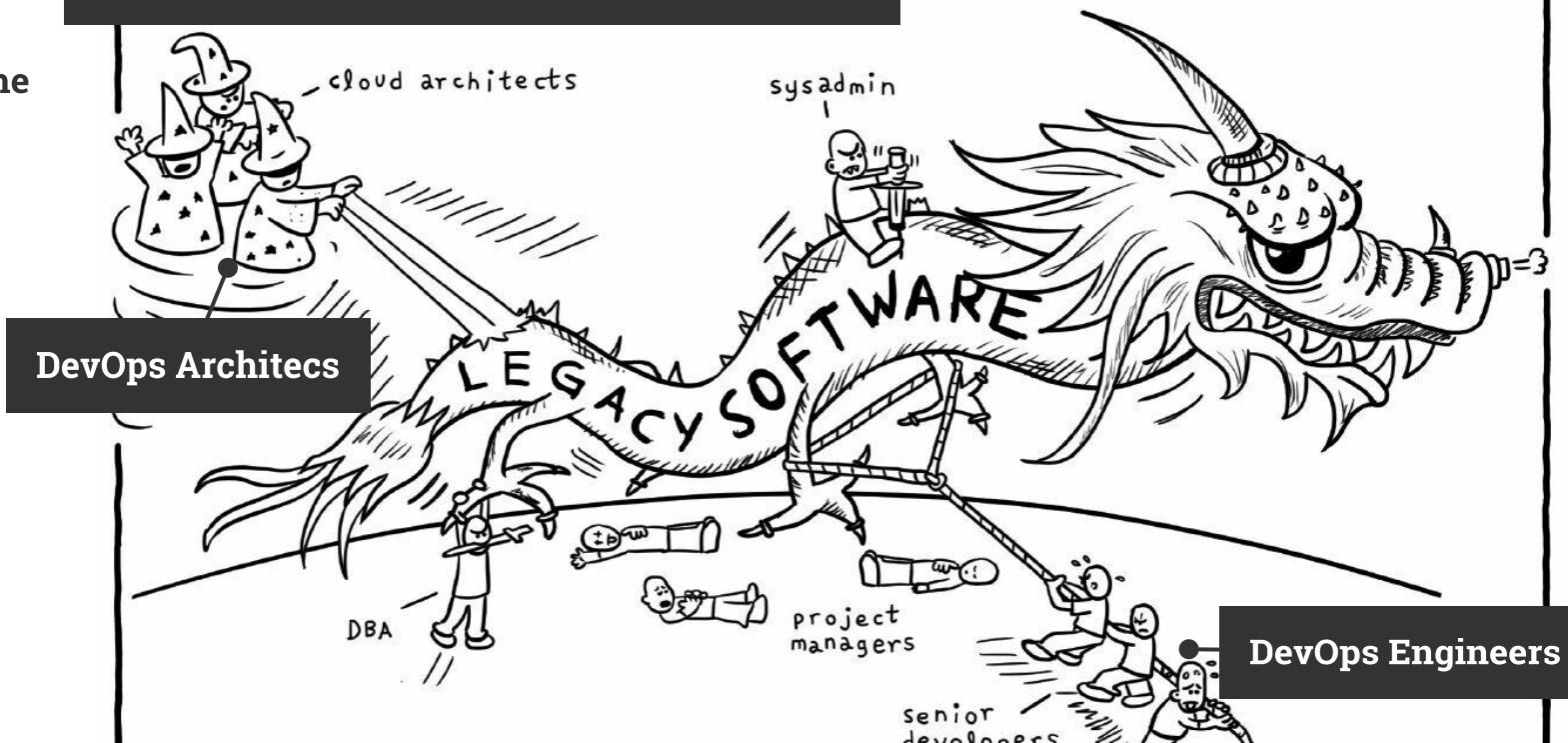
Content #6: DevUps Prankciples and practices



