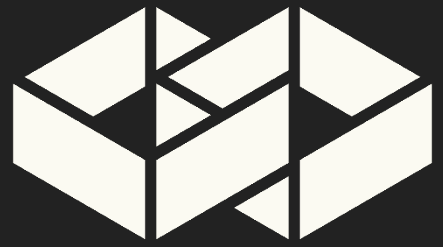
Transforming to DevOps, CJ/CD: Continuous Journey Continuous Disaster?



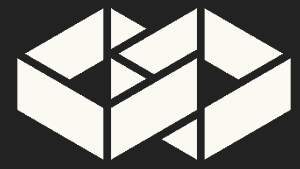
promicro

Data left unattended

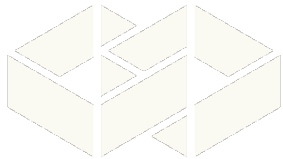
Tonie Huizer

DevOps consultant

Promicro



Database DevOps – the challenge



Tonie Huizer

DevOps consultant

Promicro

linkedin.com/in/toniehuizer

@promicroNL

github.com/promicroNL/events

www.promicro.nl

I like to read, experiment, talk and write about software and the software development process.

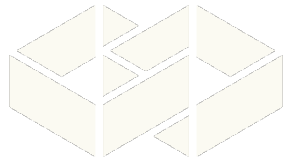
The fun I experience in my job is the combination of people and technology.

Focus areas in my working life:

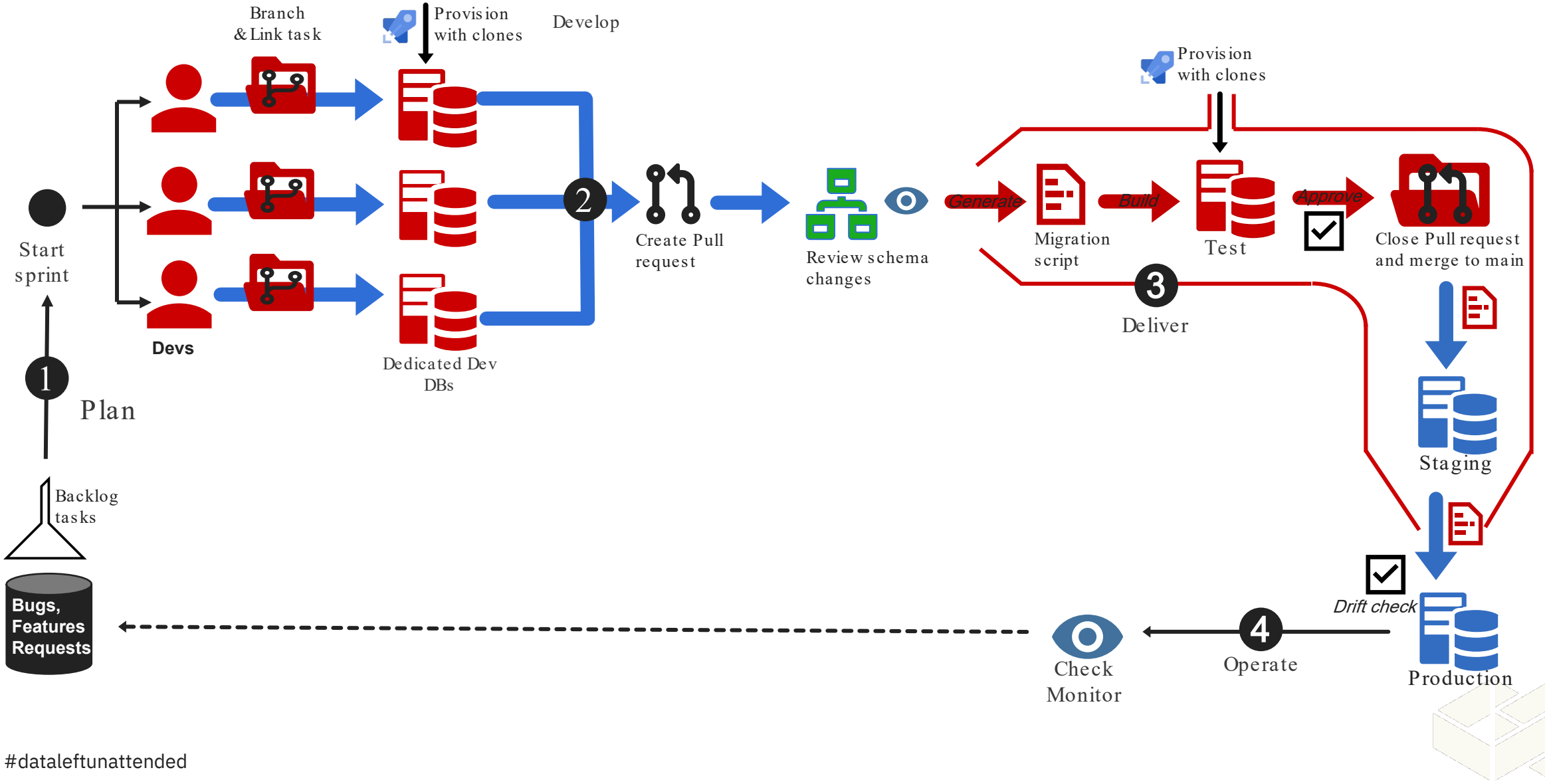
- Software development
- (Database) DevOps
- Whisky

What to expect of todays session

- Transforming to DevOps, CJ/CD?
- The 4 DevOps phases walk through
- Room for discussion & questions

