DevSecOps
Security in DevOps

Aarno Aukia, CTO @ VSHN - The DevOps Company

4.6.2019
Swiss Association for Quality

Agenda

- About Aarno & VSHN.ch
- From Dev to DevOps to DevSecOps
- DevOps/AppSec/DevSecOps/SecOps?
- Automating Operations to include security
  - Build
  - Test
  - Deployment
  - Ops
    - Software containers & container orchestration: Docker & Kubernetes
    - Cloud Native Computing
- IT Governance improvements

About Aarno & VSHN.ch

@aarnoaukia http://about.me/aarno aarno.aukia@vshn.ch

ETH → Google → Atrila → VSHN

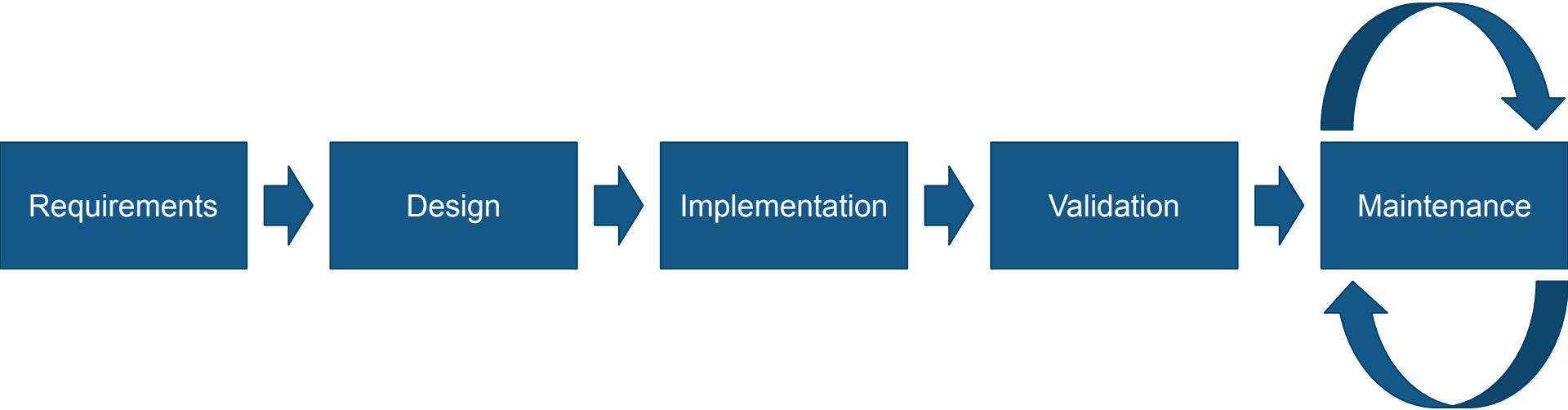
VSHN - The DevOps Company

Since 2014, currently 37 VSHNeers in Zürich, Switzerland

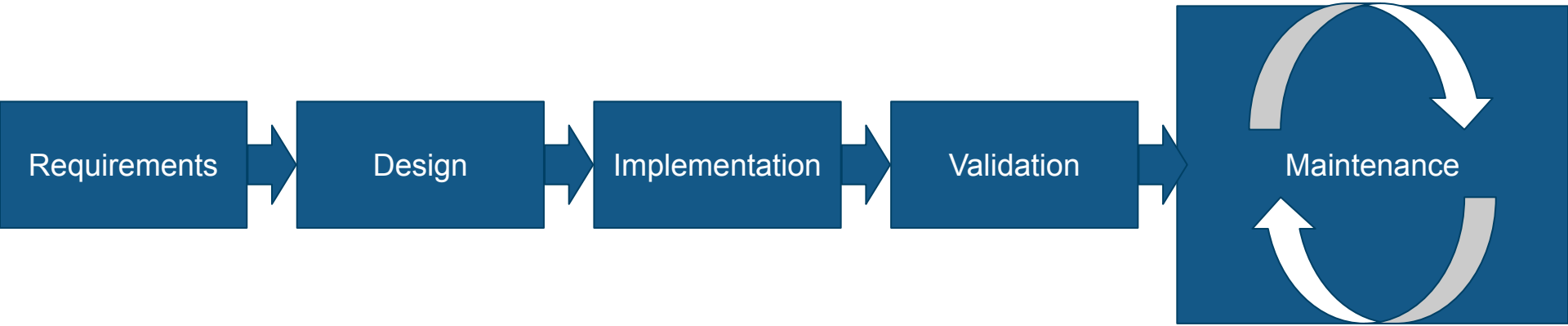
Helping Developers run applications on any infrastructure making both visitors happy with stability and developers happy with agility