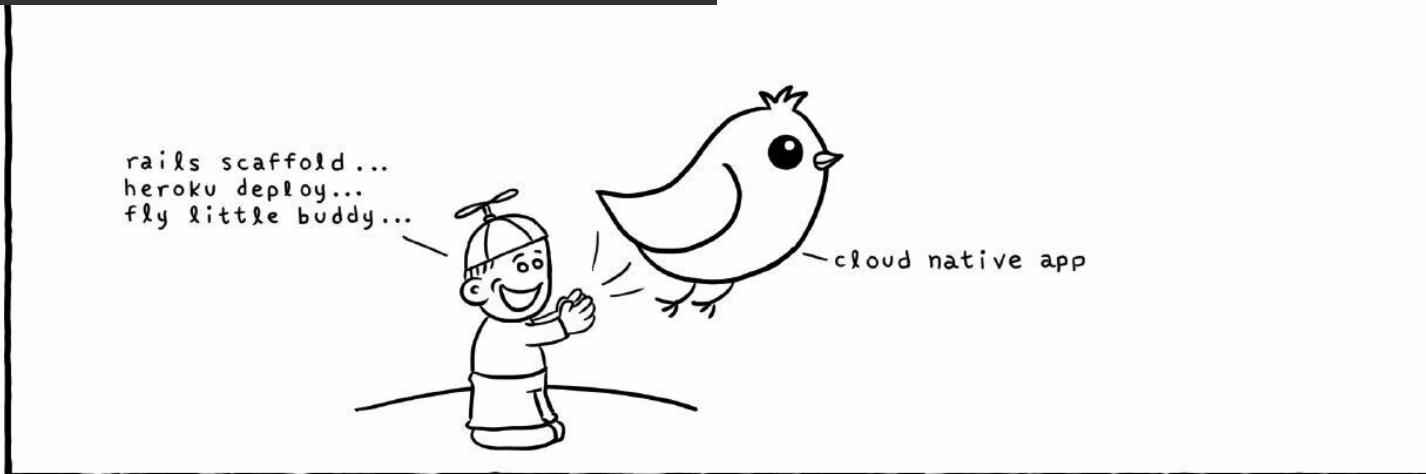
- 1 **Flow: Polite way to foster pushing teams to work faster**
Practices: Continuous Desintegration, Continuous DEBTlivery.
- 2 **Feedback: Responding to the needs of CEO and her closer friends. Foster “Fail Fast” to robust low performance reports**
Practices: Continuous ScrewUp, Continuous RollBack, Continuous ReTesting
- 3 **Continuous experimentation and learning: Try new stuff despite of there are already verified ways to do it and your company is lack of budget, make it hard, complicated so anyone wont learn anything from it.**