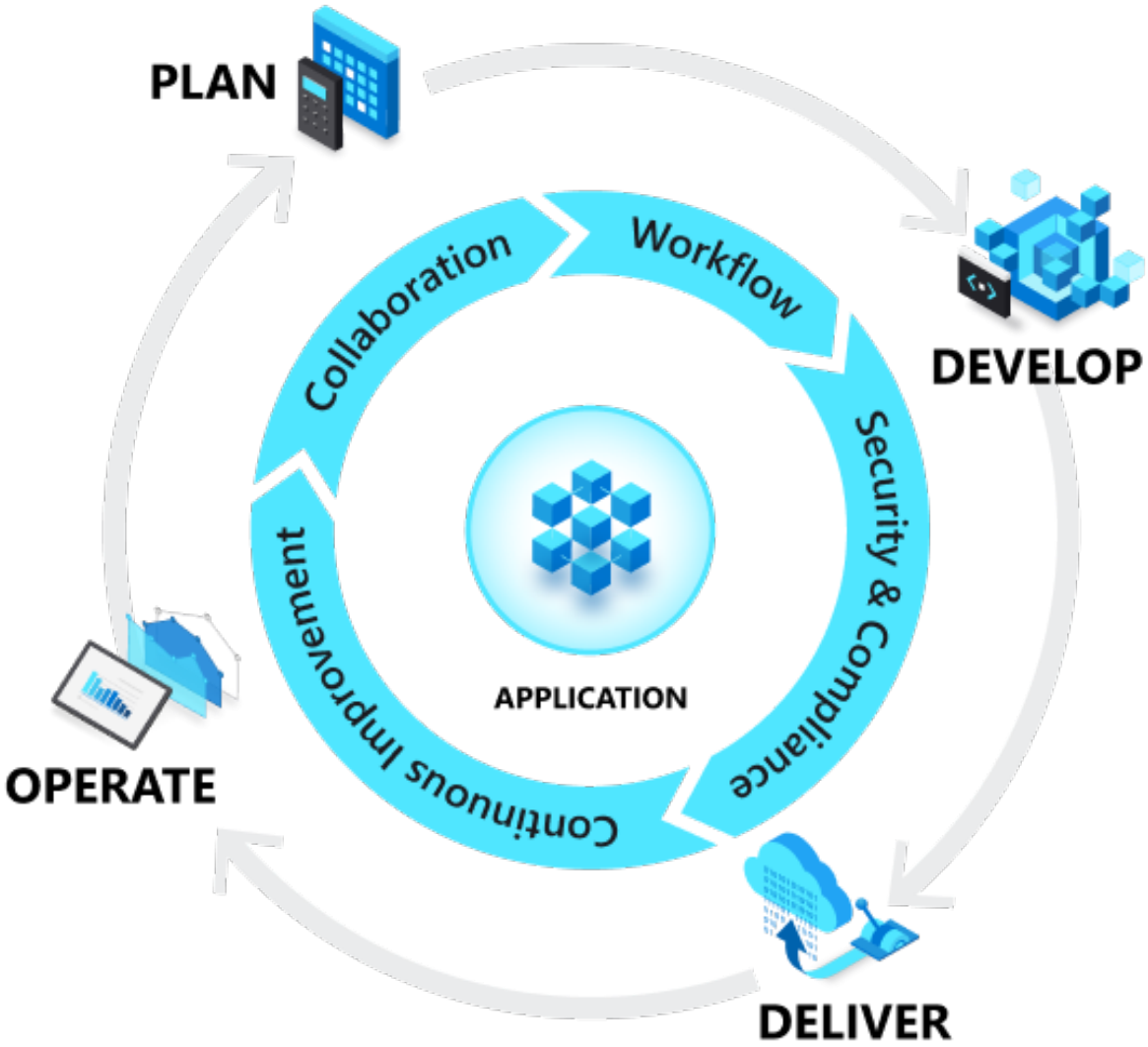


Database DevOps – the end goal



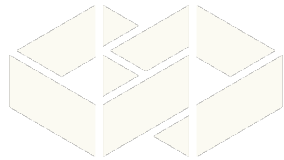
#dataleftunattended

DevOps



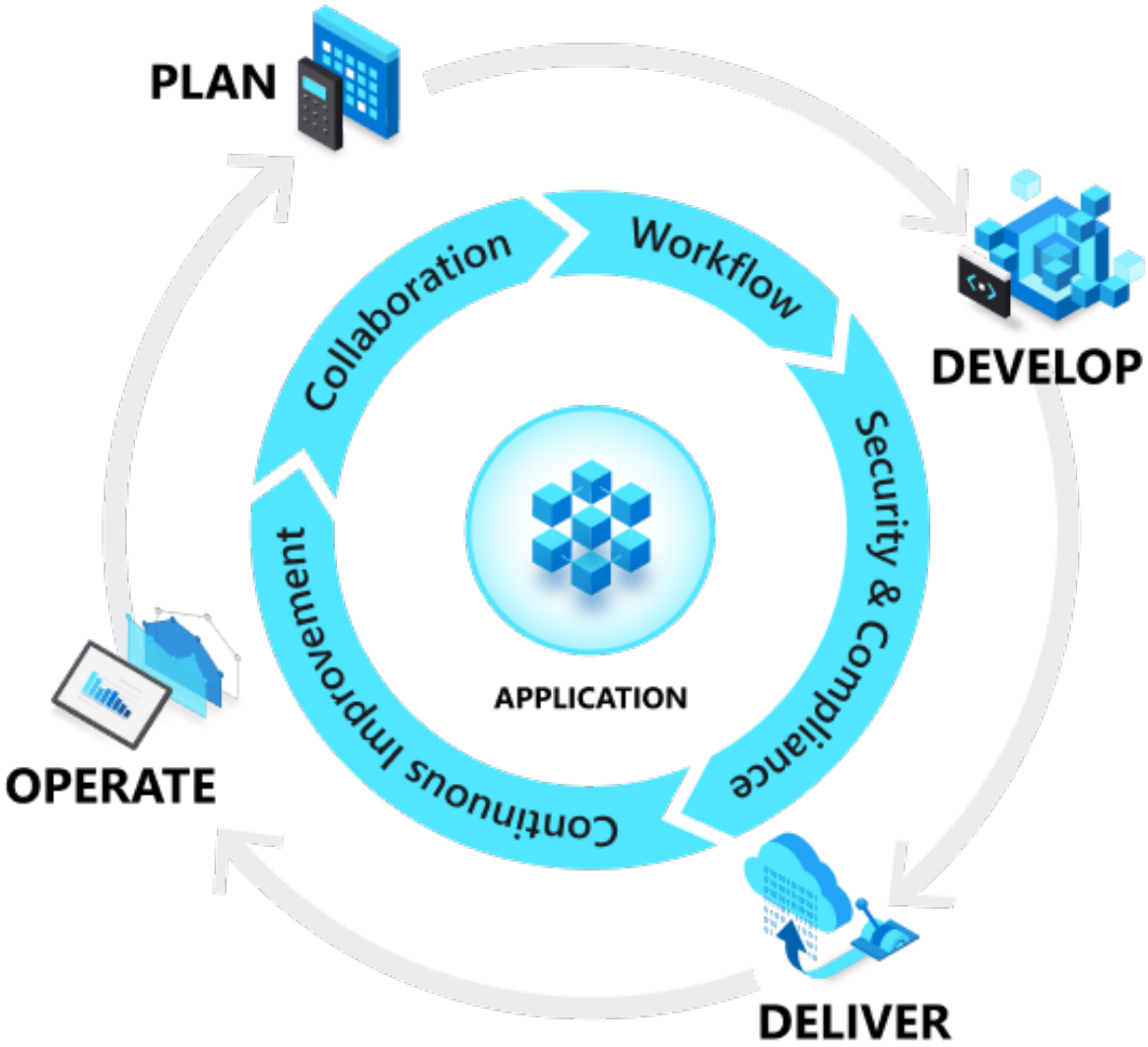
“DevOps is the union of people, process, and products to enable continuous delivery of value to our end users.”