Software Project Management

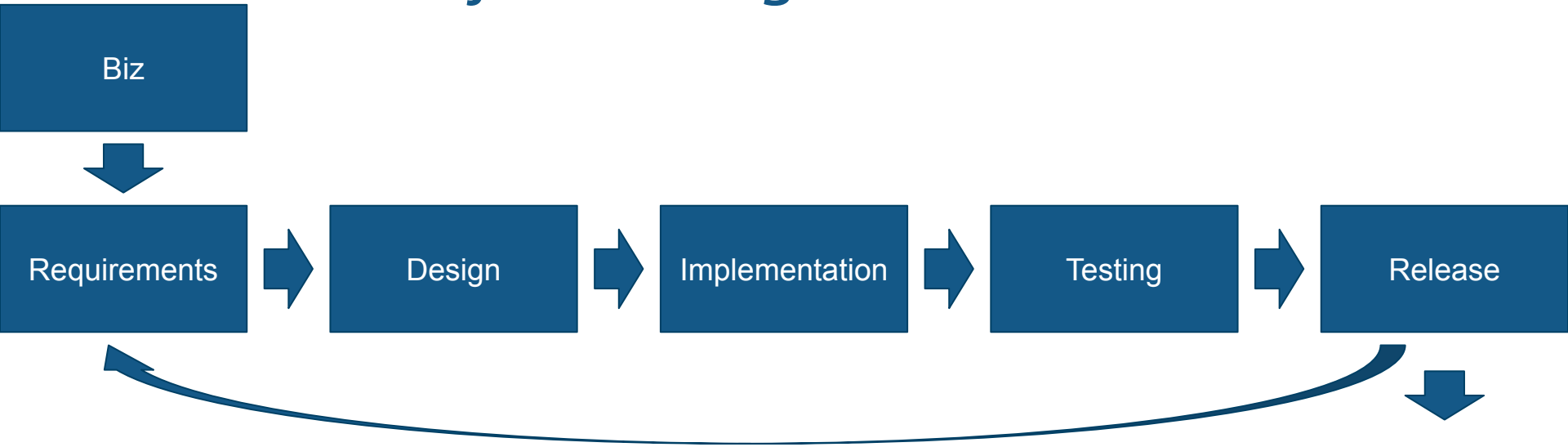
Software Project Management



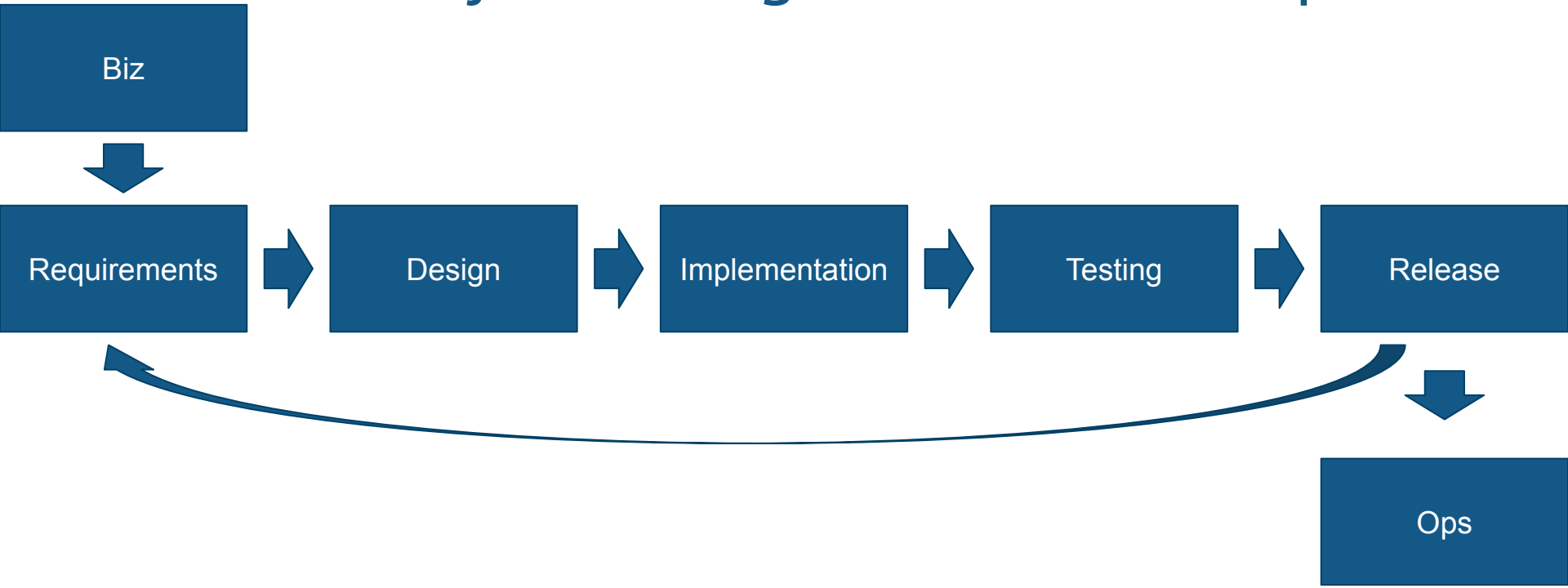
Software Project Management



Software Project Management