Enterprise Journey to Digital Transformation

Content #7: DevUps and the digital trastornation

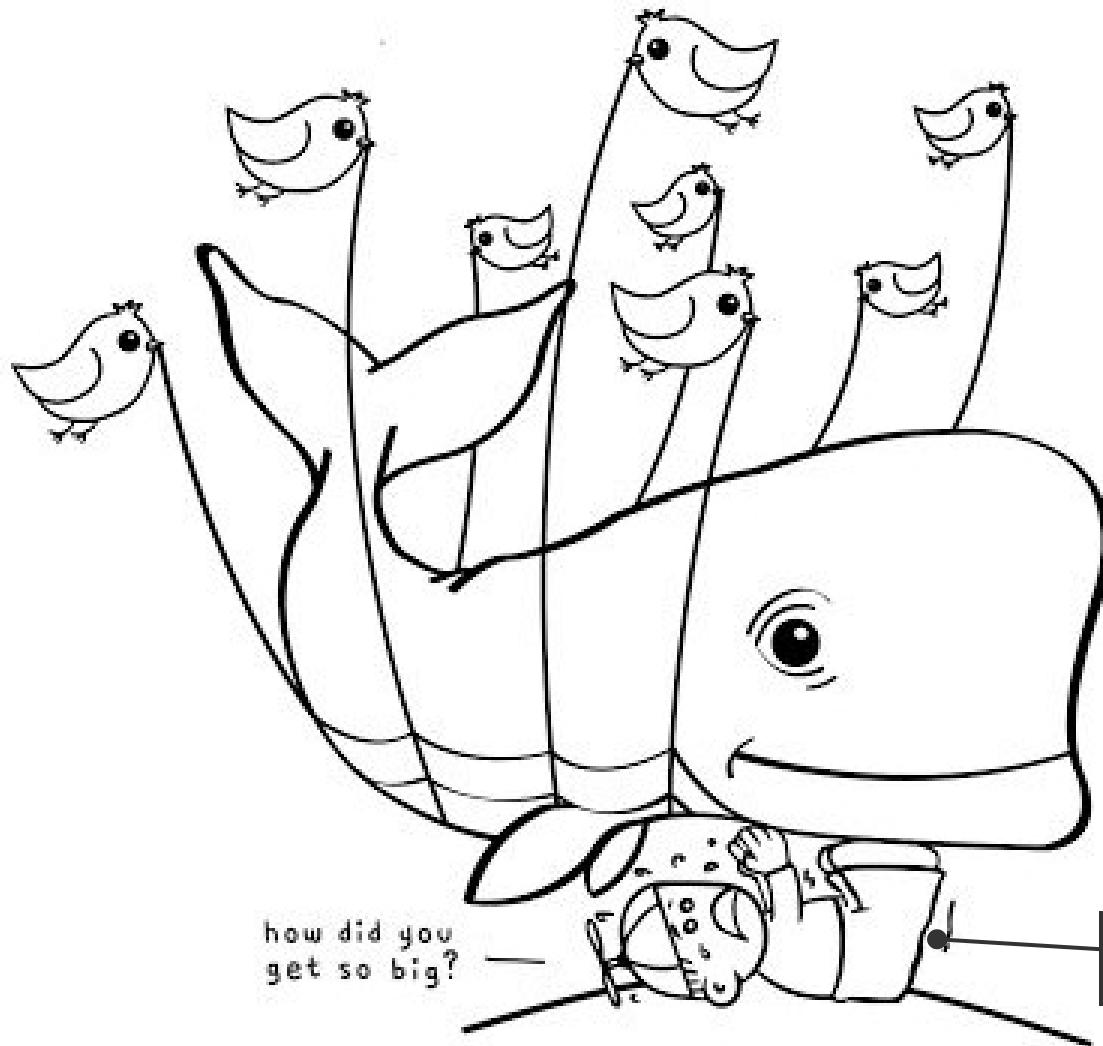


Startups journey to digital transformation



Daniel Stori {turnoff.us}

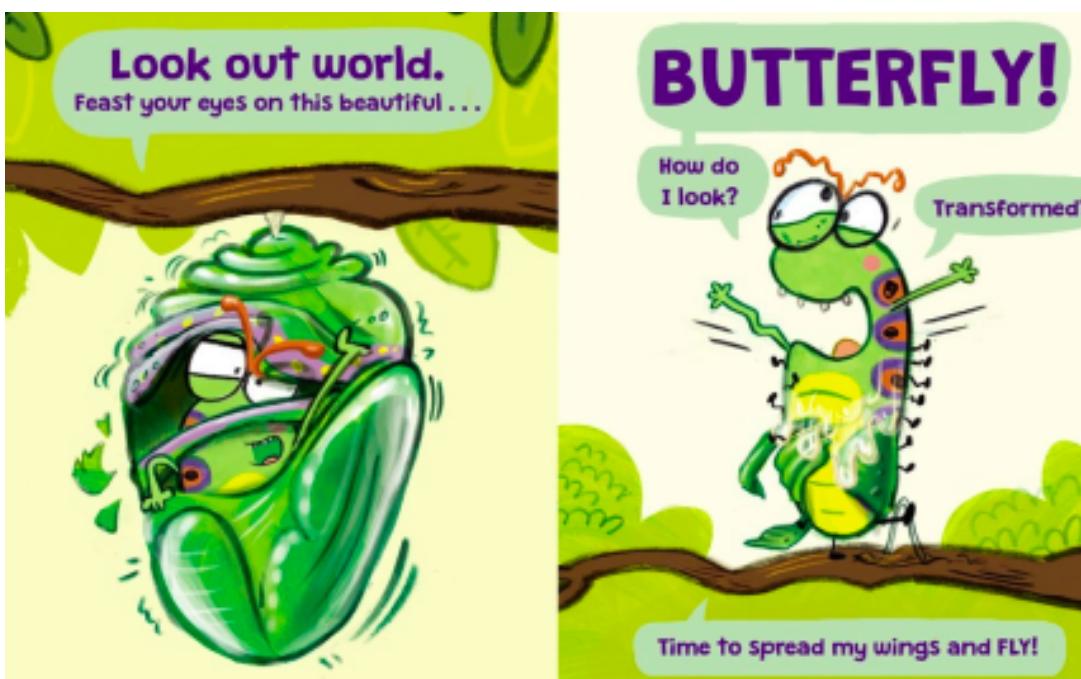
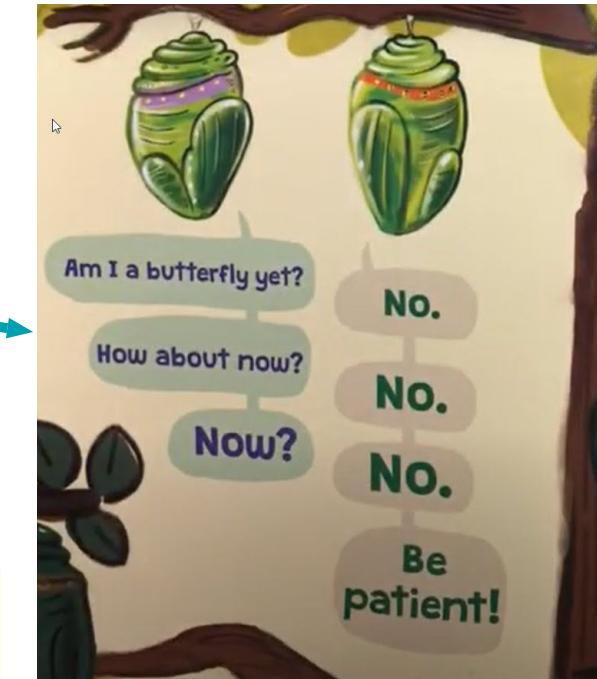
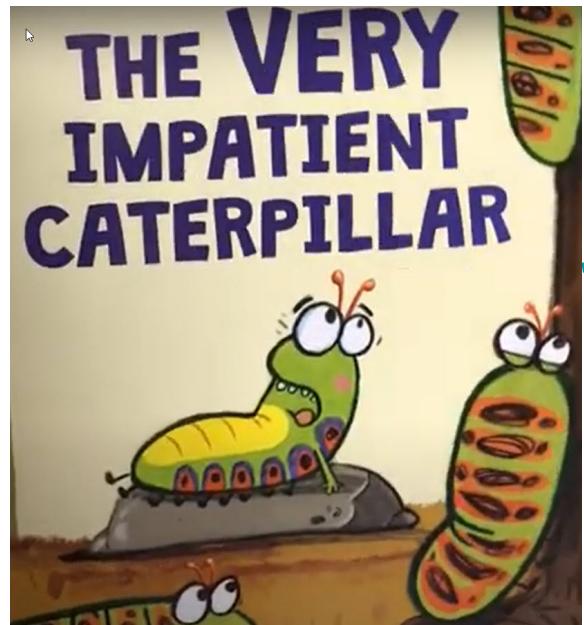
Startups journey to the enterprise digital transformation



Agile, Digital, DevOps, Republican,
Vegan, **stuff transformation** (Yes,
I know many of you don't like the
word transformation but lets keep
it simple) demands a clear purpose,
great teams and patience...please do
not be like the impatient caterpillar