- Donovan Brown, Microsoft



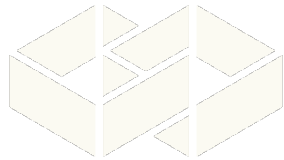

Teams
Physical team rooms
Cross discipline
10-12 people
Self managing
Clear charter and goals
Intact for 12-18 months
Own features in production
Own deployment of features
TECHORAMA

DevOps



Plan DevOps practices

- Create backlogs
- Use Kanban boards
- Visualize progress with dashboards
- Manage Agile software development with Scrum



Attempts to plan with boards & sprints

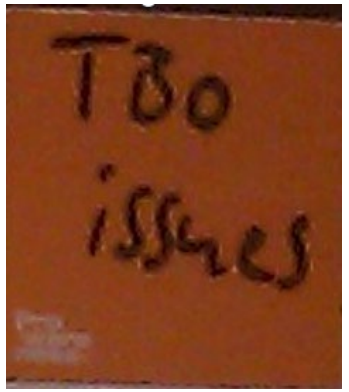
#dataleftunattended

Working in sprints didn’t work

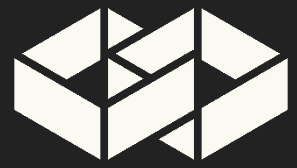
Why?

- Team was too big
- No fixed iteration path

- Little involvement of the stakeholders
- Unrefined backlog items
- No links or integration



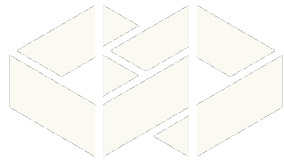
Demo



This time it worked

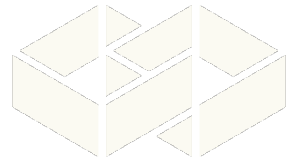
For 142 sprints in a row (and counting)

- ✓ Team split up per end-customer
- ✓ Focus and involvement of the stakeholder
- ✓ Training in scrum methodology
- ✓ Better integration of tools
- ✓ Mandatory linking development to work

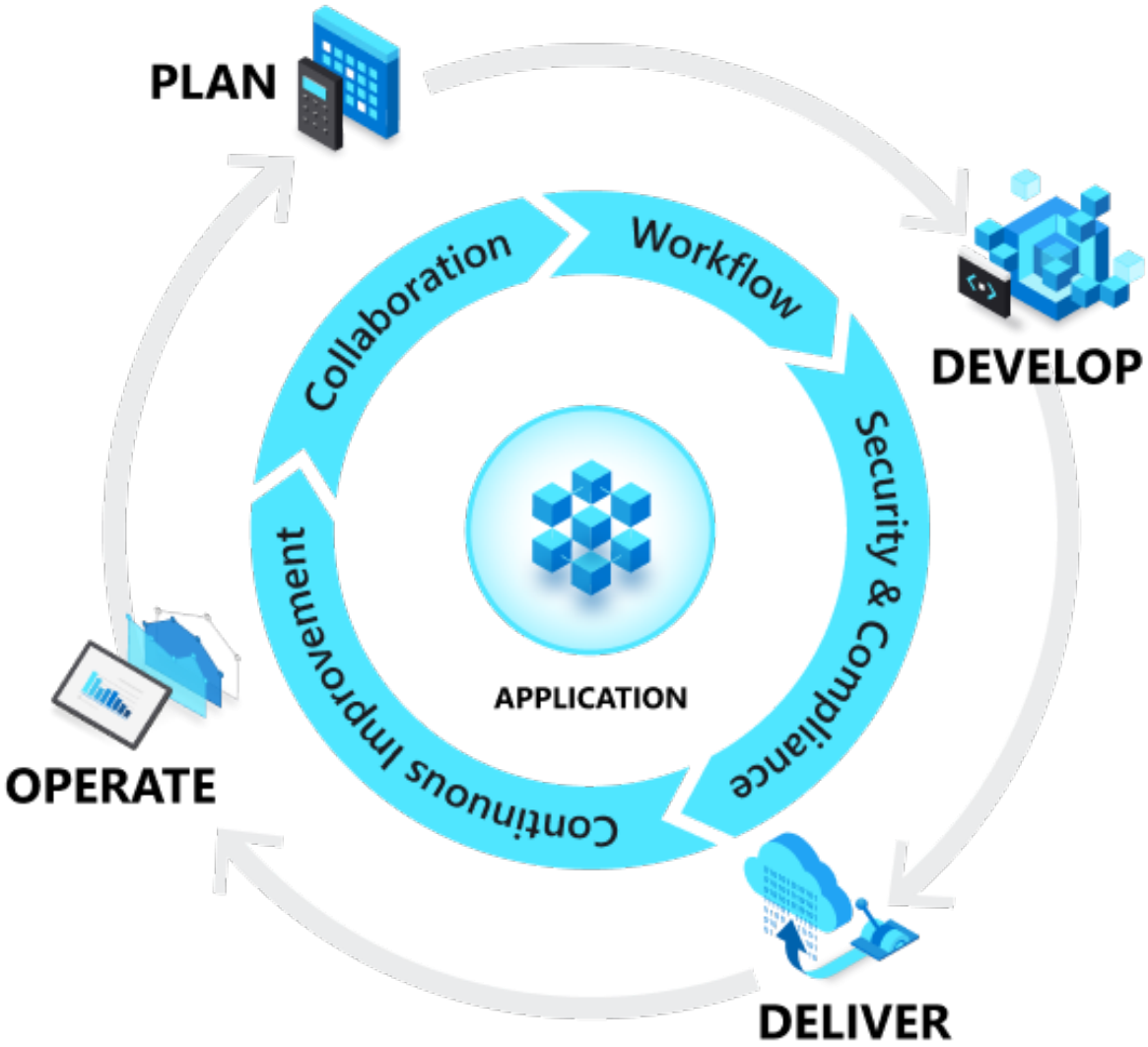


Plan DevOps practices

- ✓ Create backlogs
- ✓ Use Kanban boards
- ✓ Visualize progress with dashboards
- ✓ Manage Agile software development with Scrum

