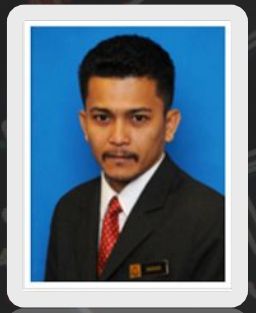
DevOps Transformation



Ts. Hariadi bin Hinta

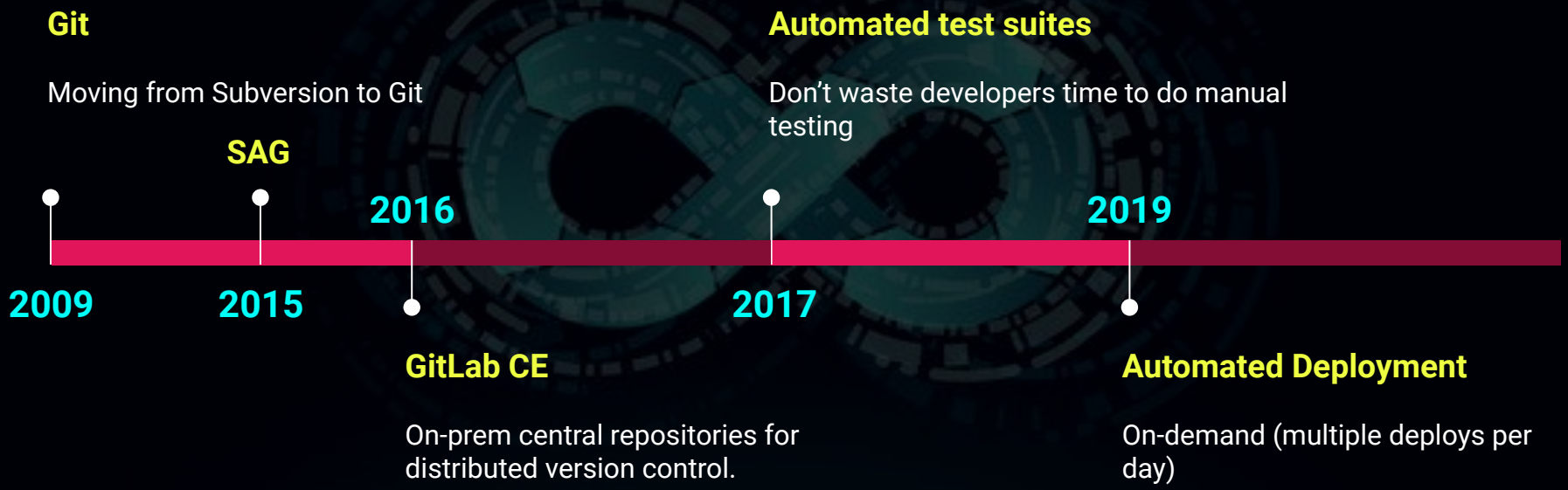
Jabatan Perkhidmatan Awam,
Sektor Aplikasi Generik,
Cawangan Pengurusan Pembangunan Aplikasi,
Bahagian Digital Dan Teknologi Maklumat

CLOUD 15
Perintis Kepakaran Cloud

Background

- 25+ generic web apps
- In-house
- Rapid Development (1-2 weeks, 1-12 month timeline)
- Dynamically changing software requirements
- Multi-tenancy (database context)

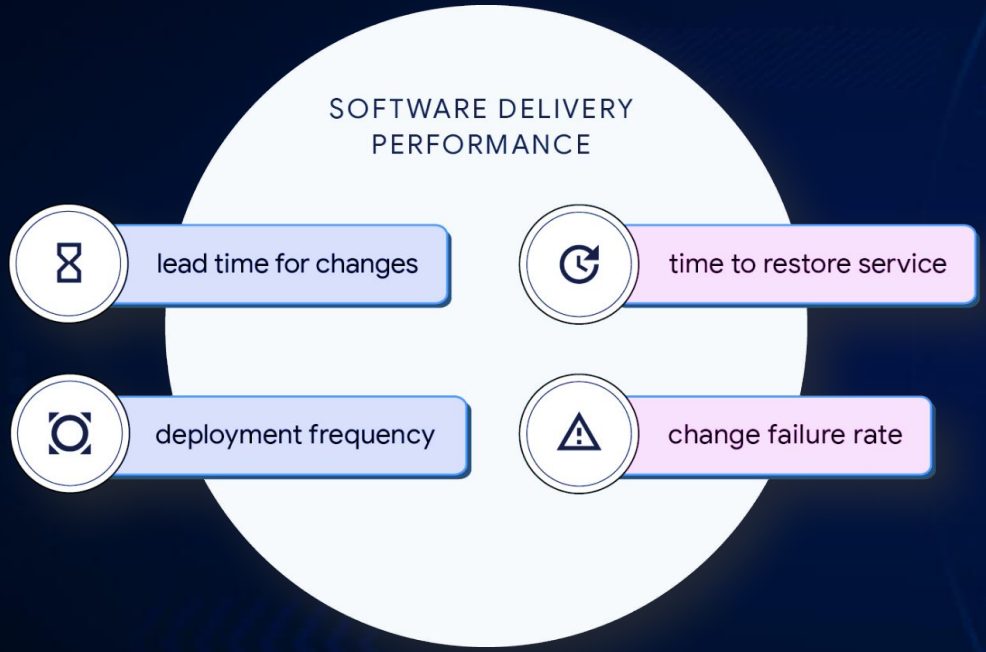
DevOps Journey



WHY WE DO DEVOPS

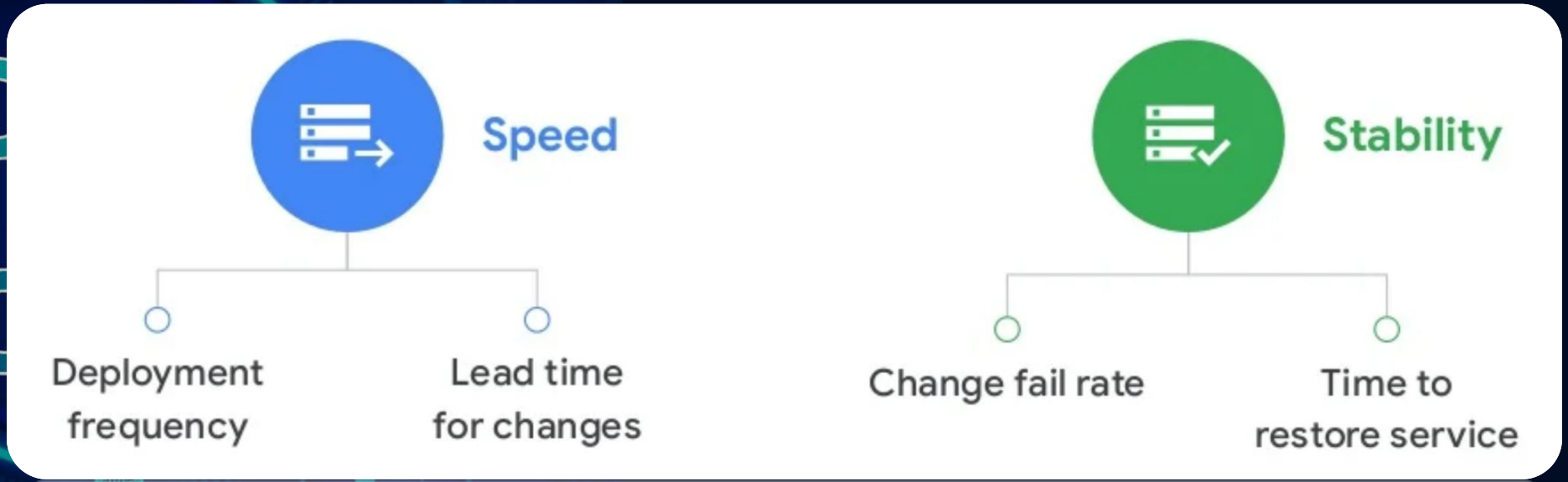
- Fun, enjoy and worthy
- Enable faster development
- Increase the speed of our deployments
- Improve the stability of our software
- Build security in from the start
- Well-architected design (Application, Governance, Infrastructure)

WHAT ASPECT



Software Delivery & Operations (SDO) Performance

Software Delivery Performance - 4 Key Metrics



DevOps Quick check

https://www.devops-research.com/quickcheck.html

Speed: Lead time for changes

How long does it take to go from code committed to code successfully running in production?

