AGILE SECURITY

By Didier BERNAUDEAU

OSSIR (January 9th, 2018)

DISCLAIMER

I don't speak on behalf of my employer.

The information and perspective that I present are personal and don't represent those of my employer.

This presentation is the result of my personal researches and experimentation.

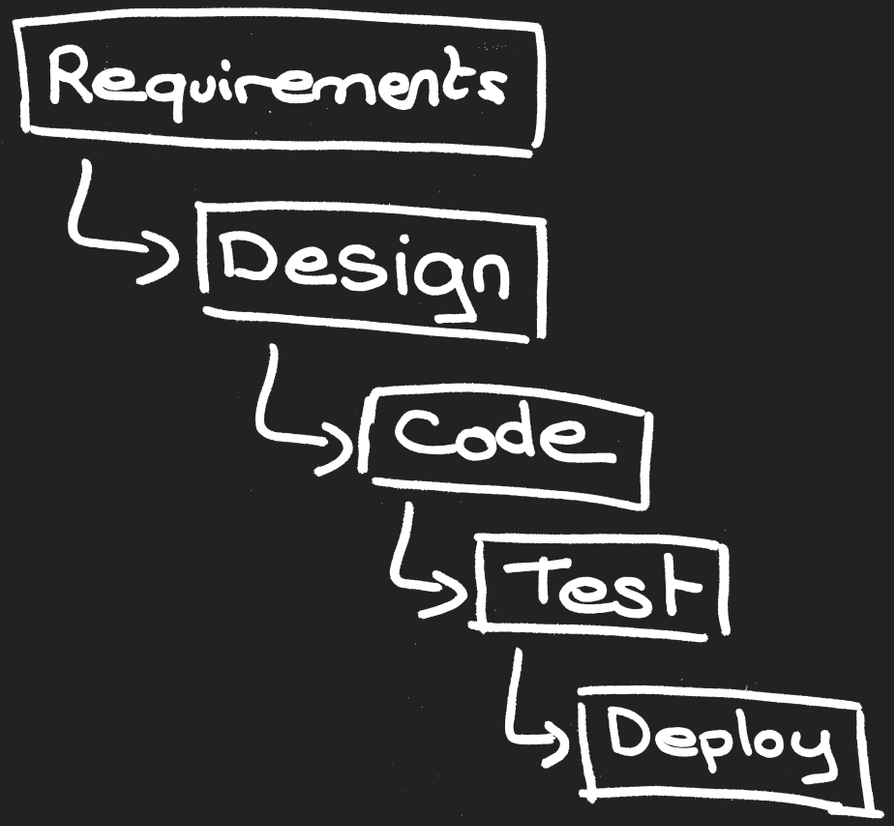
CHAPTER 1

THE REVELATION

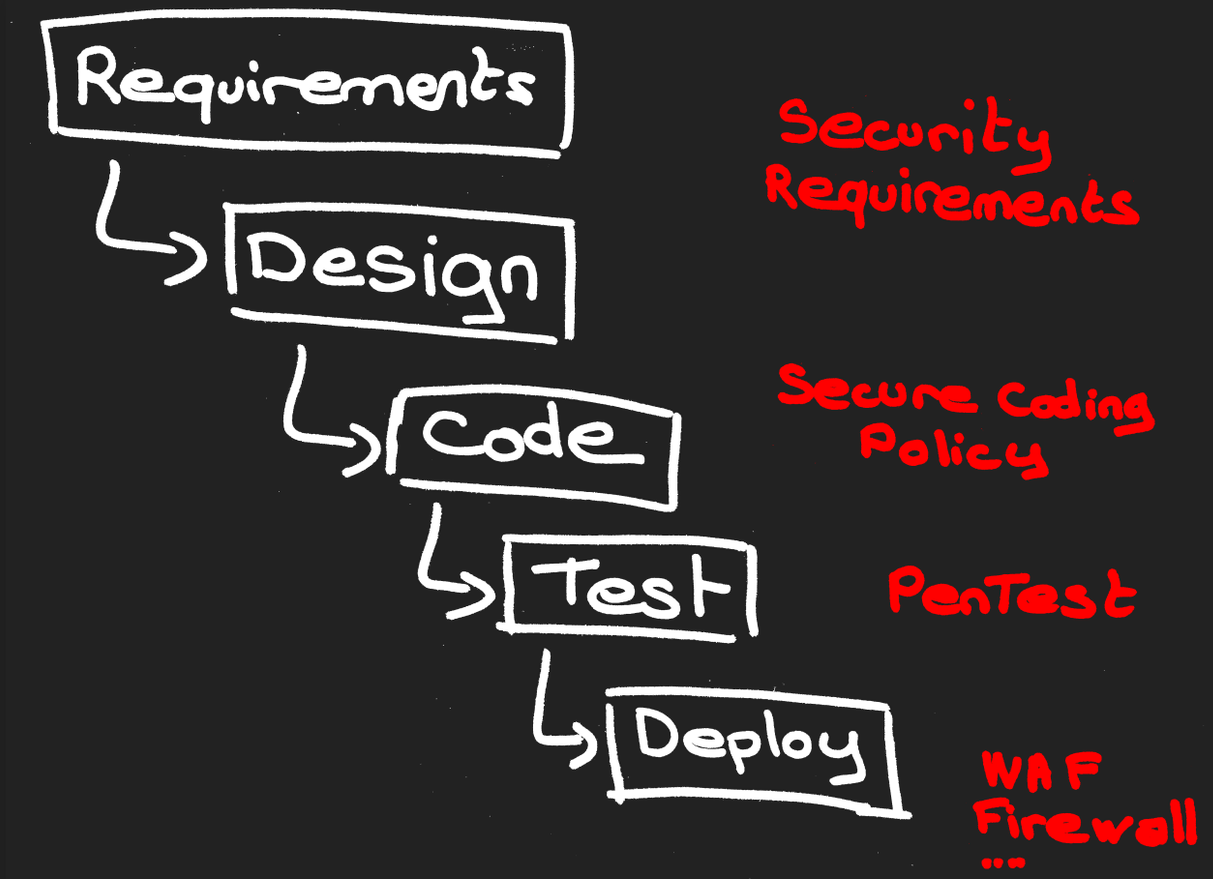
BOB

- Works for Cash Register Unlimited Company (since 2003)
- In charge of Application Security
- 70 projects per year
- 200 applications

SOFTWARE DEVELOPMENT LIFE CYCLE



SECURE SDLC


Requirements
Design
Code
Test
Deploy
Security Requirements
Secure Coding Policy
PenTest
WAF
Firewall
...

EVERYTHING ALL RIGHT!

UNTIL THE DAY OF ...

- **Alice:** Hi Bob! Are there security issues regarding email sending?

- **Bob:** Maybe, what is data?

- **Alice:** For instance, banking data (PAN, IBAN, ...)

- **Bob:** In this case, I must analyse your project. What is the deadline?

- **Alice:** This is already in production 2 weeks ago!

- **Bob:** Oh! How is it possible? without security validation? Without security acceptance testing?

- **Alice:** Well ... we use Agile Methogology!

AGILE ???

- Manifesto for Agile Software Development (February 2001)
- Scrum / Kanban
- Cash Register Unlimited Company has implemented Agile SDLC since 2013