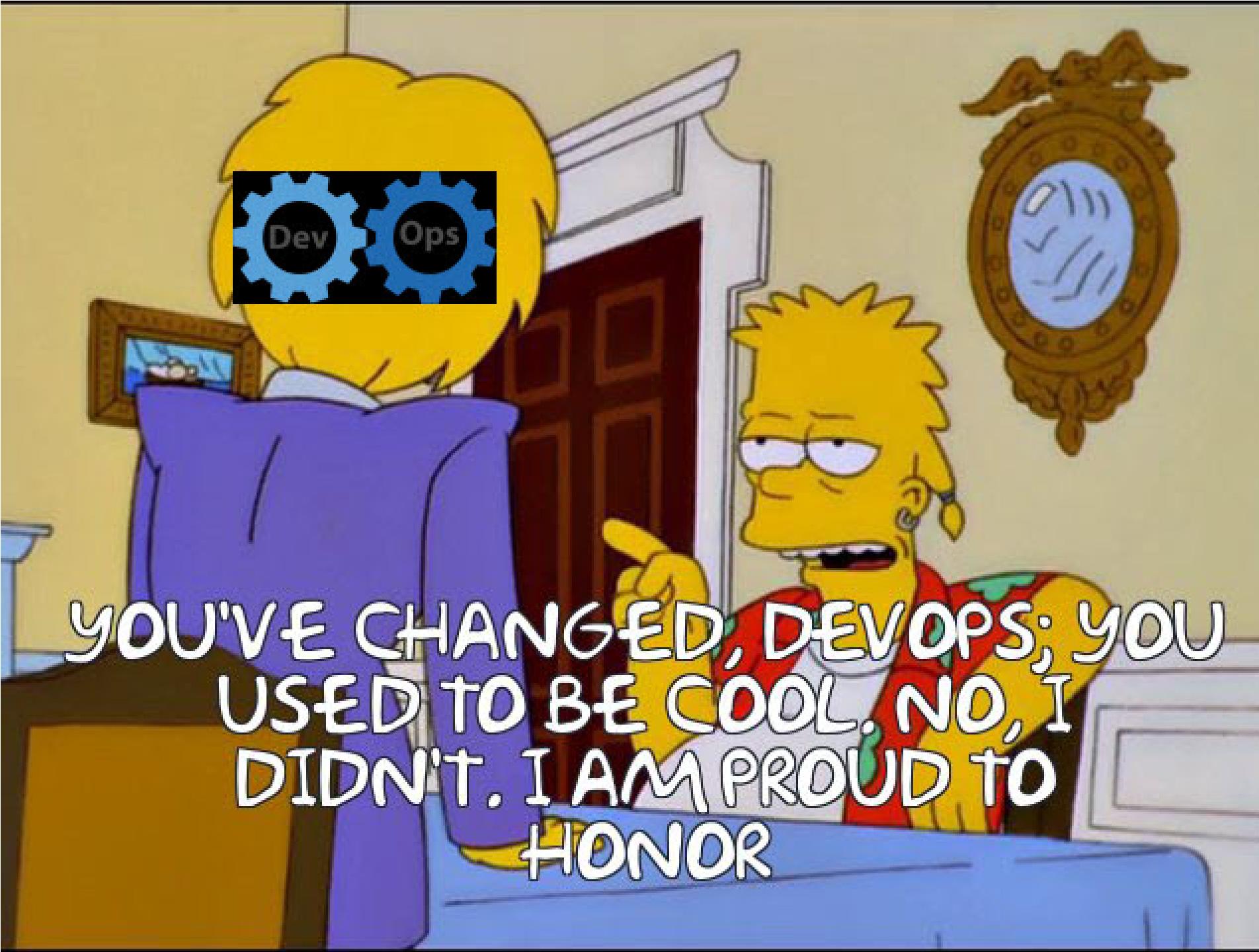


Once upon a time..





**So now I am
more confused
than before...**



We need your help



- 1 Adopting product-based operating models (please operate thinking on products)**
- 2 Measuring and gamifying engineering productivity, digital adoption, continuous improvement and collaboration.**
- 3 Support and actively participate in open source communities**
- 4 Make DevOps practices attractive as a hobby, gamify it, it will drive your company to have great teams and a new avenue to revenue**





Thank you! = ¡Muchas gracias!