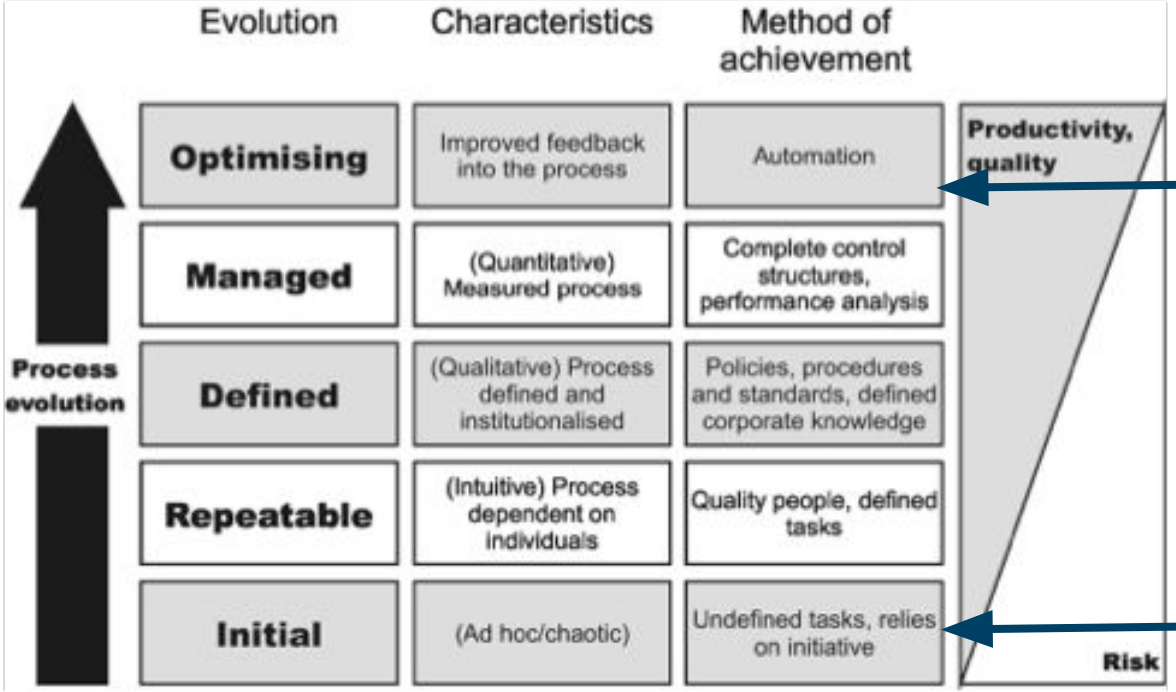


Software Project Management: Dev vs. Ops



OPS = Firefighting-as-a-Service ?

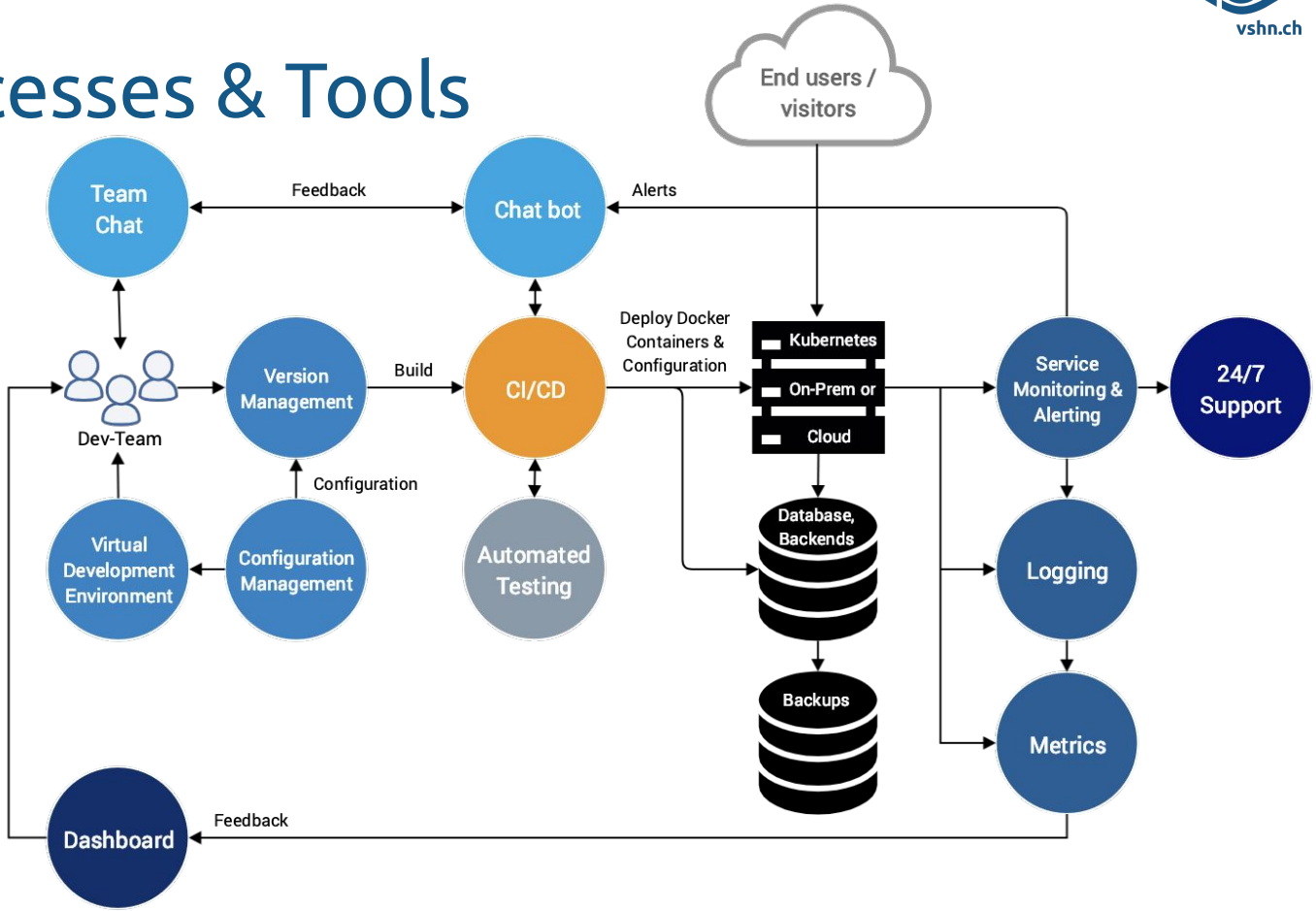
Capability Maturity Model Integration (CMMI)