CHAPTER 2

BECOME AN AGILE SECURITY OFFICER

PRODUCT THINKING

No Project

No Application

PRODUCT OWNER

- PO is the only person responsible for managing the Product Backlog.

- PO have a lot of stakeholders to take into account

Security Officer should become a major stakeholder

Alice is the Product Owner of "Cash Register 2.0"

PRODUCT BACKLOG

- It is a prioritized inventory of work to be done.

- Type of Product Backlog Item (PBI) :
  - Features (User Stories)
  - Non-Functional Requirement
  - Defects (Bug Stories)
  - Refactoring
  - ...

Security Officer should include security topics in the Product Backlog.

USER STORIES

Security features:

As seller, I want to change my password on the Cash Register

Acceptance Criteria:

The password is at least 8 characters. The password contains a character from each of the following groups: Lower case alphabet, Upper case alphabet, Numbers Special Characters (!,@,#,$,%,^,&,)*

SECURITY-FOCUSED STORIES

- Approach introduced by Safe Code
- A way to include non-functionnal requirement in the backlog

Example: *As developer, I want to verify that sensitive data is kept restricted to actors authorized to access it.*

EVIL USER STORIES

aka "Abuser Stories"

- Approach introduced by OWASP
- Using Personas: Insider Hacker, Professional hackers, Script kiddie, ...