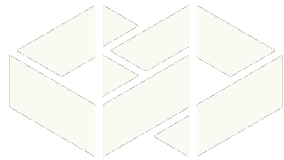


DevOps

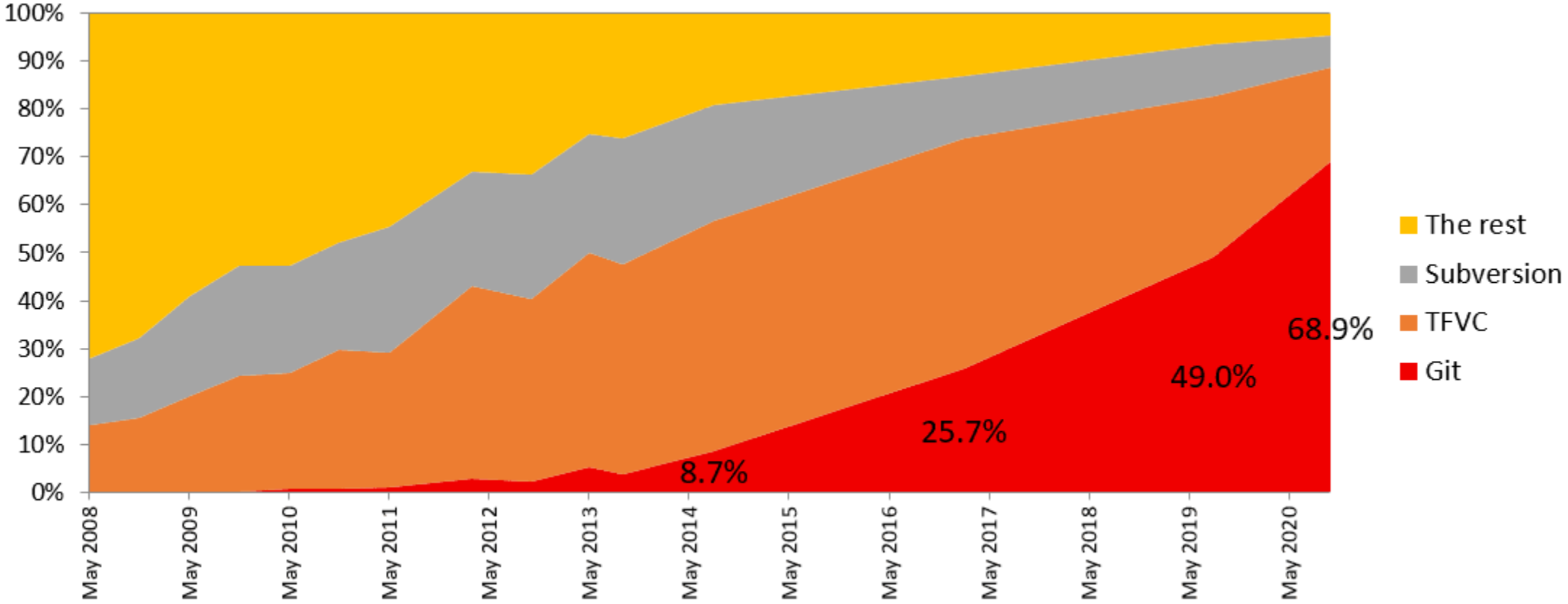


Develop DevOps practices

- Choose a VCS to collaborate and work in parallel
- Automate repetitive tasks
- Turn code into immutable artifacts