How to get to this level?

Operations 2014

DevOps:
People, Processes & Tools

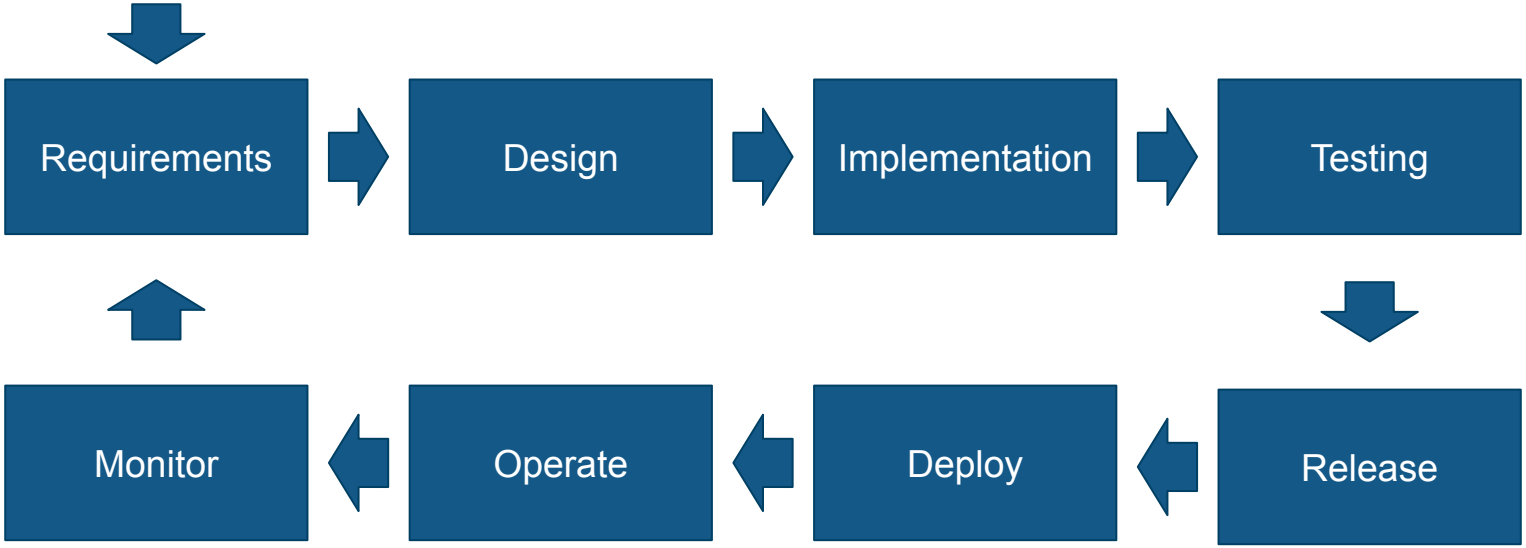


DevOps: People, Processes & Tools

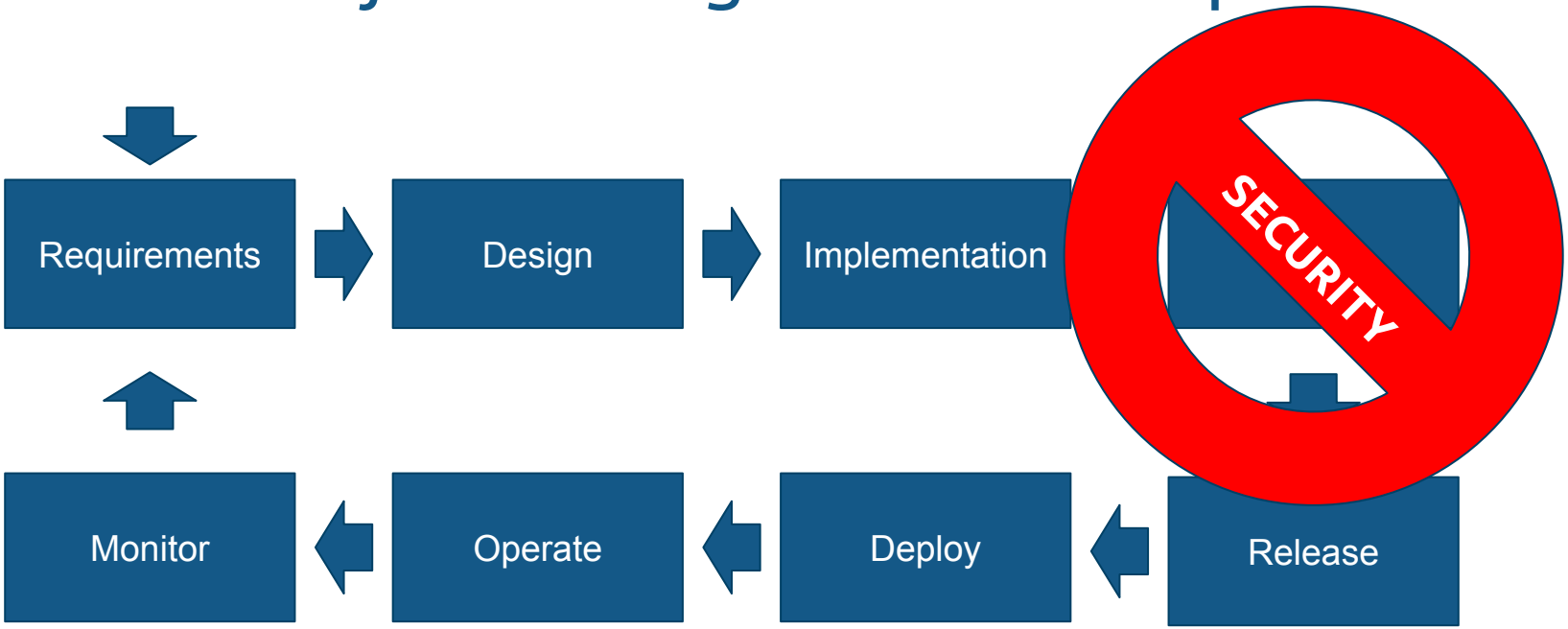
Collaboration between software developers and operations:

- Teamwork
- Continuous improvement
- Efficient and lean
- Agile: being able to react to new requirements