Example: *As a hacker, I can modify the price of an article.*

SECURITY IN PRODUCT BACKLOG

- User Stories with acceptance criteria
- Security-focused stories (NFR)
- Evil user Stories

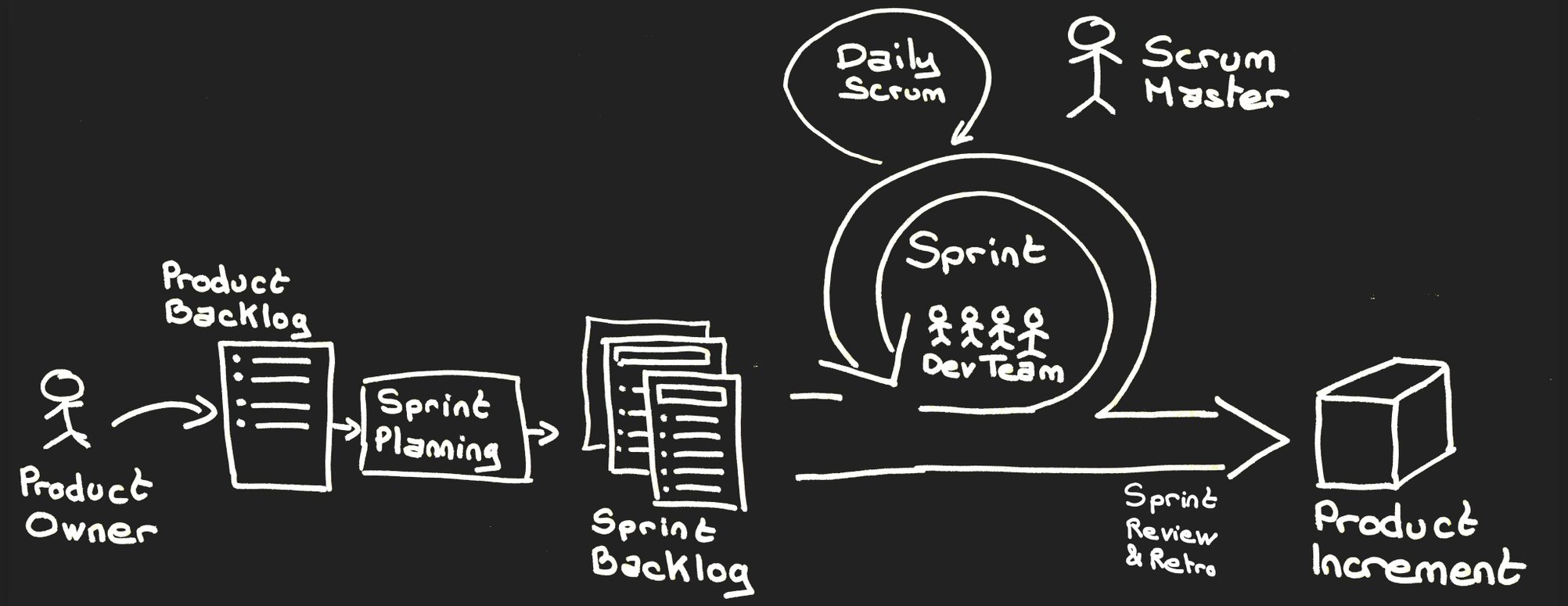
DEFINITION OF DONE

List of activities to validate each item in the Product Backlog.

Security Officer should include security in DoD.

Example of secure activity: *There should be no open critical and high vulnerability identified by Source Code Analysis*

SPRINT



Security Officer should take part in sprint meeting.

MINIMAL PRODUCT

Minimal Viable Product (MVP)

Product which allows a team to test an ideas with the least effort.

Minimal Marketable Product (MMP)

Product with the smallest possible feature set that addresses the needs of the initial users.

Security Officer should define the
Minimal Viable Security (MVS)
for the product.