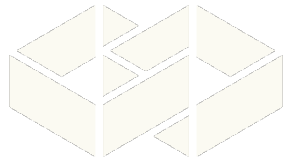


SVN and TFVC didn’t do the job

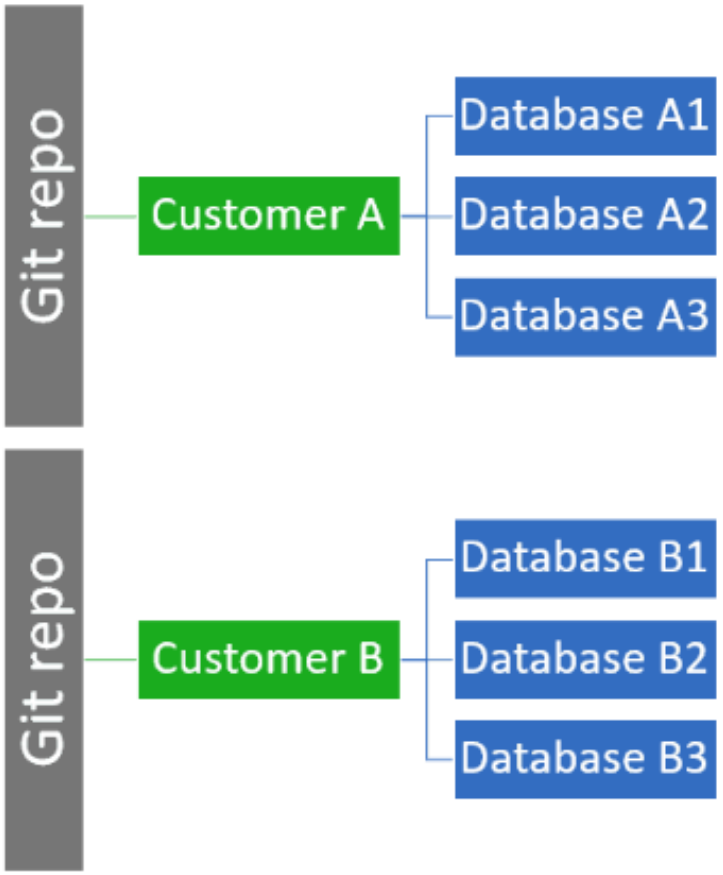


Switching to git

- Migrate a VCS
- Adopt a branching strategy
- Create a branch naming convention



Application source code

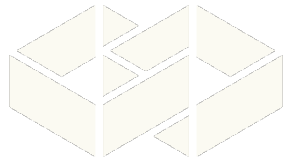


Database source code

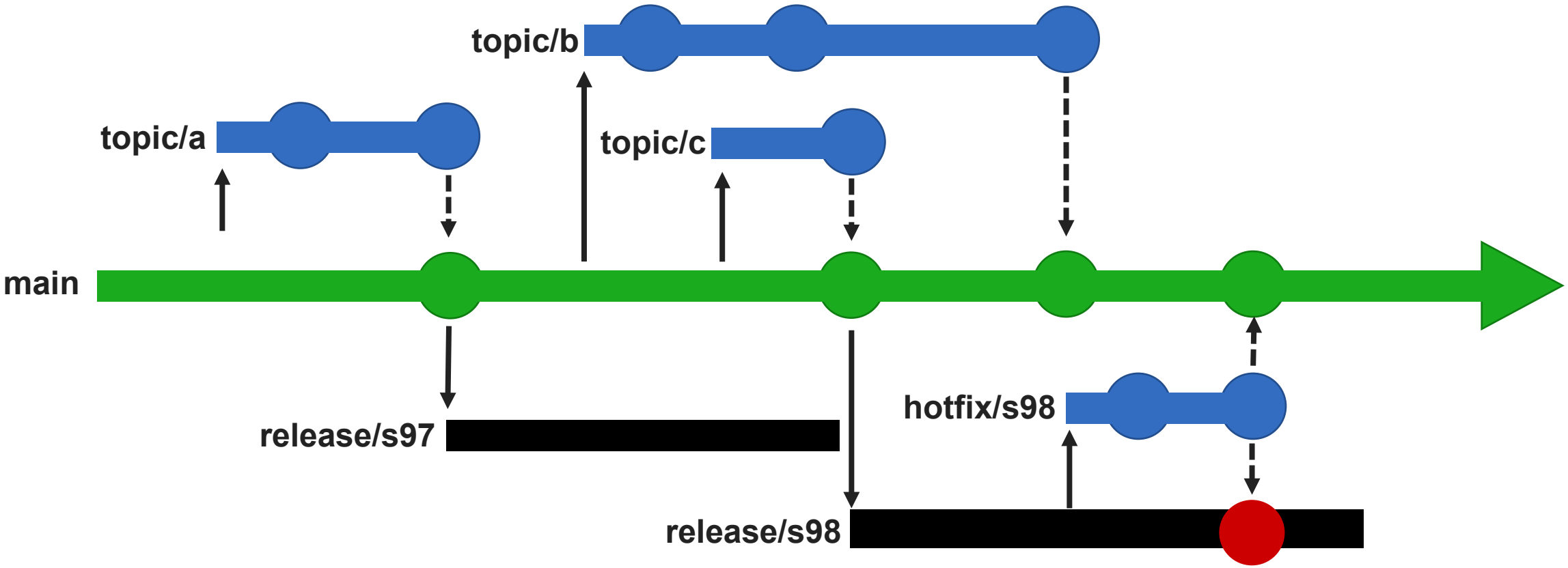
#dataleftunattended

Branching strategy

- Release flow
  - Short living topics
  - Main branch always in release state
  - Hotfix the release, cherry pick main



Release flow in action



Create a branch naming convention

Topic / hotfix

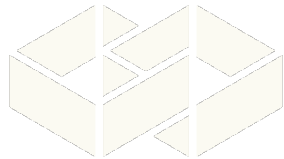
<branch category>/<hot fixed release - >bug<TicketId>-PascalCasingDescription

hotfix/s100-mms12220-FireFighting
topic/mms12345-MyDescription

Release branch

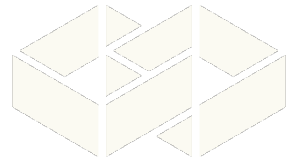
<branch category>/<unique identification>

release/s100



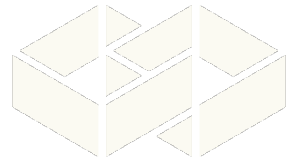
Git compared to SVN / TFVC

- ✓ One repo per solution
- ✓ Parallel work with branching
- ✓ Less time managing version control
- ✓ Always visible what is released

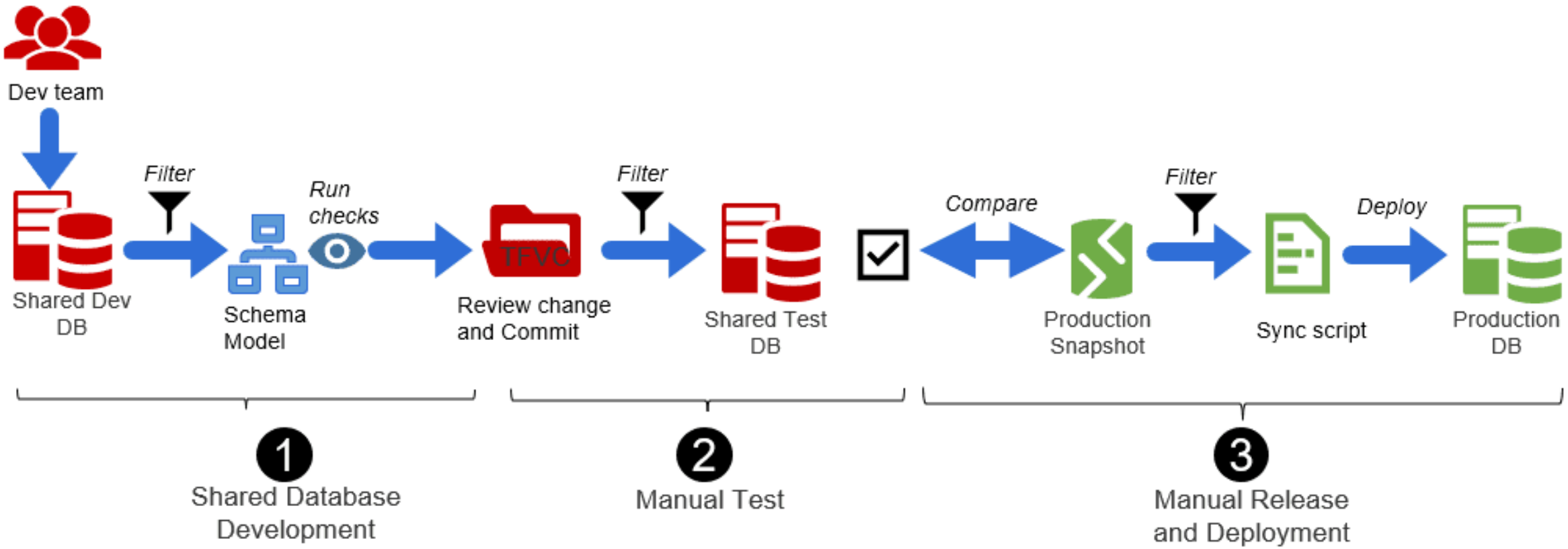


Develop DevOps practices

- ✓ Choose a VCS to collaborate and work in parallel
- Automate repetitive tasks
- Turn code into immutable build artifacts

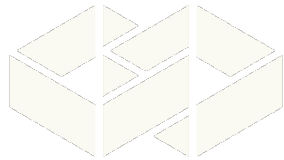


Manual, repetitive tasks

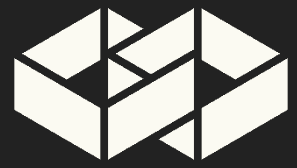


Automate repetitive tasks

- Standardize the process
- Use pipelines, but first...
  - Start local to automate
  - Use verbose logging