- Automate as much as possible (“Infrastructure as code”)

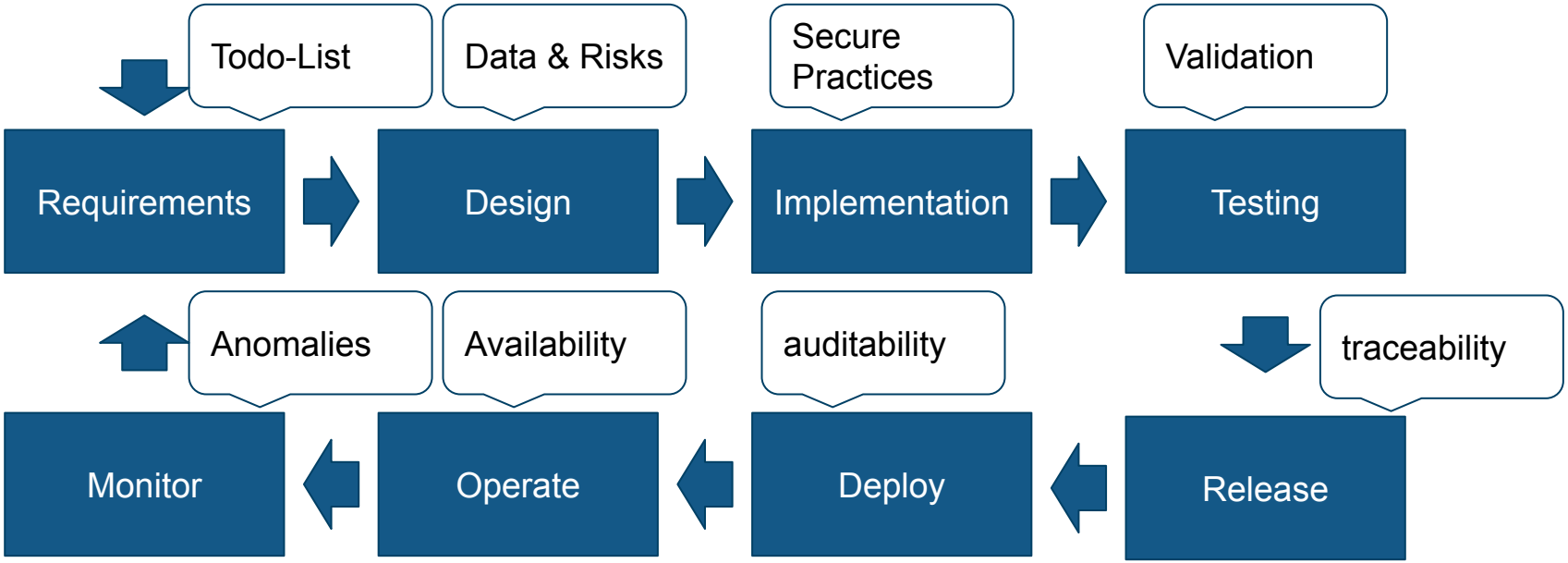
Software Project Management: DevOps



Software Project Management: DevOps



Software Project Management: DevSecOps



Areas of security improvement

- Developer education, requirements engineering, design review -> AppSec
- Software Build/Deployment/Operations -> DevSecOps
- Incident detection & management -> SecOps