Speed: Deployment frequency

x8 improvement

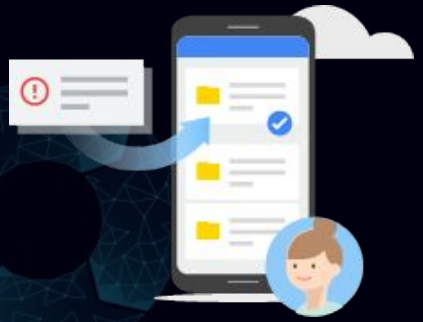
once per week

Multiple deploy per day

How often does your organization deploy code to production or release it to end users?

Stability: Time to restore / recovery

x9 improvement



Less than one hour

how long does it generally take to restore service when a service incident or a defect that impacts users occurs (for example, unplanned outage, service impairment)?

Stability: Change fail percentage

What percentage of changes to production or releases to users result in degraded service (for example, lead to service impairment or service outage) and subsequently require remediation (for example, require a hotfix, rollback, fix forward, patch)?

HOW WE DO IT

APPLICATION

GOVERNANCE

INFRASTRUCTURE

MODERN TECHNOLOGY

Single Source Of Truth

GitLab

CPU: 16

RAM: 16GB

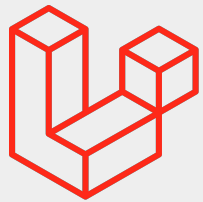
GitLab Runner

Concurrent: 4

Executor: docker

APPLICATION

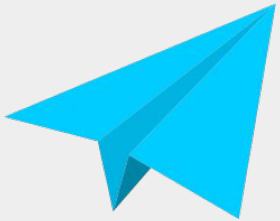
- The Twelve-Factor App (https://12factor.net)
- Version control (including database schemas) - Git
- Tech Stack: Tailwind / Alpinejs / Livewire / Laravel (TALL)
- Automated testing - grouped by modules, Testing Driven Development (TDD)
- Deployment automation - Deployer (.gitlab-ci.yaml & deploy.yaml)


Laravel


Livewire

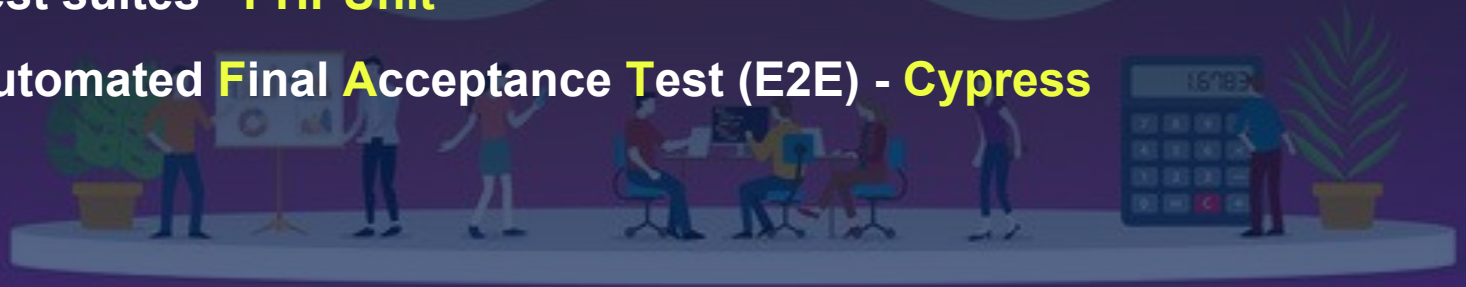

Alpinejs


Tailwind CSS


Deployer

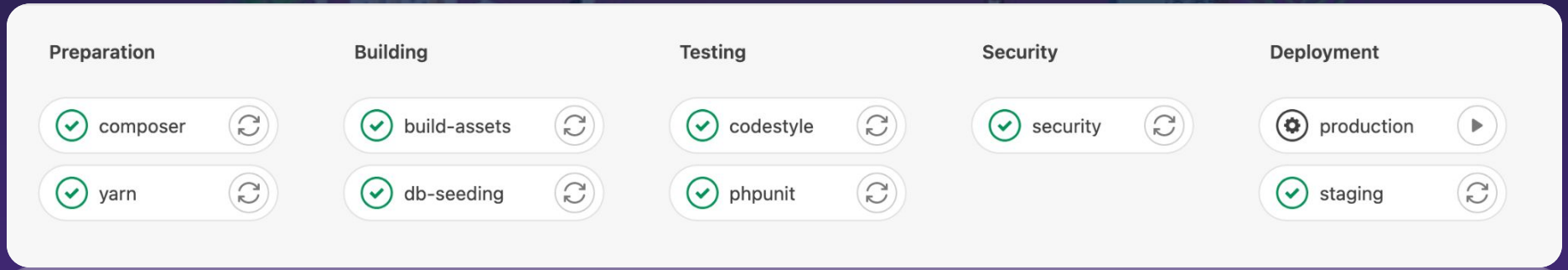
APPLICATION (cont.)

- Development - Vagrant / Valet / Kool / Docker (ship with docker-compose.yml)
- Debug Tools - Laravel Debugbar / Telescope
- Style Guide - Laravel Pint (on top PHP-CS-Fixer)
- Test suites - PHPUnit
- Automated Final Acceptance Test (E2E) - Cypress