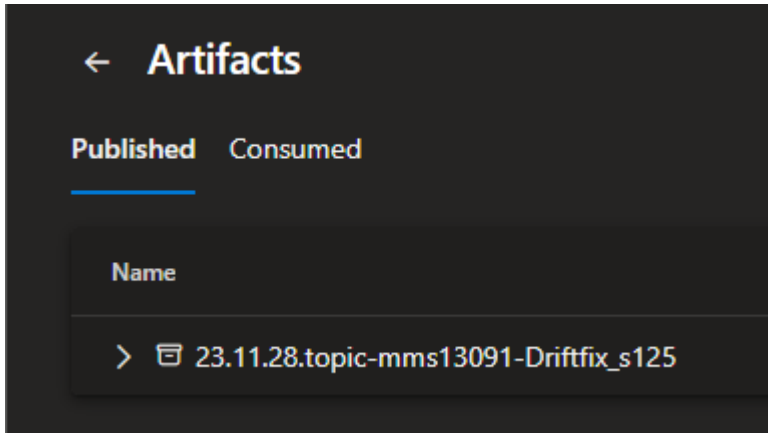


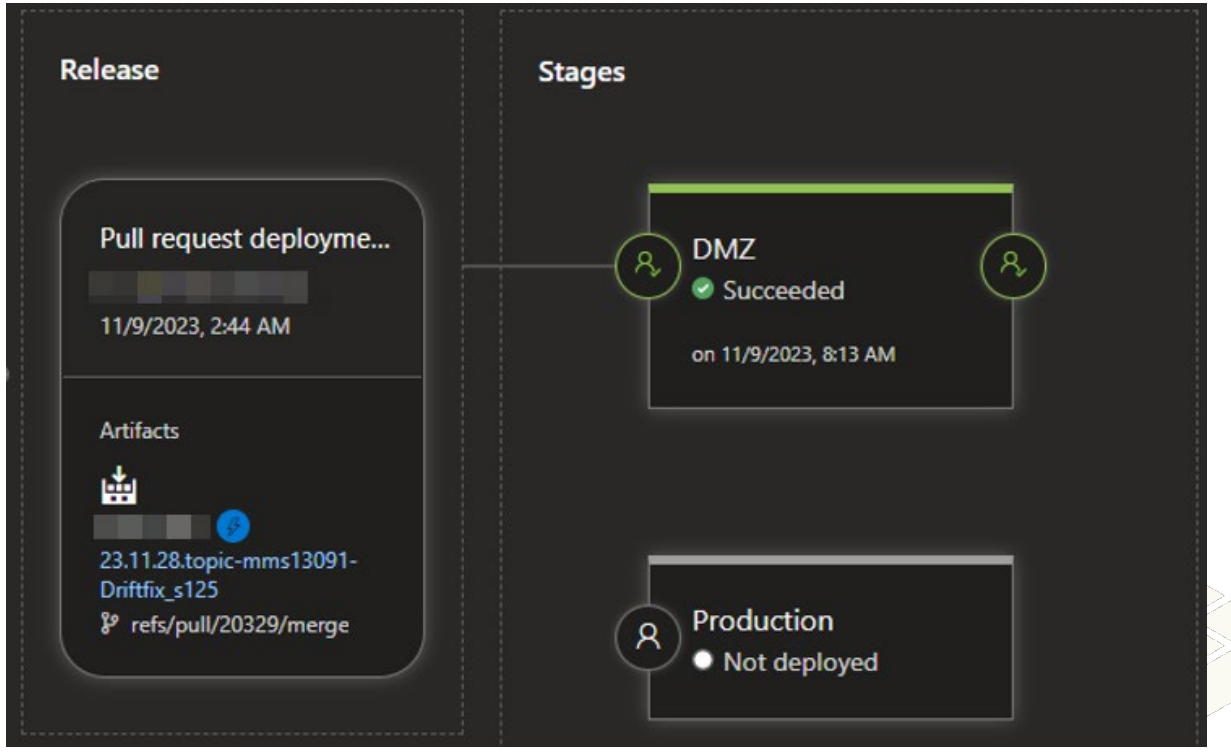
Demo



Build artifact & automate repetitive tasks

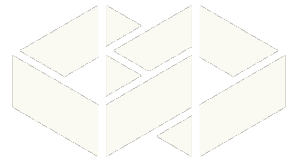
- ✓ Pipeline to generate immutable artifact
- ✓ Apply same artifact to all environments



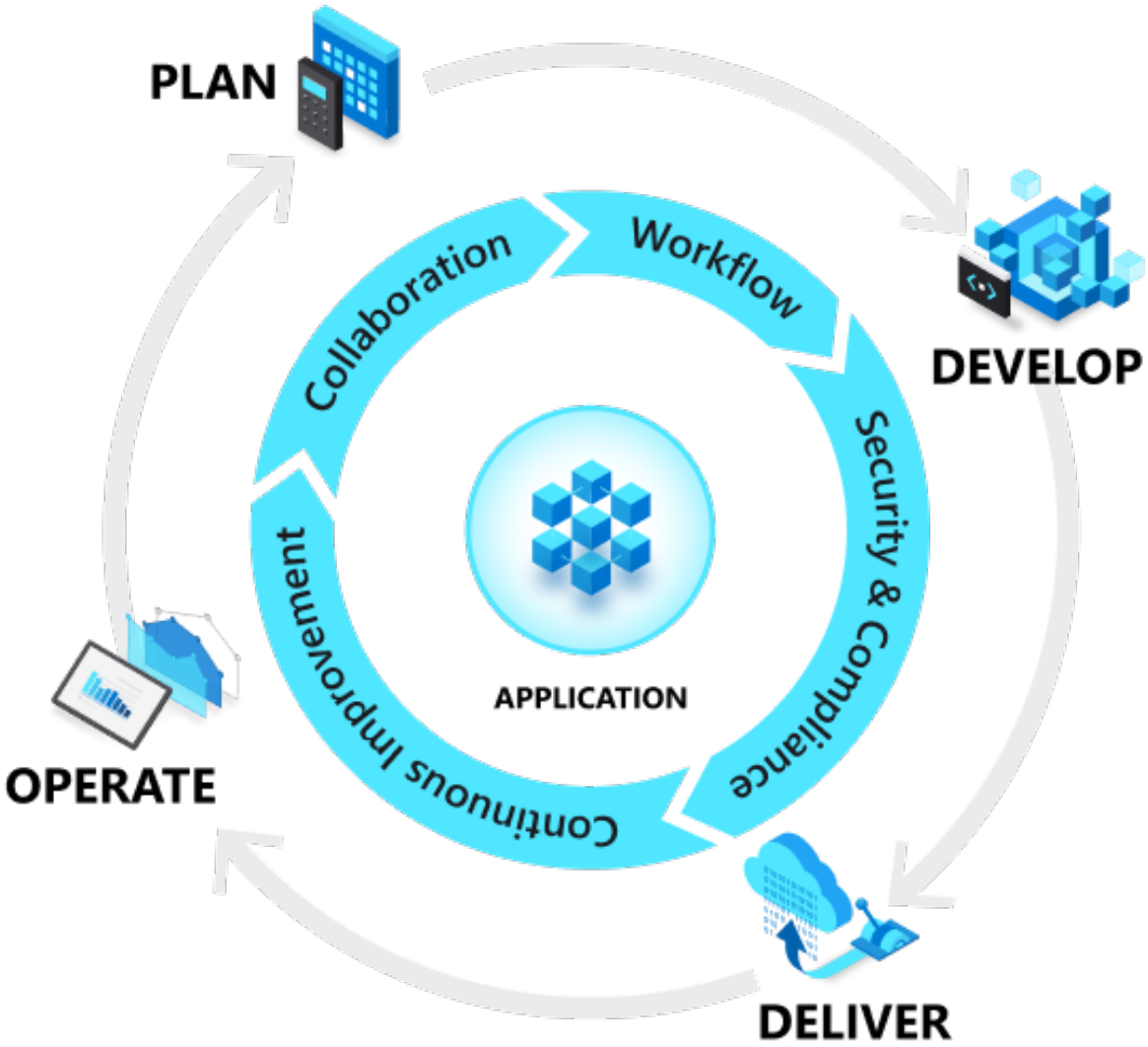


Develop DevOps practices

- ✓ Choose a VCS to collaborate and work in parallel
- ✓ Turn code into immutable build artifacts
- ✓ Automate repetitive tasks