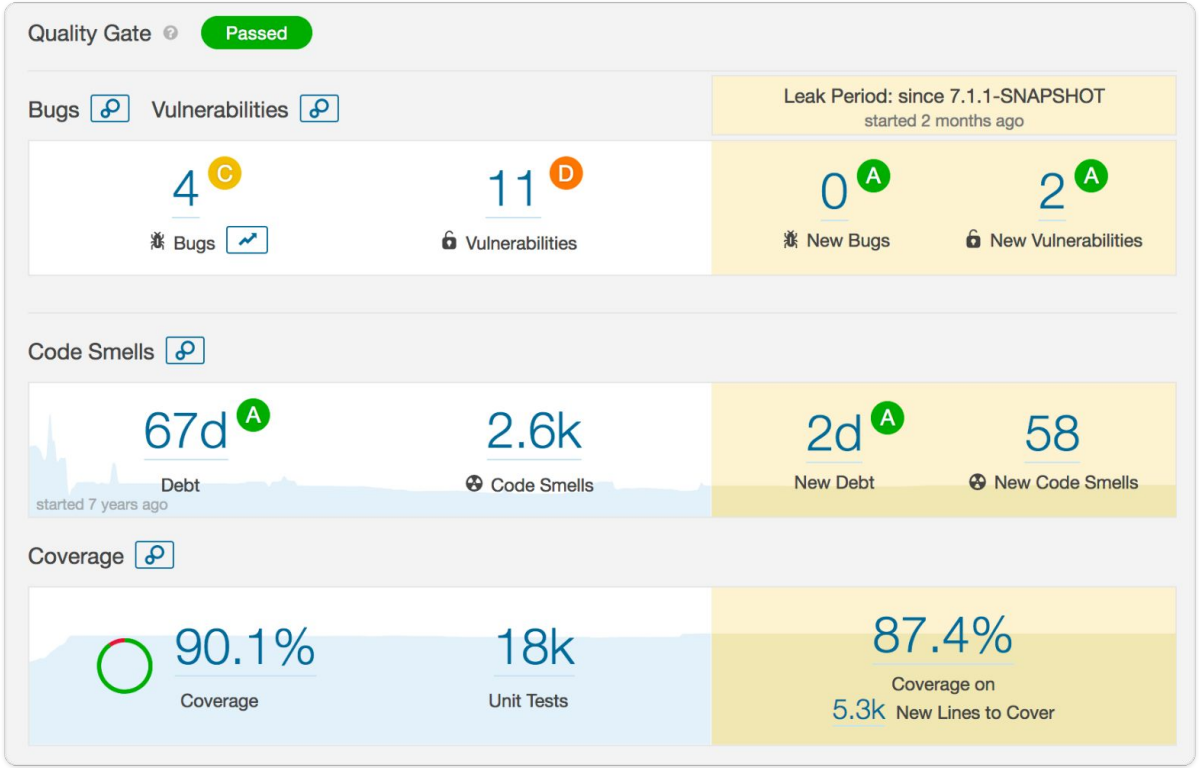
DevSecOps principles

1. Increase Trust And Transparency Between Dev, Sec, And Ops
2. Understand The Probability And Impact Of Specific Risks
3. Discard Detailed Security Road Maps In Favor Of Incremental Improvements
4. Use The Continuous Delivery Pipeline To Incrementally Improve Security Practices
5. Standardize Third-Party Software And Then Keep Current
6. Govern With Automated Audit Trails
7. Test Preparedness With Security Games

Build

- static code analysis automatically for each commit
- Dependency Management
- (base) container image scanning