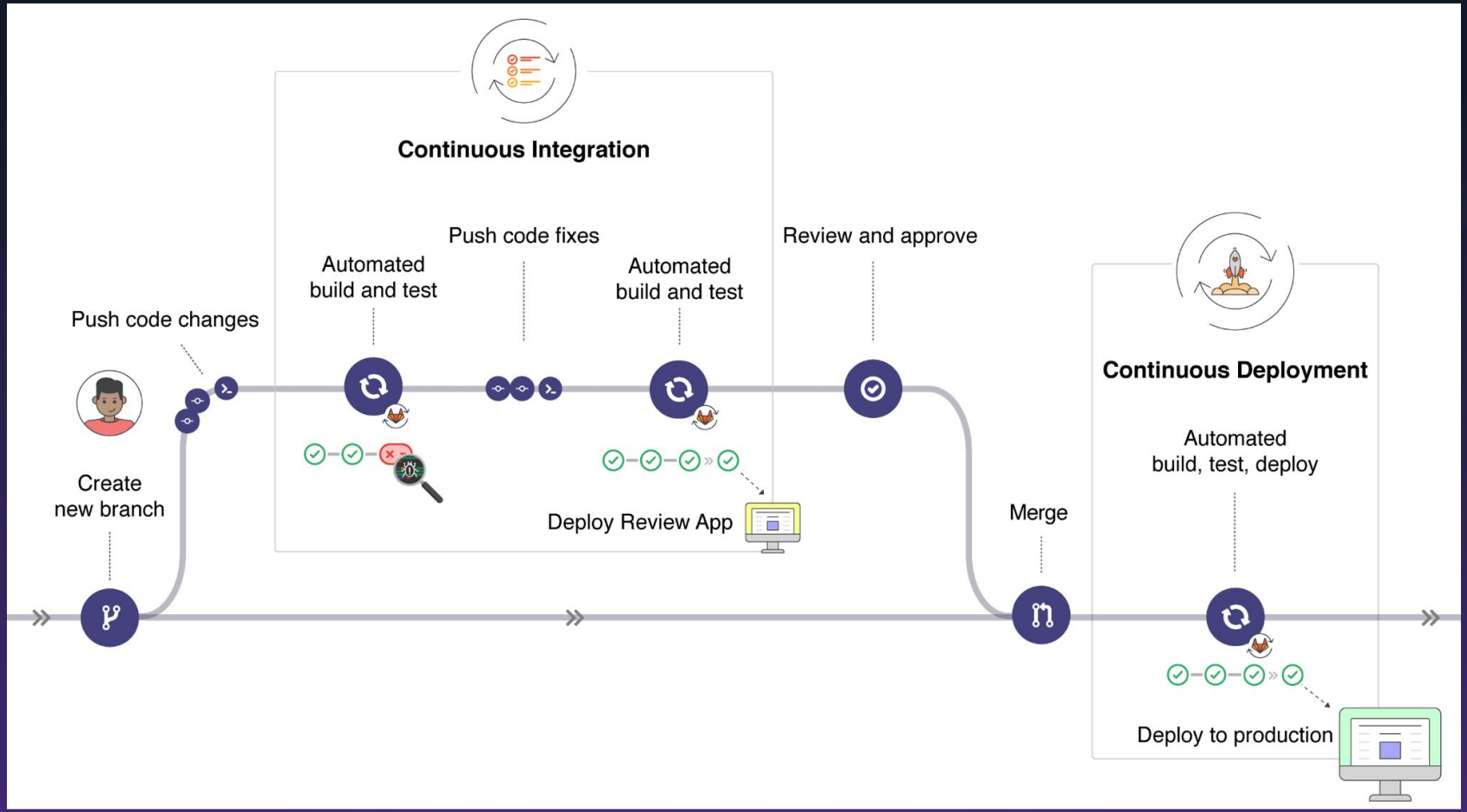


GOVERNANCE

- **Milestone** as sprint/iteration
- **Issue** prioritized by labels (xs, s, m, l, xl)
- **Issue Board** as Kanban Board
- System Analyst as **Maintainer**
- Branching Model: **main** (production) - **protected**
- **Merge Request** need to pass pipeline - Code Integration/ Code Deployment (CI/CD)