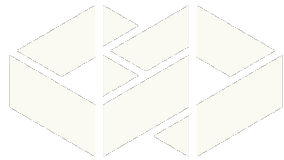


DevOps



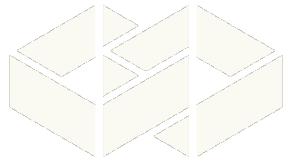
Deliver DevOps practices

- Automate delivery processes
- Use release pipelines with approvals
- Release artifacts to different environments



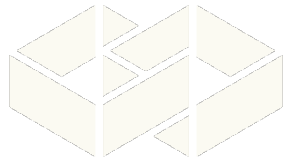
Previous deliver workflow

- Copy build from share
- Manual deploy database changes
- Manual alter config
- Informal approval
- Shared environment & database



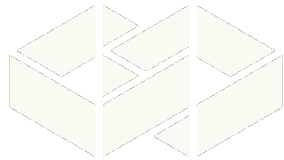
Deliver workflow improvements

- Test in parallel, just like development
- Make use of disposable environments
- Implement Pull Request Release workflow



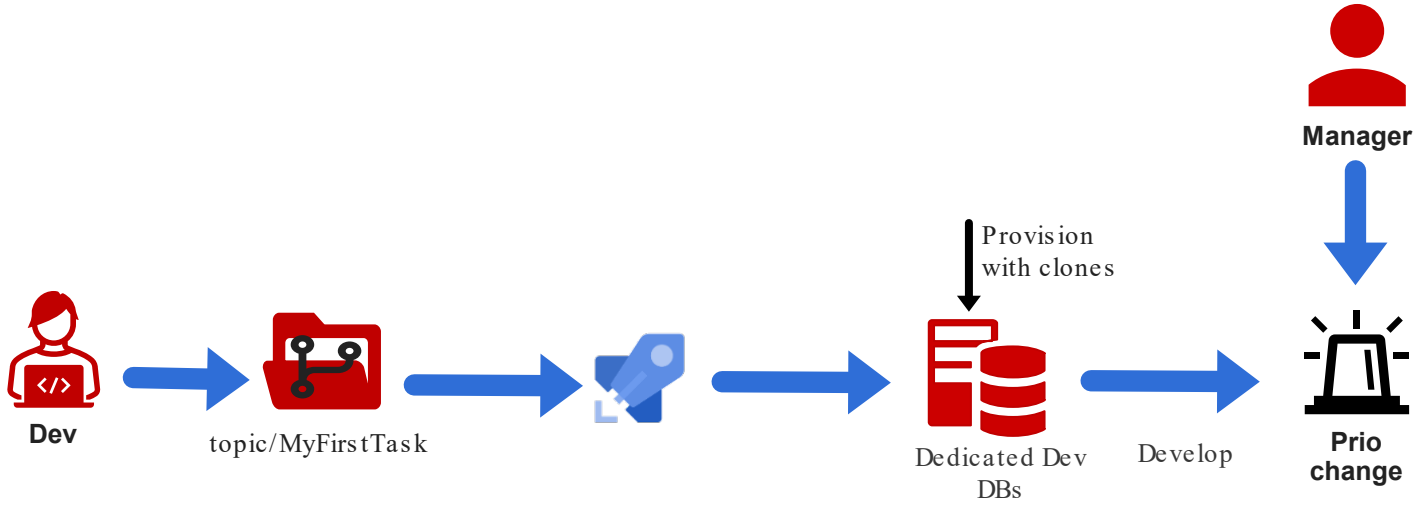
Introduction of database clones

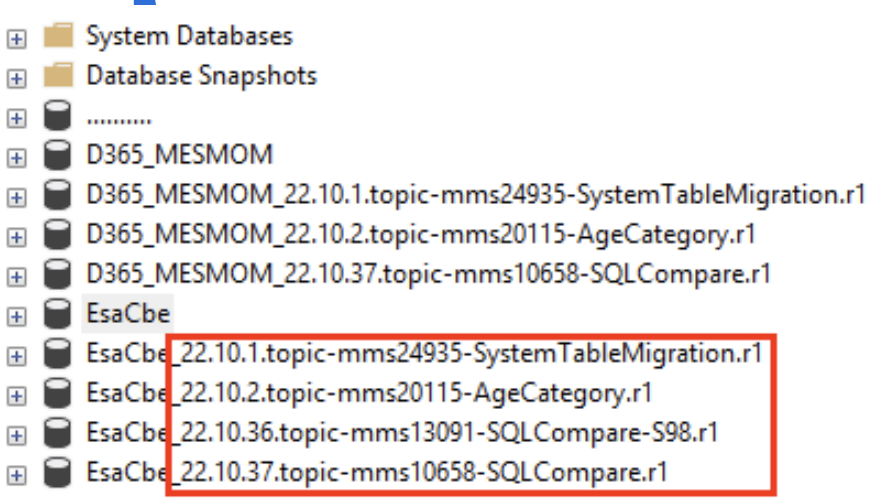
- Using cloning technology is smart
- Automating this usage is even smarter
  - The creation, use but also the house keeping