Code analysis: sonarqube



Dependency updates: https://dependabot.com

arska / python-helloworld

Unwatch 1 | Star 0 | Fork 2

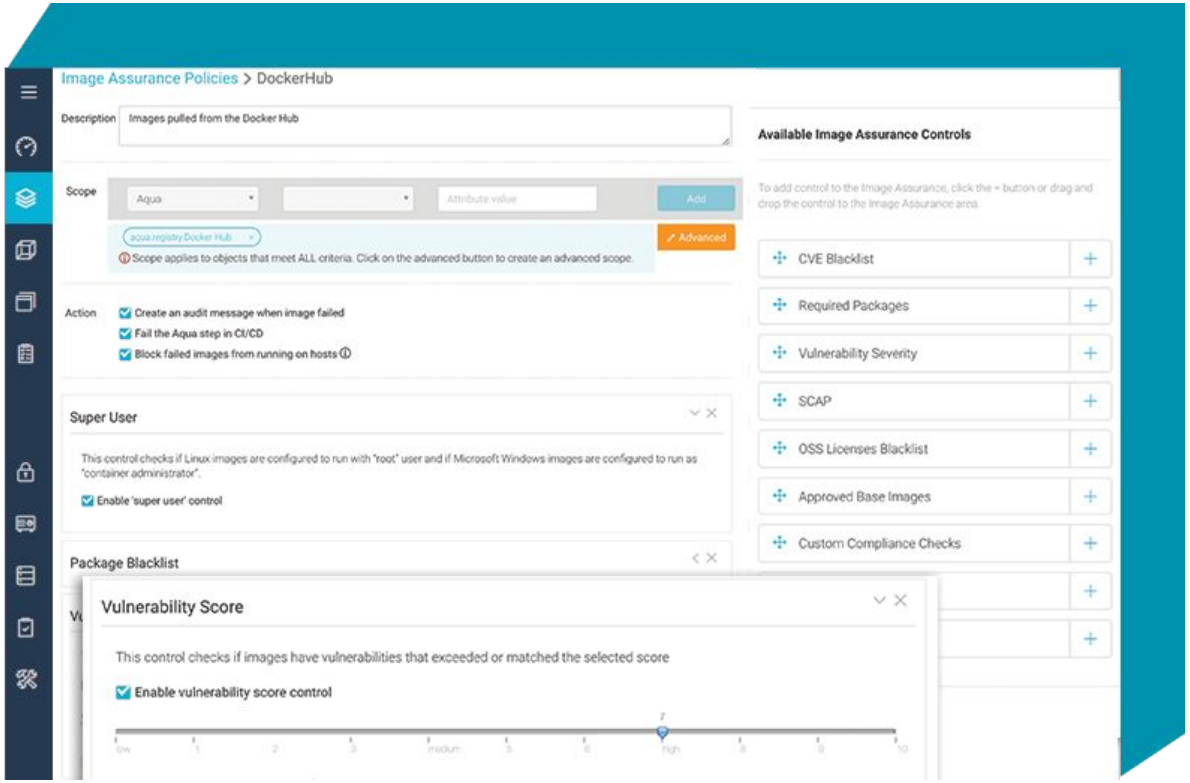
Code | Issues 0 | Pull requests 6 | Projects 0 | Wiki | Insights | Settings

Filters | is:pr is:open | Labels | Milestones | New pull request

6 Open | 13 Closed | Author | Labels | Projects | Milestones | Reviews | Assignee | Sort

- Bump redis from 2.10.6 to 3.0.1 dependencies
  #19 opened 18 days ago by dependabot bot
- Bump psycopg2 from 2.7.5 to 2.7.6.1 dependencies
  #18 opened 22 days ago by dependabot bot
- Bump markupsafe from 1.0 to 1.1.0 dependencies
  #17 opened 28 days ago by dependabot bot
- Bump itsdangerous from 0.24 to 1.1.0 dependencies
  #16 opened on 29 Oct by dependabot bot
- Bump click from 6.7 to 7.0 dependencies
  #14 opened on 26 Sep by dependabot bot
- Bump postgres from 2.2.1 to 2.2.2 dependencies
  #13 opened on 13 Sep by dependabot bot

Container scanning: aquasec



Test

- smoke tests
- test envs “à discretion”

Deployment

- atomic container deployment
- every deployment (and rollback) is a “normal deployment”
- deployment automation removes need for (all) devs root prod access and/or waiting for ops to deploy new dev version

Ops

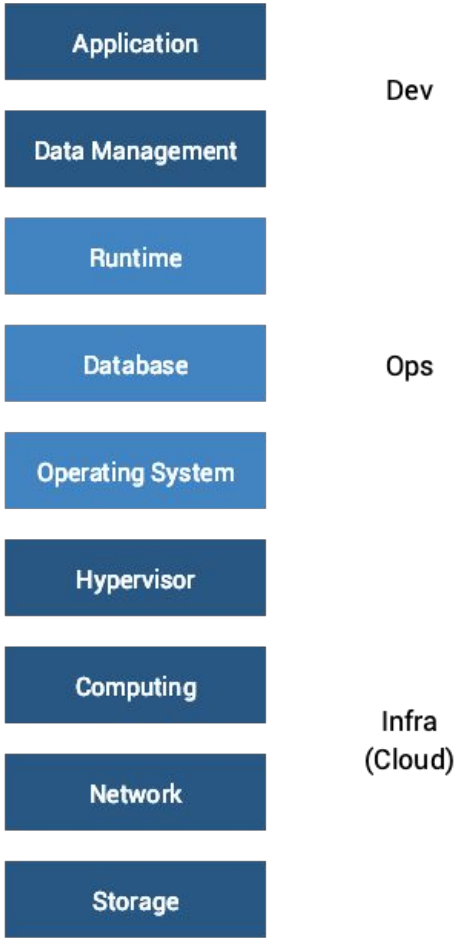
- standardization on (minimal, hardened) OS and container orchestrator
- immutable (application) infrastructure using containers
- process/storage/network separation of applications/environments
- detect/prevent configuration drift between dev/test/stage/prod envs
- documentation & automatic backup of all volumes
- documentation & monitoring of routes/loadbalancers/ingresspoints with enforcing SSL/TLS
- AAI for admin & application
- key & secrets management
- audit logging of control & application planes