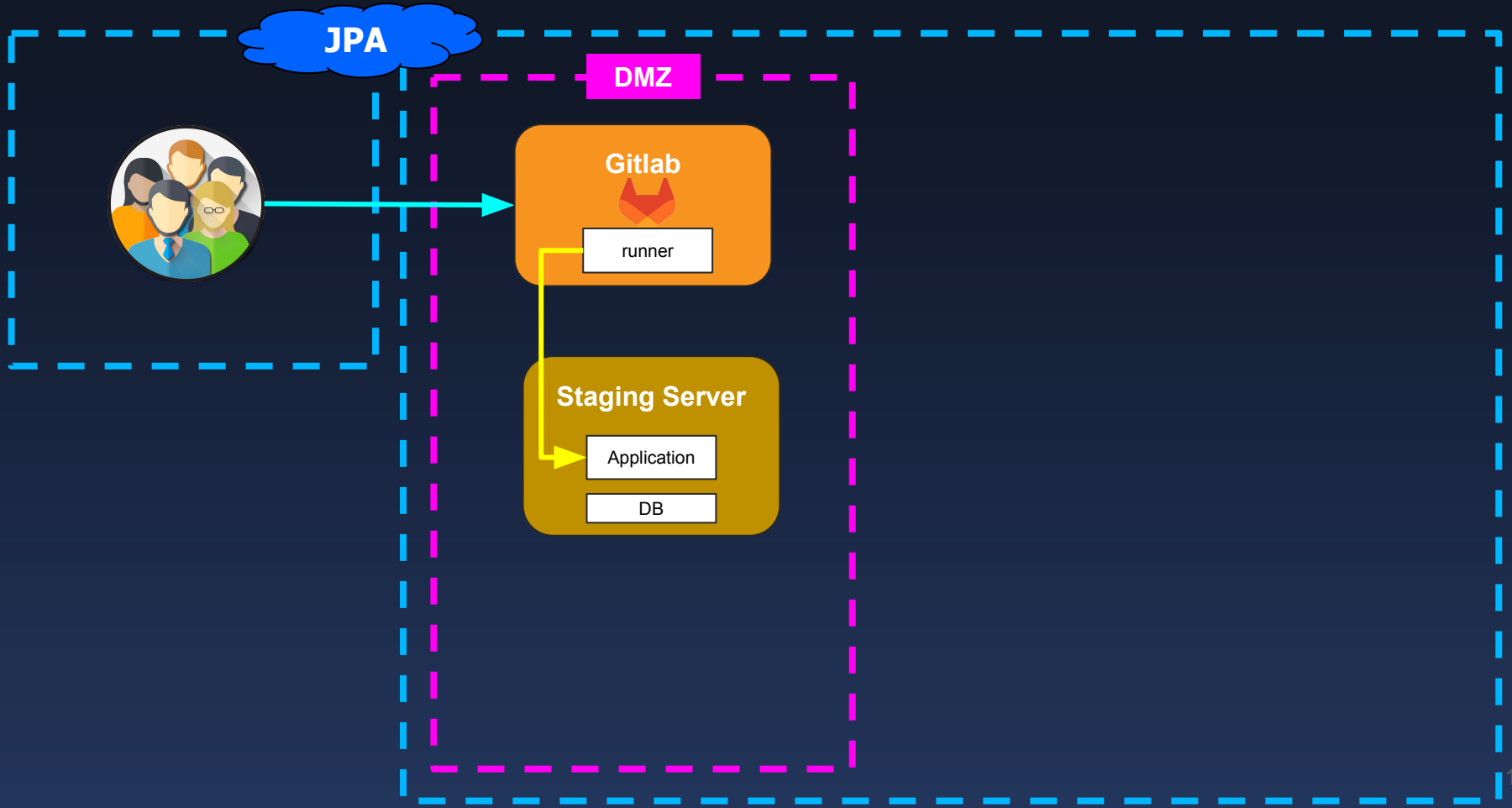
CI/CD Workflow



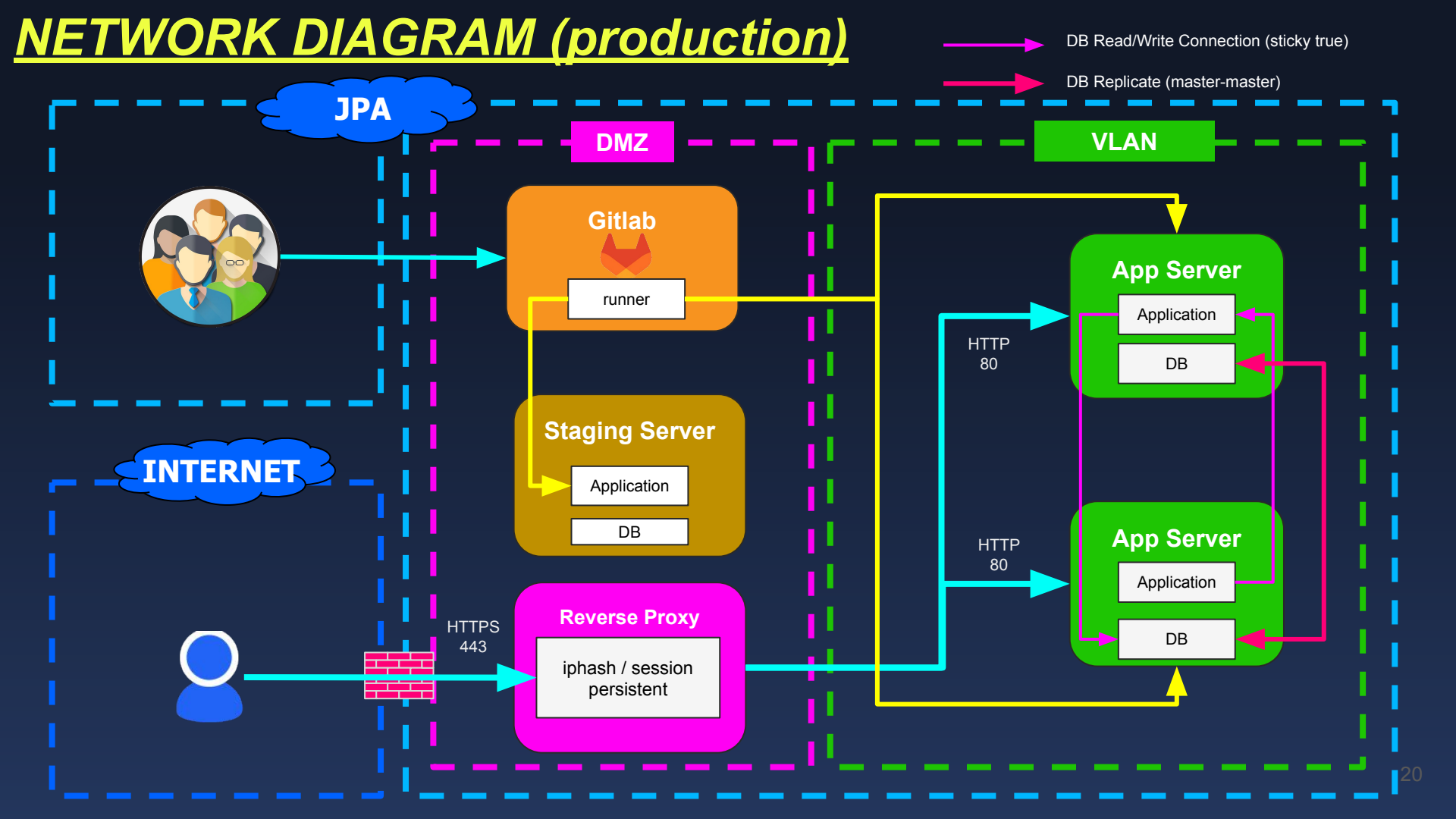
INFRASTRUCTURE

- 2 environment:
  - staging (projek-staging.jpa.gov.my)
  - production (projek.jpa.gov.my)
- Bare metal VM Specifications:
  - Ubuntu 22.04 Long Term Support (LTS)
  - Nginx
  - PHP 8.1
  - MySQL 8.0
  - Supervisor
  - Redis
  - Mailhog (staging) / MyGovUC SMTP Relay (prod)

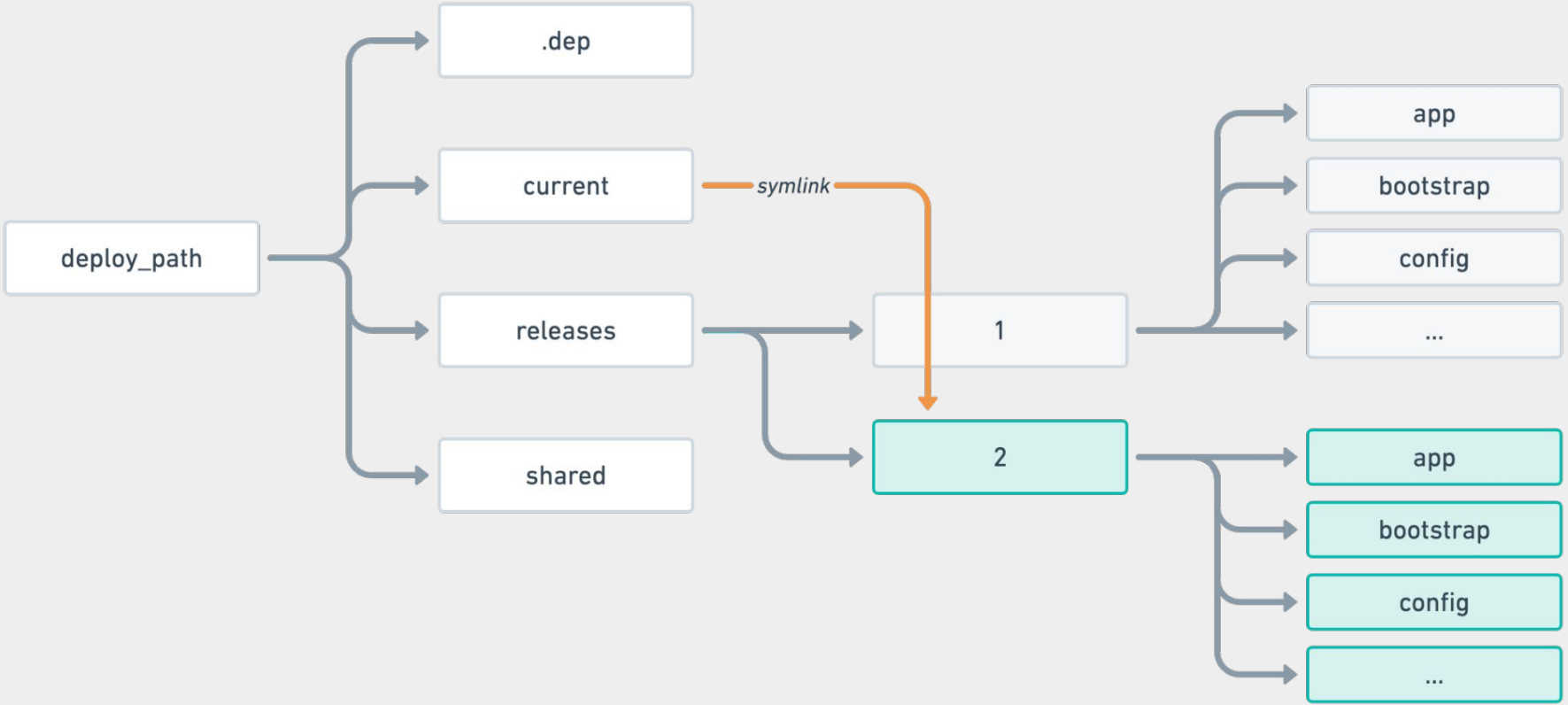
NETWORK DIAGRAM (development phase)