CHAPTER 3

SECURITY IN SPRINT PHASES

SPRINT PHASES

1. Code
2. Test
3. Deploy

PHASE 1 # CODE

COWBOY CODING

- **Prevent "cowboy" development:**
  - Define allowed frameworks
  - Define security guideline for each framework
  - Change management

- Identify framework with known vulnerabilities:
  - Artifact repository: JFrog X-Ray, BlackDuck Hub, ...
  - Build: Dependency Check / RetireJS

MICROSERVICE

- Best agile software architecture

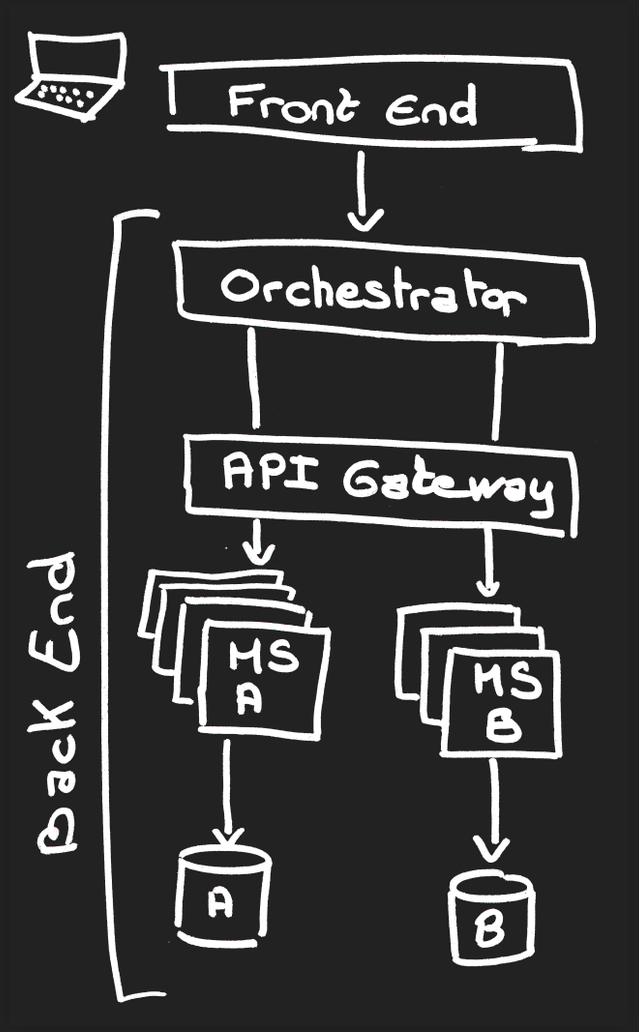
- 2 parts:
  - Front End (WebApp / MobileApp)
  - Back End (MicroService)

SECURING MICROSERVICE

- Front End (WebApp):
  - Linter (ESlint Security)
  - Minify and Obfuscate (UglifyJS)

- Back End (Microservice):
  - Stateless & Autoscalling
  - Authentication Token (OAuth / JWT)
  - HTTPS
  - Privileged **Orchestration** pattern to Choreography

MICROSERVICE



PHASE 2 # TEST

TEST-DRIVEN DEVELOPMENT

1. Start by writing an automated test case.
2. Run the test which should fail.
3. Write the minimum amount of code required to make the test pass
4. Run the tests to check the new test passes
5. Refactor the new code

Positive testing (Valid data) and **Negative testing** (Invalid data)

BEHAVIOR-DRIVEN DEVELOPMENT

- Integration test
- Test are written with DSL (Domain-specific language) like **Gherkin**

```
Feature: Account Holder withdraws cash

Scenario: Account has sufficient funds
 Given the account balance is $100
   And the card is valid
   And the machine contains enough money
  When the Account Holder requests $20
  Then the ATM should dispense $20
   And the account balance should be $80
   And the card should be returned
```

COMMON TESTING TOOLS

- Fitness
- Mockito
- Cucumber
- Selenium
- JBehave (Java)
- Behat (PHP)
- Hiptest

