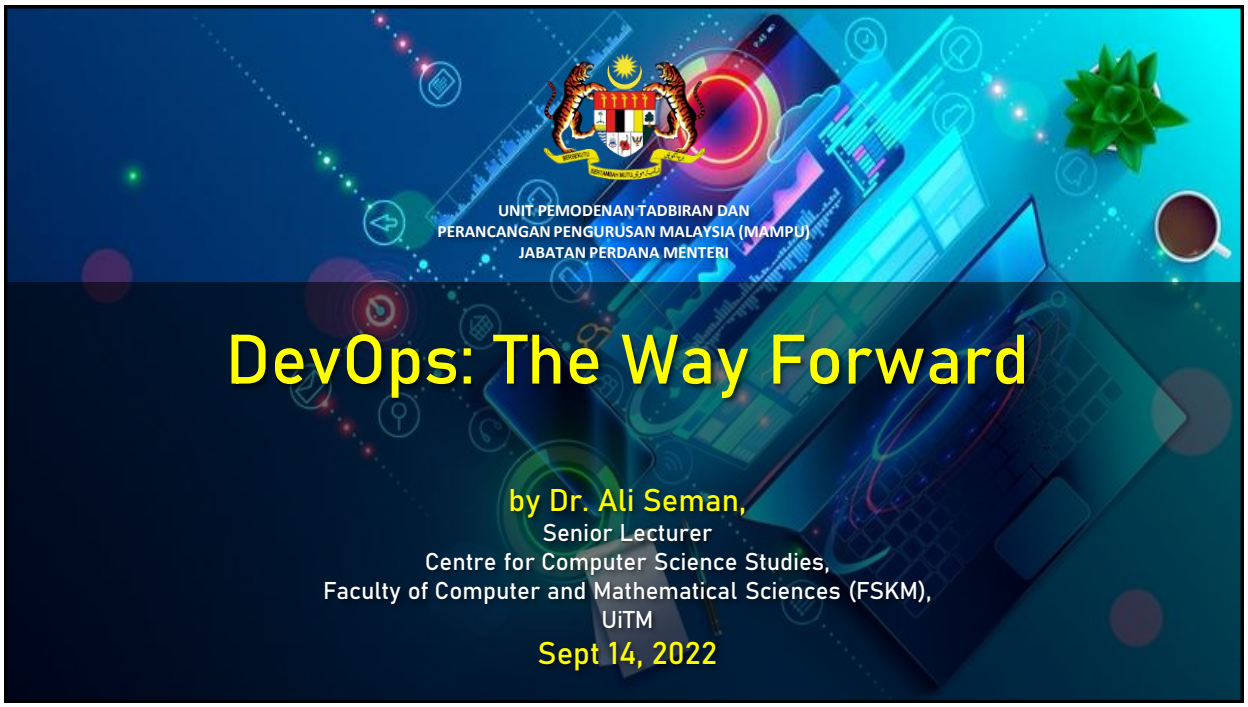


DevOps: The Way Forward

by Dr. Ali Seman,
Senior Lecturer
Centre for Computer Science Studies,
Faculty of Computer and Mathematical Sciences (FSKM),
UiTM

Sept 14, 2022

OUTLINE

- BACKGROUND
- SDLC MODELS – PRINCIPLES AND ISSUES
- DEVOPS: PROS AND CONS
- DEVOPS: THE WAY FORWARD
- FINAL THOUGHTS

OUTLINE

- BACKGROUND
- SDLC MODELS – PRINCIPLES AND ISSUES
- DEVOPS: PROS AND CONS
- DEVOPS: THE WAY FORWARD
- FINAL THOUGHTS

BACKGROUND

- **Software Crisis and Lesson Learned From Denver International Airport (DIA) Baggage Handling System**
  - ✓ **Quality:** Performed just a fraction of its original automation goals (Only one concourse for one airline)
  - ✓ **Past The Deadline:** 16 months past the 2-year deadline
  - ✓ **Over budget:** Cost about the extra $560 million (Original budget: $238 million.)

➢ BACKGROUND (Cont.)

❖ Software Development Goals

- ✓ Meeting user requirements (High Quality)
- ✓ Delivery - On Time (Fast development)