NETWORK DIAGRAM (production)



Strategy: Zero Downtime Deployment (Deployer)



.gitlab-ci.yaml

staging

```yaml
staging:
  image: lorisleiva/laravel-docker:8.1
  stage: deployment
  tags:
    - sag
  before_script:
    - *init_ssh
  script:
    - composer deploy-staging
  environment:
    name: staging
    url: https://award-gcr-staging.jpa.gov.my
  only:
    - main
```

production

```yaml
production:
  image: lorisleiva/laravel-docker:8.1
  stage: deployment
  tags:
    - sag
  before_script:
    - *init_ssh
  script:
    - composer deploy
  environment:
    name: production
    url: https://award-gcr.jpa.gov.my
  # Do not run automatically.
  # Wait for a human to click on play.
  when: manual
  only:
    - master
```

DEVOPS ROI *It is worth it?*

Value-Driven Categories

- Value Gained From Unnecessary Rework Avoided per Year
- Potential Value Added from Reinvestment in New Features

Cost-Driven Categories

- Cost of Downtime Per Year

Calculation base on **State of DevOps 2021** (32, 000 respondents/assessments)
https://services.google.com/fh/files/misc/state-of-devops-2021.pdf

Statistics from State of DevOps 2021

Software delivery performance metric	Elite	High	Medium	Low
Deployment Frequency	On-demand (multiple deploys per day)	Between once per week and once per month	Between once per month and once every 6 months	Fewer than once per six months
Lead time for changes	Less than one hour	Between one day and one week	Between one month and six months	More than six months
Time to restore service	Less than one hour	Less than one day	Between one day and one week	More than six months
Change failure rate	0%-15%	16%-30%	16%-30%	16%-30%

Value Gained From Unnecessary Rework Avoided per Year

Technical Staff Size = F: **15143**, BDTM: **174**, CPPA: **67**

Average Salary = **RM60,000** (median salary, RM5 x 12)

Benefits Multiplier = **1.5** (median salary, RM5 x 12)

Percent of Spend on Unnecessary Rework :

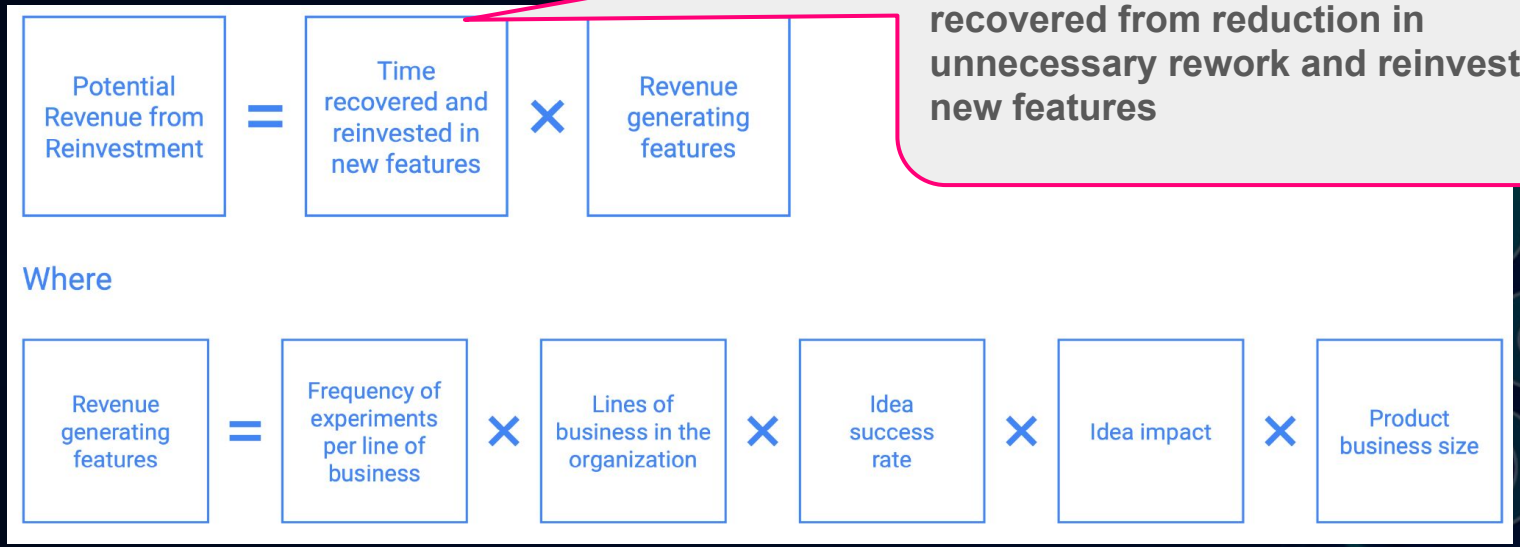
- Base: 18% (between 19% and 40% of code is reworked prior to final release*)
- **Elite**: **1%** (amount of unnecessary rework reported is **19%**)
- **High**: **1.5%** (amount of unnecessary rework reported is **19.5%**)
- **Medium**: **2%** (amount of unnecessary rework reported is **20%**)
- **Low**: **2%** (amount of unnecessary rework reported is **20%**)

*Morozoff, E. (2009, September 4). Using a Line of Code Metric to Understand Software Rework. Retrieved from http: ieeexplore.ieee.org/document/5232799/

Yearly returns possible from cost of unnecessary rework avoided (value of rework recovered)

	Elite	High	Medium	Low
F	15143 staff x RM60,000 salary x 1.5 benefits x 1% rework **= RM13.6M**	15143 staff x RM60,000 salary x 1.5 benefits x 1.5% rework **= RM20M**	15143 staff x RM60,000 salary x 1.5 benefits x 2% rework **= RM27.2M**	15143 staff x RM60,000 salary x 1.5 benefits x 2% rework **= RM27.2M**
BDTM	174 staff x RM60,000 salary x 1.5 benefits x 1% rework **= RM157K**	174 staff x RM60,000 salary x 1.5 benefits x 1.5% rework **= RM235K**	174 staff x RM60,000 salary x 1.5 benefits x 2% rework **= RM313K**	174 staff x RM60,000 salary x 1.5 benefits x 2% rework **= RM313K**
CPPA	67 staff x RM60,000 salary x 1.5 benefits x 1% rework **= RM60K**	67 staff x RM60,000 salary x 1.5 benefits x 1.5% rework **= RM90K**	67 staff x RM60,000 salary x 1.5 benefits x 2% rework **= RM121K**	67 staff x RM60,000 salary x 1.5 benefits x 2% rework **= RM121K**

We calculate Potential Value Added from Reinvestment using the following equation:



Key idea: Leverage time recovered from reducing inefficiencies, and turn that into value by using it to generate revenue through new features for our users.