SECURITY TESTING TOOLS

- ZAP
- Gauntlt (Be mean to your code and like it)
- BDD Security

PHASE 3 # DEPLOY

STRATEGY DEPLOYMENT

New environment for each deployment

"Blue/green" or Canary release

INFRASTRUCTURE AS CODE (IAC)

- Tools: Chef, Puppet, Ansible, ...
- Test Driven Infrastructure:
  - Linter: puppet-lint, Ansible lint, Foodcritic, RuboCop, ...
  - Unit testing: RSpec-Puppet, ChefSpec, ...
  - Acceptance testing: Beaker for puppet, Test kitchen for Chef, ...
- Network As Code (LaaS and FaaS): Neutron from RedHat

CONTAINER

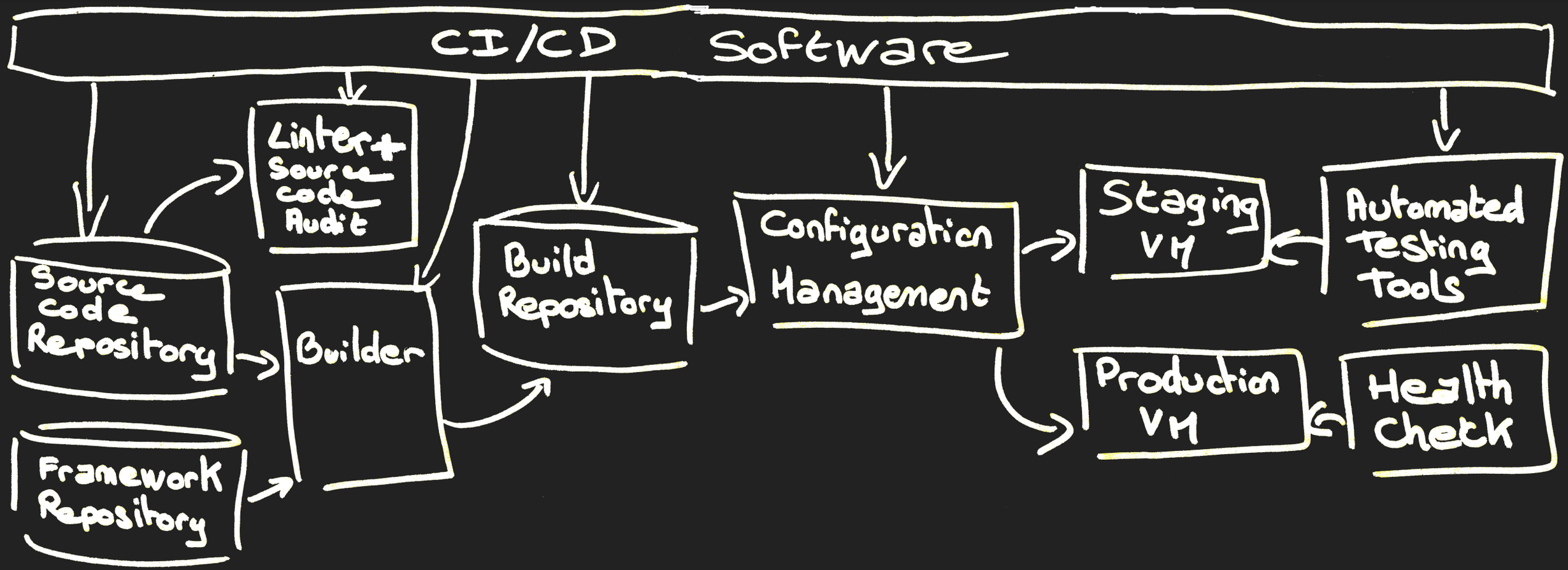
- Software:
  - Container runtime: runC, Docker, Rocket, Garden, ...
  - PAAS: OpenShift, CloudFoundry, Bluemix, ...