❖ Software Owner Goals

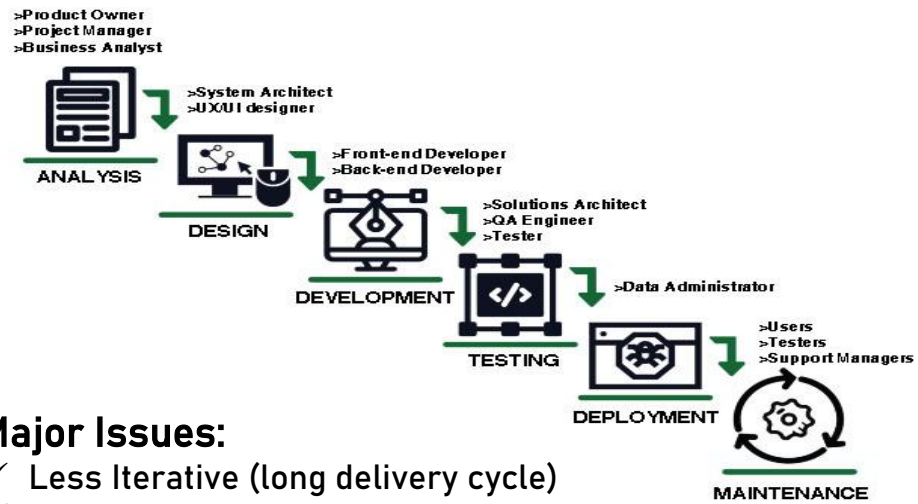
- ✓ Reduce Cost/ ROI
- ✓ Not just computerized BUT automated system

➢ BACKGROUND (Cont.)

OUTLINE

- BACKGROUND
- SDLC MODELS – PRINCIPLES AND ISSUES
- DEVOPS: PROS AND CONS
- DEVOPS: THE WAY FORWARD
- FINAL THOUGHTS

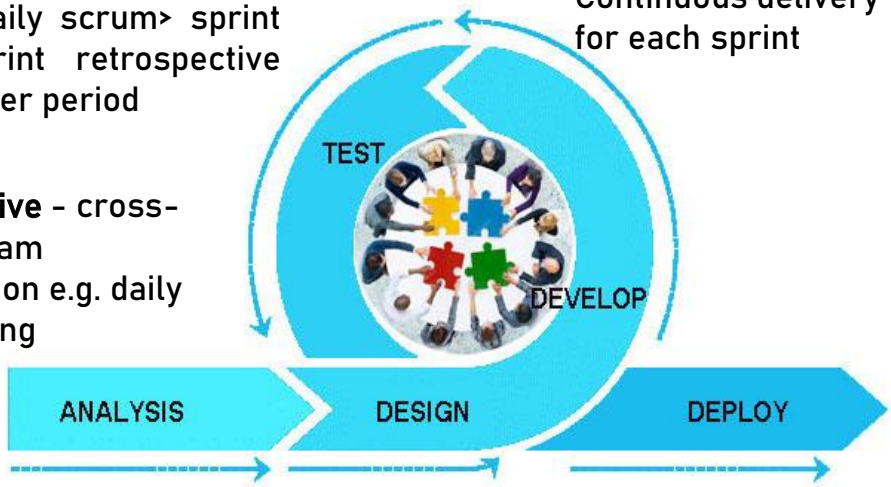
LINEAR (SEQUENTIAL) MODEL (e.g. Waterfall Model)



- **Major Issues:**
  - ✓ Less Iterative (long delivery cycle)
  - ✓ Less collaborative

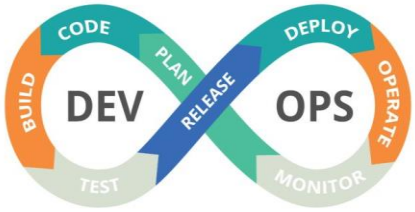
➢ ITERATIVE, COLLABORATIVE & COMMUNICATIVE MODEL (Agile)

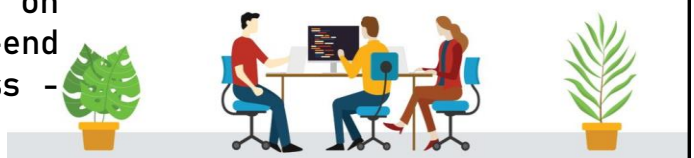
- **Iterative** called Sprint >sprint planning> daily scrum> sprint review> sprint retrospective within a sorter period
- **Communicative** - cross-functional team communication e.g. daily scrum meeting
- **Collaborative** - Continuous delivery for each sprint