Potential value added from reinvestment in new features (**RM500M** / **RM100** / **RM10M** portfolio)

	Elite	High	Medium	Low
F (15,143 technical staff)	**1%** time recovered x **730** experiments/year x **20** lines of business x **1/3** success rate x **1%** idea impact x **RM500M** product business **= RM243.3M return**	**1.5%** time recovered x **52** experiments/year x **20** lines of business x **1/3** success rate x **1%** idea impact x **RM500M** product business **= RM26M return**	**2%** time recovered x **12** experiments/year x **20** lines of business x **1/3** success rate x **1%** idea impact x **RM500M** product business **= RM8M return**	**2%** time recovered x **2** experiments/year x **20** lines of business x **1/3** success rate x **1%** idea impact x **RM500M** product business **= RM1.3M return**
BDTM (174 technical staff)	1% time recovered x 730 experiments/year x 8 lines of business x 1/3 success rate x 1% idea impact x **RM100M** product business **= RM48.7M return**	1.5% time recovered x 52 experiments/year x 8 lines of business x 1/3 success rate x 1% idea impact x **RM100M** product business **= RM5.2M return**	2% time recovered x 12 experiments/year x 8 lines of business x 1/3 success rate x 1% idea impact x **RM100M** product business **= RM1.6M return**	2% time recovered x 2 experiments/year x 8 lines of business x 1/3 success rate x 1% idea impact x **RM100M** product business **= RM267K return**
CPPA (67 technical staff)	1% time recovered x 730 experiments/year x 1 lines of business x 1/3 success rate x 1% idea impact x **RM10M** product business **= RM243K return**	1.5% time recovered x 52 experiments/year x 1 line of business 1/3 success rate x 1% idea impact x **RM10M** product business **= RM26K return**	2% time recovered x 12 experiments/year x 1 line of business 1/3 success rate x 1% idea impact x **RM10M** product business **= RM8K return**	2% time recovered x 2 experiments/year x 1 line of business 1/3 success rate x 1% idea impact x **RM10M** product business **= RM1.3K return**

Potential value added from reinvestment in new features (**RM500M** / **RM200** / **RM50M** portfolio)

	Elite	High	Medium	Low
F (15,143 technical staff)	**1%** time recovered x **730** experiments/year x **20** lines of business x **1/3** success rate x **1%** idea impact x **RM500M** product business **= RM243.3M return**	**1.5%** time recovered x **52** experiments/year x **20** lines of business x **1/3** success rate x **1%** idea impact x **RM500M** product business **= RM26M return**	**2%** time recovered x **12** experiments/year x **20** lines of business x **1/3** success rate x **1%** idea impact x **RM500M** product business **= RM8M return**	**2%** time recovered x **2** experiments/year x **20** lines of business x **1/3** success rate x **1%** idea impact x **RM500M** product business **= RM1.3M return**
BDTM (174 technical staff)	1% time recovered x 730 experiments/year x 8 lines of business x 1/3 success rate x 1% idea impact x **RM200M** product business **= RM97.3M return**	1.5% time recovered x 52 experiments/year x 8 lines of business x 1/3 success rate x 1% idea impact x **RM200M** product business **= RM10.4M return**	2% time recovered x 12 experiments/year x 8 lines of business x 1/3 success rate x 1% idea impact x **RM200M** product business **= RM3.2M return**	2% time recovered x 2 experiments/year x 8 lines of business x 1/3 success rate x 1% idea impact x **RM200M** product business **= RM533K return**
CPPA (67 technical staff)	1% time recovered x 730 experiments/year x 1 lines of business x 1/3 success rate x 1% idea impact x **RM50M** product business **= RM243K return**	1.5% time recovered x 52 experiments/year x 1 line of business 1/3 success rate x 1% idea impact x **RM50M** product business **= RM26K return**	2% time recovered x 12 experiments/year x 1 line of business 1/3 success rate x 1% idea impact x **RM50M** product business **= RM8K return**	2% time recovered x 2 experiments/year x 1 line of business 1/3 success rate x 1% idea impact x **RM50M** product business **= RM1.3K return**

To calculate Cost of Downtime per Year, we use the following equation:

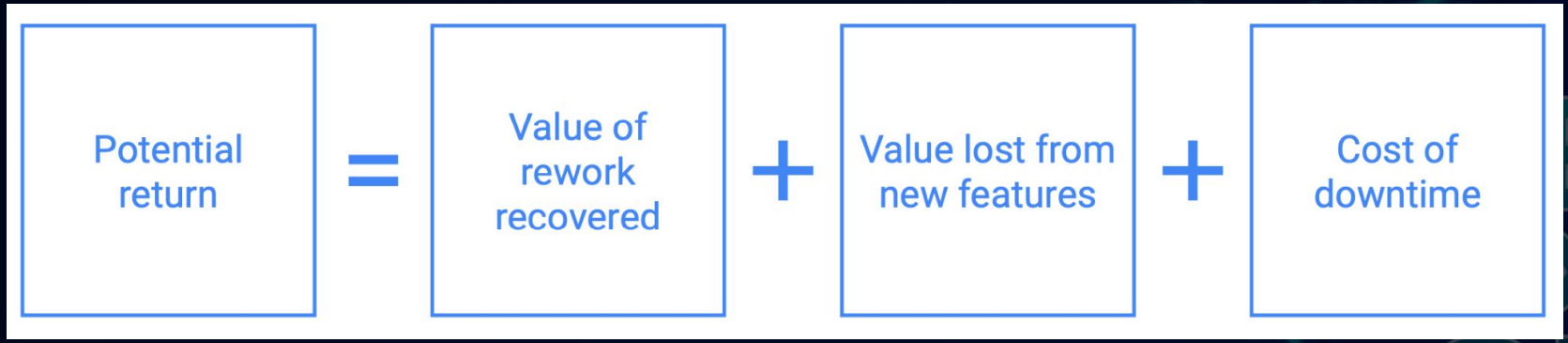
	Elite	High	Medium	Low
Deployment Frequency	**730 per year** (2 deploys per day)	**209 per year** (between once per day and once per week)	**32 per year** (Between once per week and once per month)	**7 per year** (Between once per month and once every six months)
Change fail rate percentage	7.5%	7.5%	7.5%	53%
Mean time to restore (MTTR)	½ hour	4 hours	8 hours	120 hours
Outage cost	RM500,000	RM500,000	RM500,000	RM500,000

Returns Possible from Cost of Downtime Avoided

	Elite	High	Medium	Low
Deployment Frequency	730	209	32	7
Change fail rate percentage	7.5%	7.5%	7.5%	53%
Mean time to restore (MTTR)	½ hour	4 hours	8 hours	120 hours
Outage cost	RM500,000	RM500,000	RM500,000	RM500,000
downtime cost per year	**RM13.7M**	**RM31.5M**	**RM9.6M**	**RM222.6M**
downtime cost per deployment	**RM18.8K**	**RM150K**	**RM300K**	**RM31.8M**

Elliot, S. (2014). DevOps and the Cost of Downtime: Fortune 1000 Best Practice Metrics Quantified. http: info.appdynamics.com/DC-Report-DevOps-and-the-Cost-of-Downtime.html

To calculate **Potential Return per Year**, we use the following equation:



Potential Return (Portfolio RM500M/200M/50M)