Container isolation

- Kernel namespacing (process & network)
- Control groups (resource quota to prevent DoS)
- SELinux (additional syscall filter)
- prevent running as root inside container, no user-provided privileged containers (enforce best practice)
- readonly container filesystem (harder to persist exploit at runtime)

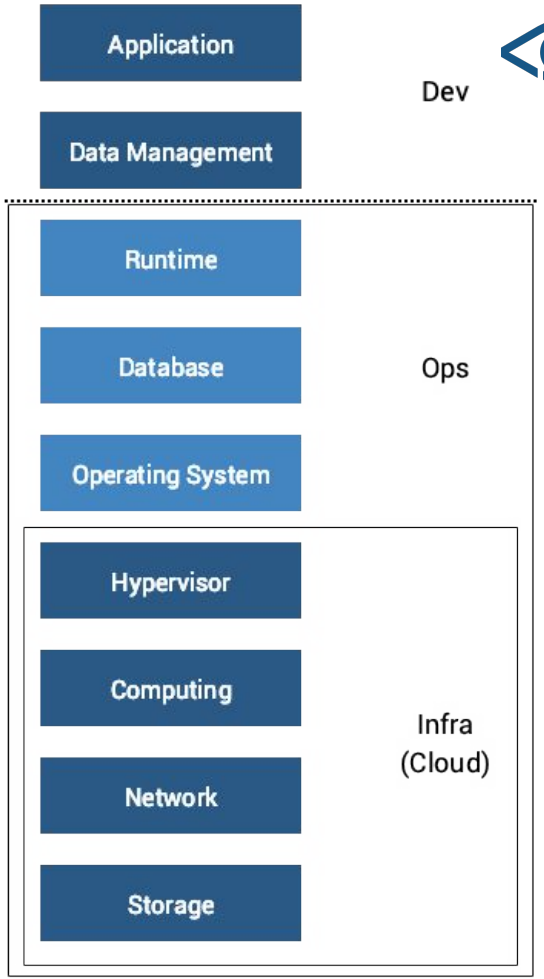
Traditional IT governance

- “Full Stack Audit”
- Review design document
- Every layer was custom built
  - physical hardware
  - handcrafted servers
  - manual application deployment
- Review each layer
- Review each layer again next year...



Cloud native IT governance

- Standardized components
  - already audited, some even externally certified
  - re-used, economies of scale, CMMI level 5
  - tech controls (AAI, RBAC, logs/SIEM) implemented once
  - financial controls implemented once
- Infrastructure: private/public cloud
- Ops: Container orchestration platform
- Review design document & platform configuration



IT governance controls in container platforms

- prevent configuration drift
  - immutable (application) infrastructure using containers
  - deploy dev/test/stage/prod envs from CI/CD
- prevent manual errors
  - validate configuration in CI/CD before deployment
  - standardization on (minimal, hardened) OS and container orchestrator
  - deployment automation removes need for (most) root prod access
- security by default
  - image scanning, dependency vulnerability management
  - process/storage/network separation of applications/environments
  - volumes & ingresspoints best practice (documentation, monitoring, backup, SSL/TLS/WAF)
  - AAI for admin & application, audit trail logging of CI/CD, control & application planes
  - key & secrets management
-

Thank you

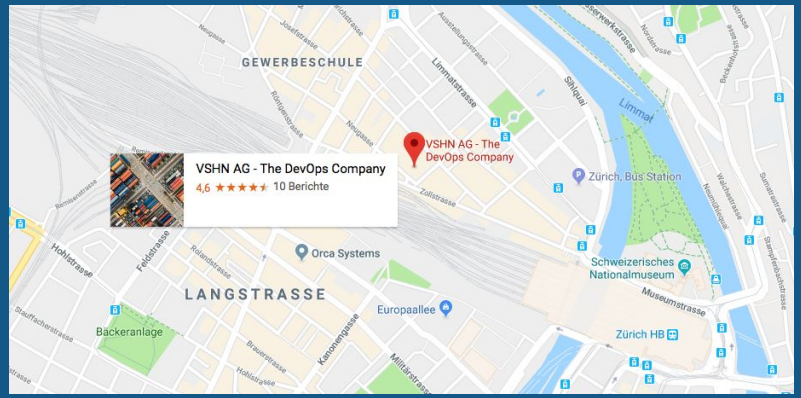
- Please get in touch with feedback
- Twitter: @aarnoaukia
- Linkedin: https://www.linkedin.com/in/aukia/
- Email: aarno.aukia@vshn.ch

DevSecOps Forum:

https://www.sig-switzerland.ch/devsecops_forum/

vshn.ch

Come visit us for a coffee!



https://vshn.ch/kontakt/

Follow us on Twitter!

@vshn_ch

VSHN AG - Neugasse 10 - CH-8005 Zürich - +41 44 545 53 00 - https://vshn.ch/ - info@vshn.ch