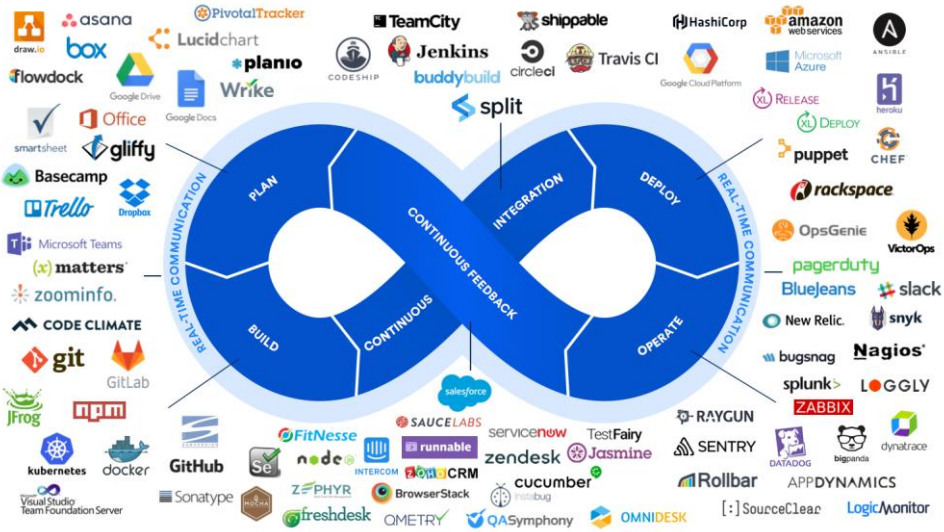
➢ ITERATIVE, COLLABORATIVE & COMMUNICATIVE MODEL (DevOps)

- **Iterative** (Infinity loop >build> test> release> deploy> operate> monitor> plan> code.
- **Collaborative & Communicative**– Focuses on uniting the Dev(elopement) and Op(eration)s teams
- **The plus point:** emphasize on automation (tools) - end-to-end software engineering process - CI/CD pipeline





- **AUTOMATION TOOLS** (End-To-End Software Engineering Process)

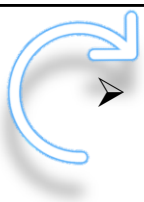


- **AUTOMATION TOOLS**

1) **Version Control tools** (GitHub, Bitbucket, GitLab)
2) **Container Management tools** (Docker <OS>, Kubernetes <OS>, Mesos)
3) **Application Performance Monitoring tools** (Prometheus <OS>, Dynatrace, AppDynamics)
4) **Deployment & Server Monitoring tools** (Splunk, Datadog, Sensu <OS>)
5) **Configuration Management tools** (Chef <OS>, Puppet, Ansible<OS>)
6) **CI / Deployment Automation tools** (Bamboo, Jenkins <OS>, IBM UrbanCode)
7) **Test Automation tools** (Test.ai, Ranorex, Selenium)
8) **Artifact Management tools** (Sonatype NEXUS, JFRog Artifactory, CloudRepo)
9) **Codeless Test Automation tools** (AccelQ, Appvance, Testim.io)

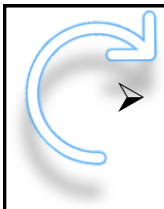
OUTLINE

- BACKGROUND
- SDLC MODELS – PRINCIPLES AND ISSUES
- **DEVOPS: PROS AND CONS**
- DEVOPS: THE WAY FORWARD
- FINAL THOUGHTS



DEVOPS: PROS <1>

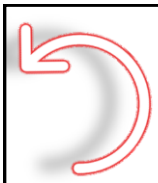
- **Achieve Software Development Goals**
  - ✓ Meeting user requirements (High Quality)
  - ✓ Delivery - On Time (Fast development)



➢ DEVOPS: PROS <2>

❖ Fast Learning Curve

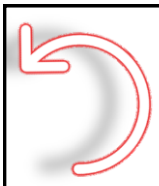
- ✓ Collaborative & Communicative Approach
- ✓ Automation tools – simplified the manual process



➢ DEVOPS: CONS <1>

❖ High cost and low ROI

- ✓ First Time Setup Cost
- ✓ Training and Technical Supports



➢ DEVOPS: CONS <2>

- ❖ **High Dependency** e.g. Tools, products, brands, etc.
  - ✓ Proprietary e.g. licensing
  - ✓ Open Source e.g. technical supports

OUTLINE

- ❖ BACKGROUND
- ❖ SDLC MODELS – PRINCIPLES AND ISSUES
- ❖ DEVOPS: PROS AND CONS
- ❖ **DEVOPS: THE WAY FORWARD**
- ❖ FINAL THOUGHTS

[1] No choice (SDG) but DevOps (Adapt it fully!)

- ✓ DevOps culture and mindset

- ✓ Collaborative & Communicative across agencies

- ✓ Continuous improvement – CI/CD pipeline
- ✓ Automation (Software development vs Application)

[2] Unlocking The Tools (Reduce dependency)

- ✓ Proprietary vs Open Source
- ✓ Competency e.g. Contributor to OS

OUTLINE

- BACKGROUND
- SDLC MODELS – PRINCIPLES AND ISSUES
- DEVOPS: PROS AND CONS
- DEVOPS: THE WAY FORWARD
- **FINAL THOUGHTS**

- **Point 1 –** DevOps is "supposedly not a something new!"
  - **Point 2 –** It is timely for continuous improvement – CI/CD pipeline
    - **Point 3 –** Unlocking the potential – Open Source!
- **Point 4 –** Need to go for a full-blown DevOps approach!