	Elite	High	Medium	Low
F (15,143 technical staff)	**RM13.6M** value of rework recovered + **RM243.3M** value lost from new features + **RM13.7M** cost of downtime **= RM270.6M return**	**RM20M** value of rework recovered + **RM26M** value lost from new features + **RM31.4M** cost of downtime **= RM77.4M return**	**RM27.2M** value of rework recovered + **RM8M** value lost from new features + **RM9.6M** cost of downtime **= RM44.8M return**	**RM27.2M** value of rework recovered + **RM1.3M** value lost from new features + **RM222.6M** cost of downtime **= RM251.1M return**
BDTM (174 technical staff)	**RM157K** value of rework recovered + **RM97.3M** value lost from new features + **RM13.7M** cost of downtime **= RM111.2M return**	**RM235K** value of rework recovered + **RM10.4M** value lost from new features + **RM31.4M** cost of downtime **= RM42M return**	**RM313K** value of rework recovered + **RM3.2M** value lost from new features + **RM9.6M** cost of downtime **= RM13.1M return**	**RM313K** value of rework recovered + **RM533K** value lost from new features + **RM222.6M** cost of downtime **= RM223.4M return**
CPPA (67 technical staff)	**RM60K** value of rework recovered + **RM243K** value lost from new features + **RM13.7M** cost of downtime **= RM14.5M return**	**RM90K** value of rework recovered + **RM26K** value lost from new features + **RM31.4M** cost of downtime **= RM32.6M return**	**RM121K** value of rework recovered + **RM8K** value lost from new features + **RM9.6M** cost of downtime **= RM9.8M return**	**RM121K** value of rework recovered + **RM1.3K** value lost from new features + **RM222.6M** cost of downtime **= RM222.8M return**

Demonstrating Return on Investment

Payback Period (BDTM)

Using the potential return of a **BDTM**, in the **Elite category,** we are considering an investment that will cost **RM26.7M** and will generate **RM111.2M per year** in returns

$$\text{Payback Period} = \frac{\text{Investment}}{\text{Return}} = \frac{\text{RM26,723,000}}{\text{RM111,157,000}} = 0.24 \text{ years}$$

The **payback period** is **0.24** years, or about **87 days**, meaning this investment will “pay itself back” in **4** months. In this calculation, faster is better.

DEVOPS Return on Investment BDTM

$$\text{ROI} = \frac{\text{Return - Investment}}{\text{Investment}} = \frac{\text{RM111,157,000 - RM26,723,000}}{\text{RM26,723,000}} = \mathbf{3.12}$$

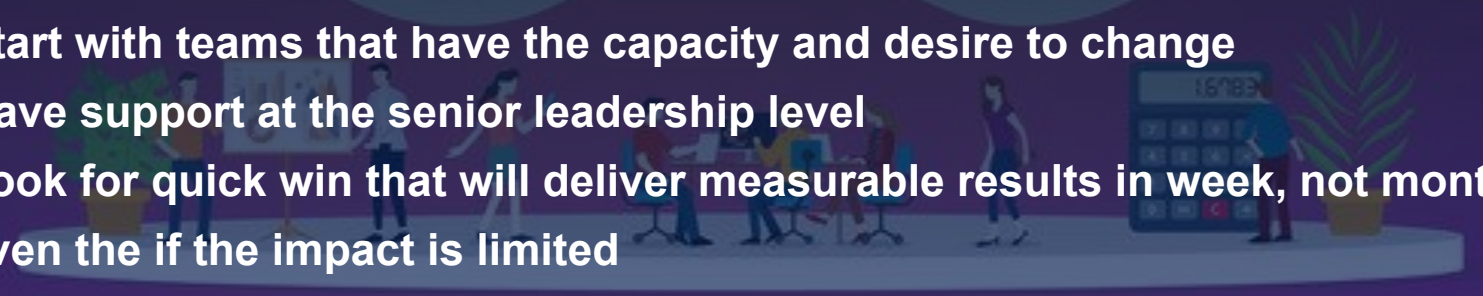
The ROI for this investment (BDTM) is **3.12**

*BDTM made **~RM3.12** for every ringgit*

DevOps Transformation Pays Off

Lesson Learnt

- Working in small batches - Create Read Update Delete (CRUD)
- Deliver by phase
- Implement test suites (don't trust "asal jalan")
- Choose your Scrum Master / Project Manager wisely
- Reduce use of I/O as possible (in-memory)
- Separate runner from main GitLab server
- Start with teams that have the capacity and desire to change
- Have support at the senior leadership level
- Look for quick win that will deliver measurable results in week, not months, even the if the impact is limited
- Be prepared for early disappointments - don't give up



Lesson Learnt (cont.)

- **High quality documentation (READMEs):**
  - **3.8 times more likely to implement security practice**
  - **2.4 times more likely to meet or exceed reliability target**
  - **3.5 more likely to implement Site Reliability Engineer**
  - **3.5 times more likely to fully leverage on cloud**

- Aghajani, E. et al. (2019). Software Documentation Issues Unveiled. Proceedings of the 2019, IEEE/ACM 41st International Conference on Software Engineering, 1199-1210. https://doi.org/10.1109/ICSE.2019.00122
- Plösch, R., Dautovic, A., & Saft, M. (2014). The Value of Software Documentation Quality, Proceedings of the International Conference on Quality Software, 333-342. https://doi.org/10.1109/QSIC.2014.22
- Zhi, J. et al. (2015). Cost benefits and quality of software development documentation: A systematic mapping. Journal of Systems and Software, 99(C), 175-198. https://doi.org/10.1016/j.jss.2014.09.042

Moving Forward

- Containerization (kubernetes) - scaling up / down automagically
- Rancher / Spinnaker - UI Cloud (private) Management
- Cloud-native architecture and technologies

References

1. Kim, G. (n.d.) The Amazing DevOps Transformation of The HP LaserJet Firmware Team (Gary Gruver). https://itrevolution.com/the-amazing-devops-transformation-of-the-hp-laserjet-firmware-team-gary-gruver/
2. DevOps Research and Assessment & Google Cloud. (n.d.).2019 Accelerate: State of DevOps Report (Rep.)
3. DevOps Research and Assessment & Google Cloud. (n.d.).2020 Accelerate: State of DevOps Report (Rep.)
4. DevOps Enterprise Summit 2014. (2014, October 29). DOES14 – Courtney Kissler – Nordstrom – Transforming to a Culture of Continuous Improvement. https://www.youtube.com/watch?v=0ZAcsrZBSlo
5. Earnshaw, A. (2013, July 18). DevOps Solves Business Problems: Gene Kim's Top Aha Moments. https: puppet.com/blog/devops-solves-business-problems-gene-kim%E2%80%99s-top-aha-moments
6. Divine, C. (2011, April 20). Leadership in an Agile Age: An Interview With Scott Cook. Retrieved from https://web.archive.org web/20160205050418/http://network.intuit.com/2011/04/20/leadership-in-the-agile-age/
7. Ippolito, B., & Murman, E. (2001, December). Improving the Software Upgrade Value Stream. In 43rd AIAA Aerospace Sciences Meeting and Exhibit (p. 1252).

References

8. Chron 200 / Interview with CEO of the Year Charles Schwab. (2007, April 9). Retrieved from http://www.sfgate com/business/article/Chron-200-Interview-with-CEO-of-the-Year-2603664.phphttp://dspace.mit.edu/bitstream handle/1721.1/83541/REP_0101_Ippo.pdf?sequence=1

9. Hainzinger, Brittany. “DevOps Salary Report for 2019 Is Here.” App Developer Magazine, 22 Jan. 2019, appdevelopermagazine.com/devops-salary-report-for-2019-is-here/.