- Secret storage: Vault from Hashicorp, Barbican from RedHat, ...
- Network Overlay & Micro Segmentation
- Segregate Containers by host (CoreOS)
- Container vulnerabilities Scanner (Clair)

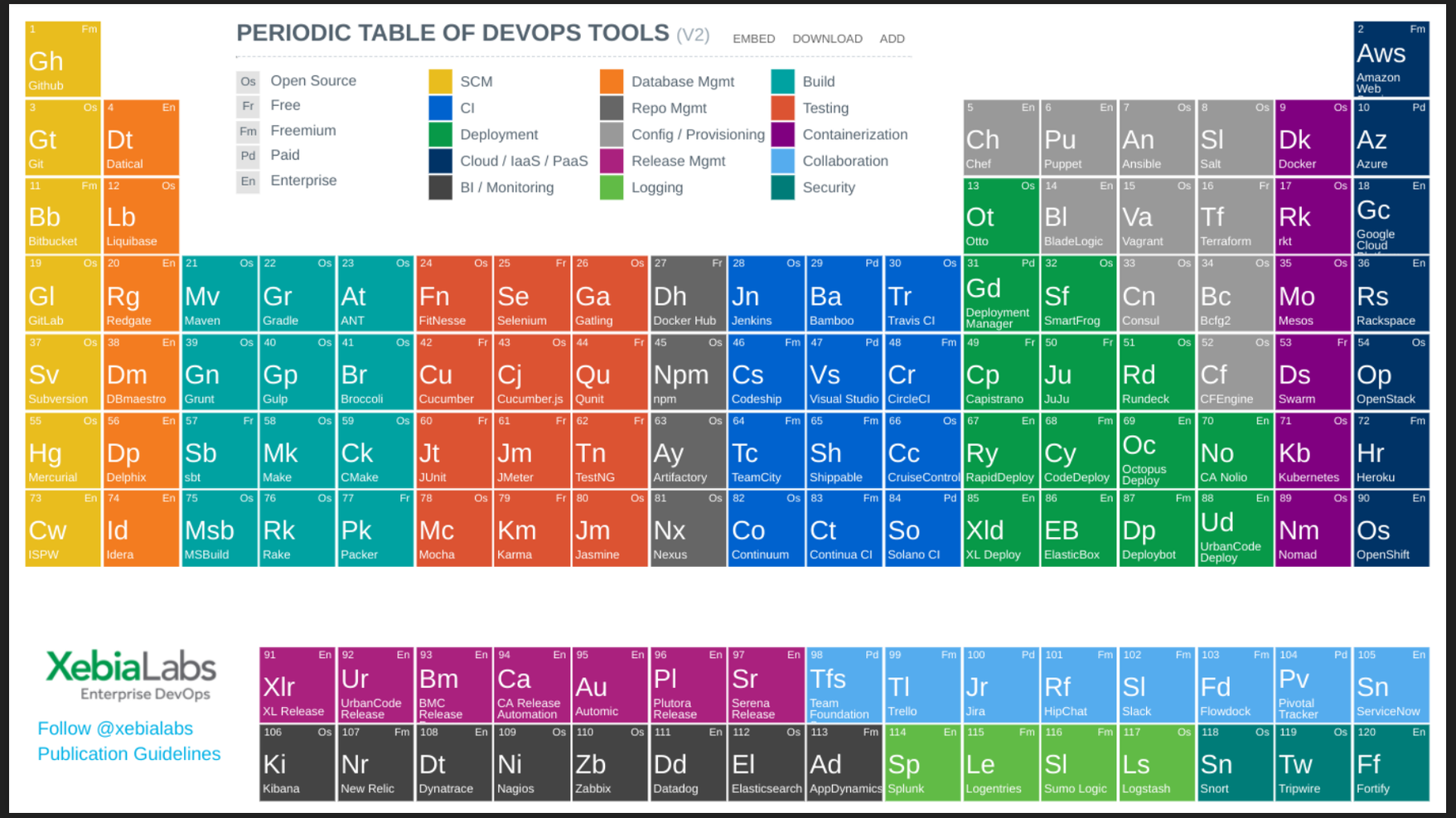
CHAPTER 4

SECURE SOFTWARE SUPPLY CHAIN

SOFTWARE SUPPLY CHAIN



TOOLS


PERIODIC TABLE OF DEVOPS TOOLS (V2)
EMBED
DOWNLOAD
ADD
Os Open Source
Fr Free
Fm Freemium
Pd Paid
En Enterprise
SCM
CI
Deployment
Cloud / IaaS / PaaS
BI / Monitoring
Database Mgmt
Repo Mgmt
Config / Provisioning
Release Mgmt
Logging
Build
Testing
Containerization
Collaboration
Security
1 Fm Gh Github
2 Fm Aws Amazon Web
3 Os Gt Git
4 En Dt Datical
5 En Ch Chef
6 En Pu Puppet
7 Os An Ansible
8 Os Sl Salt
9 Os Dk Docker
10 Pd Az Azure
11 Fm Bb Bitbucket
12 Os Lb Liquibase
13 Os Ot Otto
14 En Bl BladeLogic
15 Os Va Vagrant
16 Fr Tf Terraform
17 Os Rk rkt
18 En Gc Google Cloud
19 Os Gl GitLab
20 En Rg Redgate
21 Os Mv Maven
22 Os Gr Gradle
23 Os At ANT
24 Os Fn FitNesse
25 Fr Se Selenium
26 Os Ga Gatling
27 Fr Dh Docker Hub
28 Os Jn Jenkins
29 Pd Ba Bamboo
30 Os Tr Travis CI
31 Pd Gd Deployment Manager
32 Os Sf SmartFrog
33 Os Cn Consul
34 Os Bc Bcfg2
35 Os Mo Mesos
36 En Rs Rackspace
37 Os Sv Subversion
38 En Dm DBmaestro
39 Os Gn Grunt
40 Os Gp Gulp
41 Os Br Broccoli
42 Fr Cu Cucumber
43 Os Cj Cucumber.js
44 Fr Qu Qunit
45 Os Npm npm
46 Fm Cs Codeship
47 Pd Vs Visual Studio
48 Fm Cr CircleCI
49 Fr Cp Capistrano
50 Fr Ju JuJu
51 Os Rd Rundeck
52 Os Cf CFEngine
53 Fr Ds Swarm
54 Os Op OpenStack
55 Os Hg Mercurial
56 En Dp Delphix
57 Fr Sb sbt
58 Os Mk Make
59 Os Ck CMake
60 Fr Jt JUnit
61 Fr Jm JMeter
62 Fr Tn TestNG
63 Os Ay Artifactory
64 Fm Tc TeamCity
65 Fm Sh Shippable
66 Os Cc CruiseControl
67 En Ry RapidDeploy
68 Fm Cy CodeDeploy
69 En Oc Octopus Deploy
70 En No CA Nolio
71 Os Kb Kubernetes
72 Fm Hr Heroku
73 En Cw ISPW
74 En Id Idera
75 Os Msb MSBuild
76 Os Rk Rake
77 Fr Pk Packer
78 Os Mc Mocha
79 Fr Km Karma
80 Os Jm Jasmine
81 Os Nx Nexus
82 Os Co Continuum
83 Fm Ct Continua CI
84 Pd So Solano CI
85 En Xld XL Deploy
86 En EB ElasticBox
87 Fm Dp Deploybot
88 En Ud UrbanCode Deploy
89 Os Nm Nomad
90 En Os OpenShift
XebiaLabs
Enterprise DevOps
Follow @xebialabs
Publication Guidelines
91 En Xlr XL Release
92 En Ur UrbanCode Release
93 En Bm BMC Release
94 En Ca CA Release Automation
95 En Au Automic
96 En Pl Plutora Release
97 En Sr Serena Release
98 Pd Tfs Team Foundation
99 Fm Tl Trello
100 Pd Jr Jira
101 Fm Rf HipChat
102 Fm Sl Slack
103 Fm Fd Flowdock
104 Pd Pv Pivotal Tracker
105 En Sn ServiceNow
106 Os Ki Kibana
107 Fm Nr New Relic
108 En Dt Dynatrace
109 Os Ni Nagios
110 Os Zb Zabbix
111 En Dd Datadog
112 Os El Elasticsearch
113 Fm Ad AppDynamics
114 En Sp Splunk
115 Fm Le Logentries
116 Fm Sl Sumo Logic
117 Os Ls Logstash
118 Os Sn Snort
119 Os Tw Tripwire
120 En Ff Fortify

SOME BEST PRACTICES

For securing your Software Supply Chain

- HTTPS
- Authentication
- Access Management

ANY QUESTIONS ?

View online at https://git.io/vNtqD