10. Morozoff, E. (2009, September 4). Using a Line of Code Metric to Understand Software Rework. Retrieved from http: ieeexplore.ieee.org/document/5232799/

11. Kohavi, R., Crook, T., Longbotham, R., Frasca, B., Henne, R., Ferres, J., Melamed, T. (2009). Online Experimentation at Microsoft. Retrieved from http://ai.stanford.edu/~ronnyk/ExPThinkWeek2009Public.pdf

12. Kohavi, R., Crook, T., Longbotham, R., Frasca, B., Henne, R., Ferres, J., Melamed, T. (2009). Online Experimentation at Microsoft. Retrieved from http://ai.stanford.edu/~ronnyk/ExPThinkWeek2009Public.pdf

13. Elliot, S. (2014). DevOps and the Cost of Downtime: Fortune 1000 Best Practice Metrics Quantified. Retrieved from http: info.appdynamics.com/DC-Report-DevOps-and-the-Cost-of-Downtime.html

References

14. Shimel, A. (2015, February 11). The real cost of downtime. Retrieved from http://devops.com/2015/02/11/real-cost downtime/.

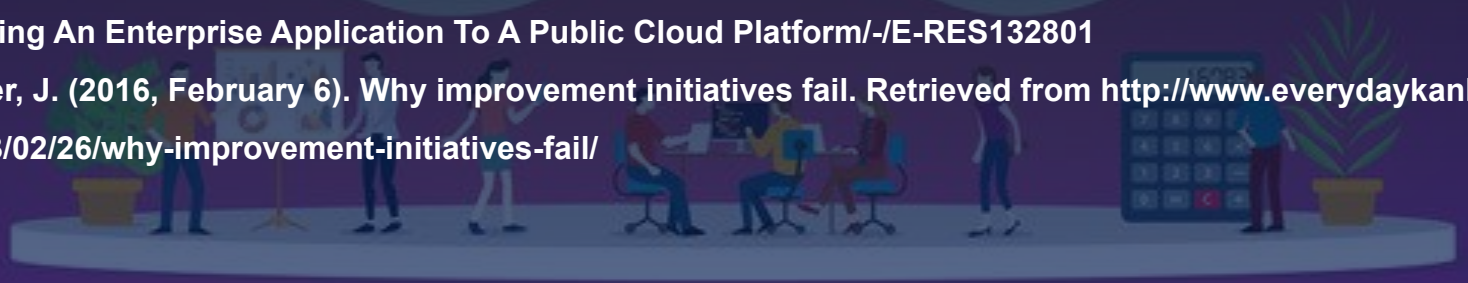
15. DevOps Research and Assessment, LLC. (n.d.). 2019 Accelerate: State of DevOps Report (Rep.)

16. Reichheld, F. F. (2003, December). The One Number You Need to Grow. Retrieved from https://hbr.org/2003/12/the-one number-you-need-to-grow

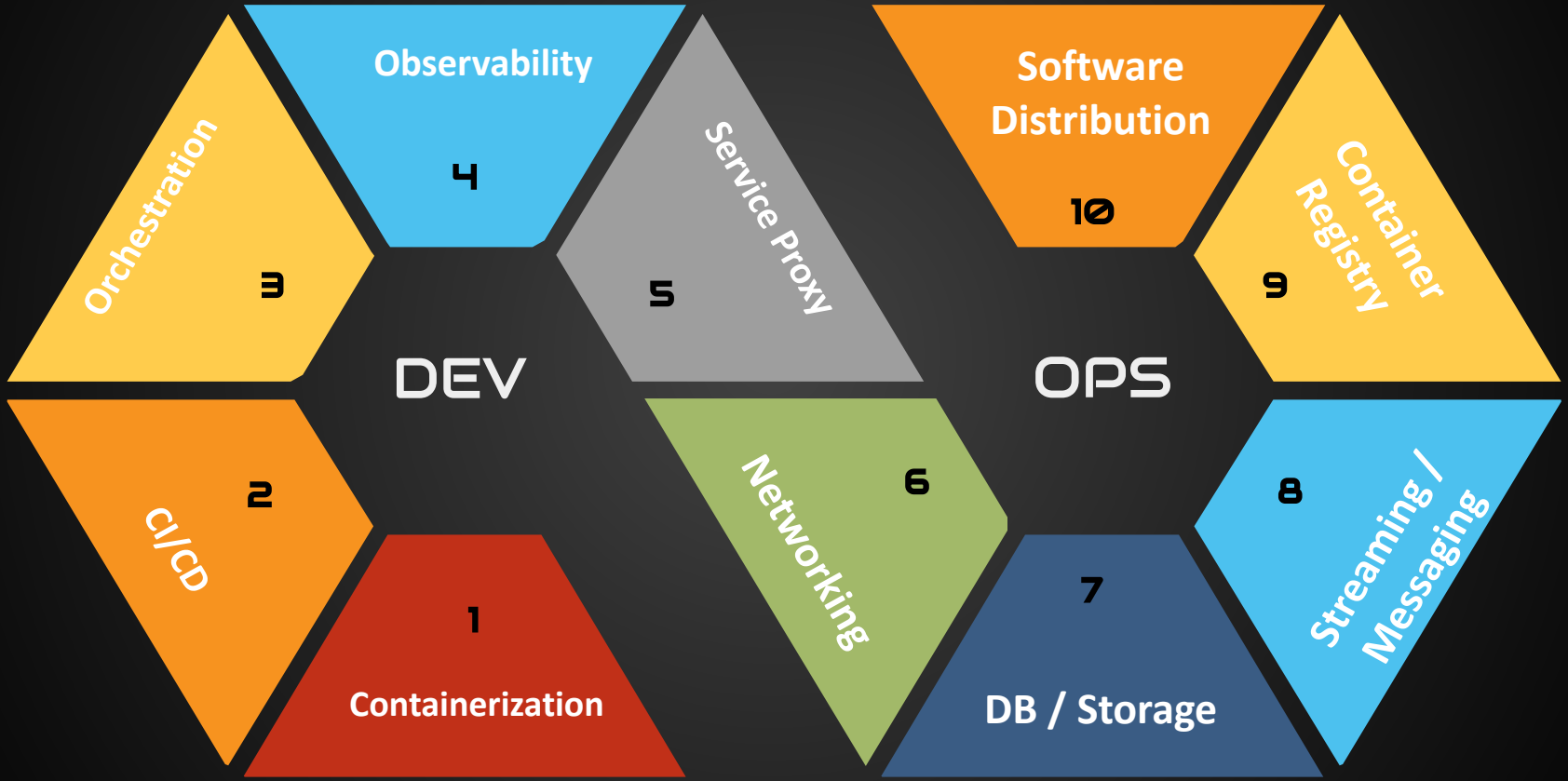
17. Patterson, D. (2002, Nov 3-8, 2002). A Simple Way to Estimate the Cost of Downtime. Paper presented at the Large Installation System Administrator's Conference (LISA '02), Philadelphia, PA

18. Rymer, J. R., Bartoletti, D, Martorelli, B, Mines, C, Tajima, C. (2016, March 9). Brief: The Cost Of Migrating An Enterprise Application To A Public Cloud Platform. Retrieved from https://www.forrester.com/report/Brief The Cost Of Migrating An Enterprise Application To A Public Cloud Platform/-/E-RES132801

19. Wester, J. (2016, February 6). Why improvement initiatives fail. Retrieved from http://www.everydaykanban com/2013/02/26/why-improvement-initiatives-fail/



Cloud Native References



https://landscape.cncf.io

Sekian,
Terima Kasih

Ts. Hariadi bin Hinta
hariadi@jpa.gov.my