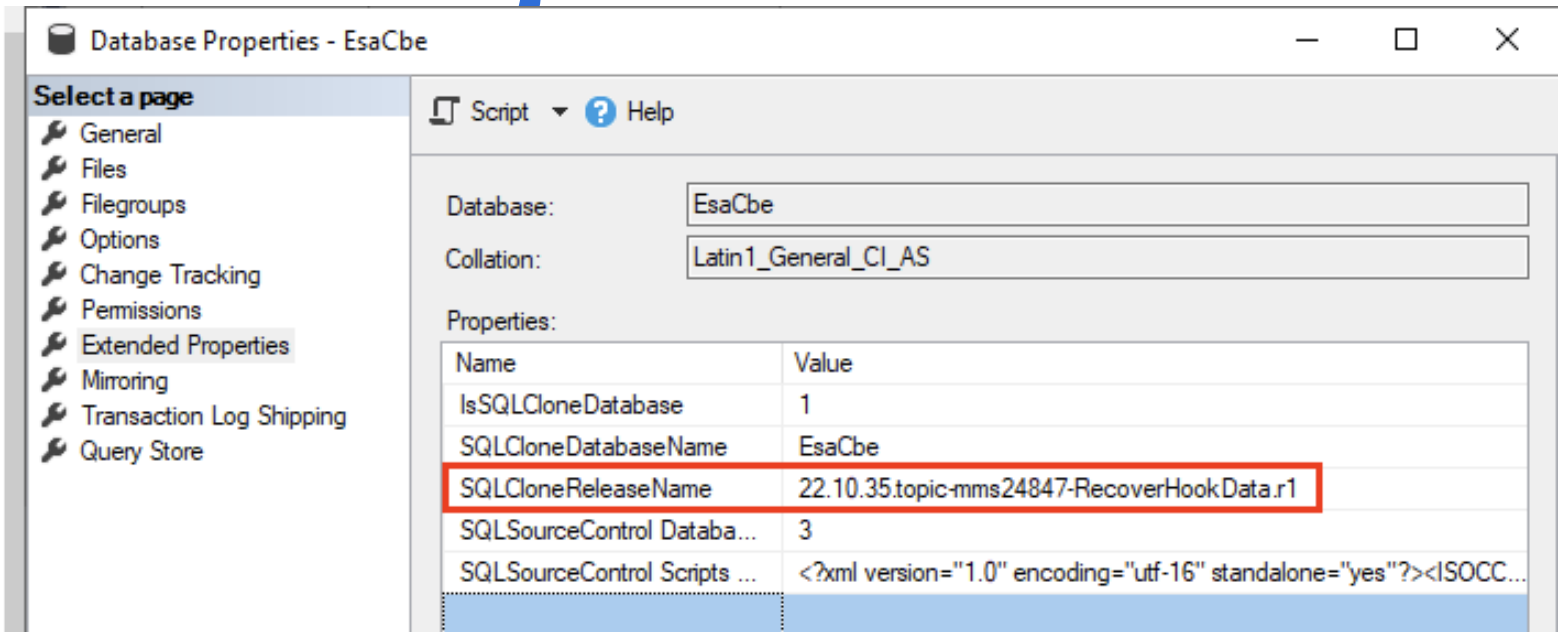


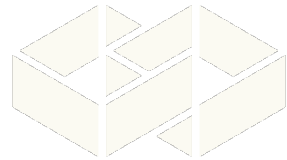
The concept of database stashing

- Changing priorities
- `git stash` but then for DBs
- Personal instance / environment
- Through AzDo not on your system

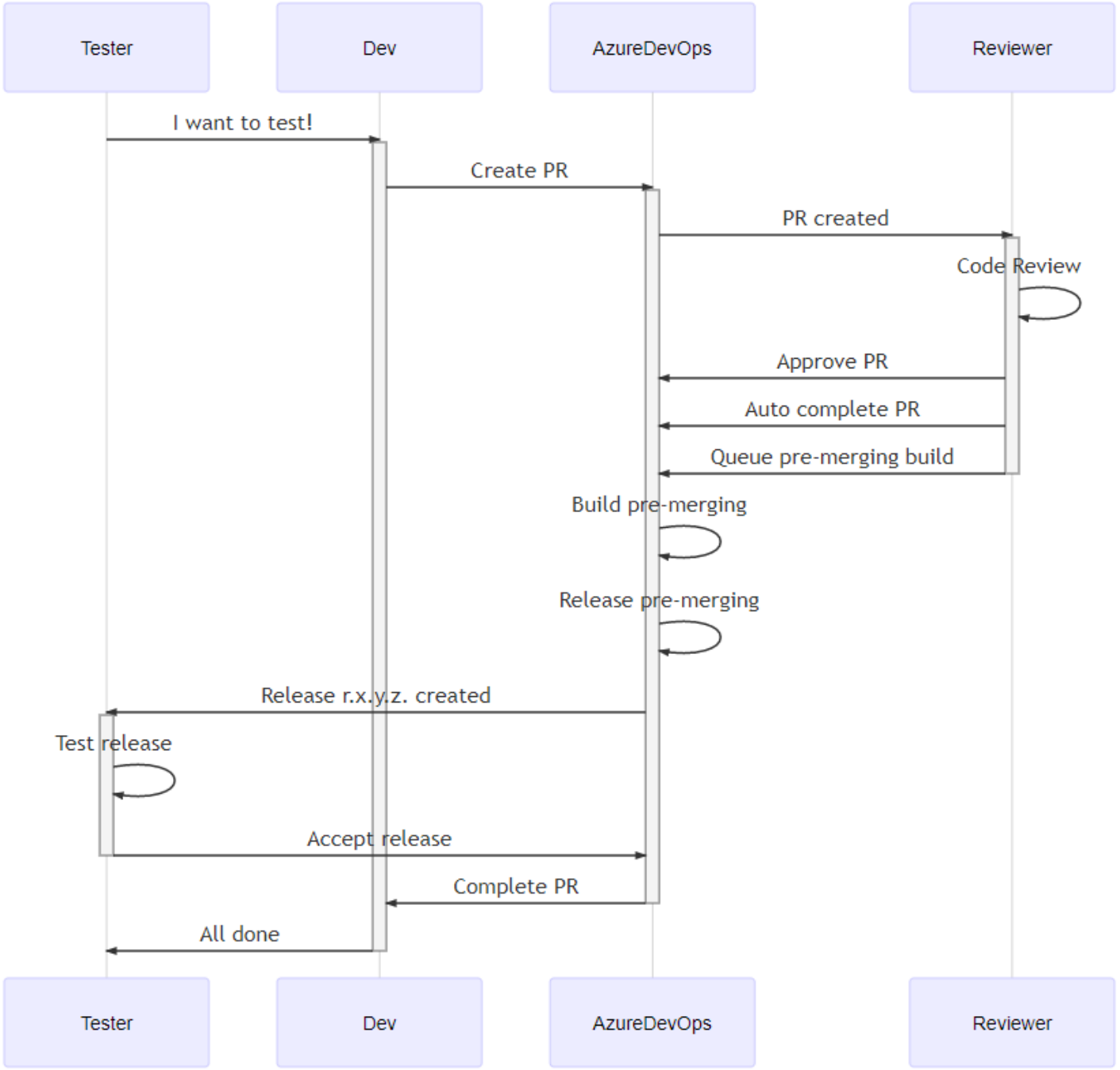








#dataleftunattended

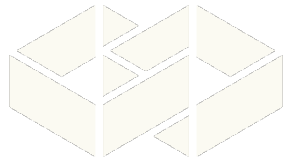


#dataleftunattended

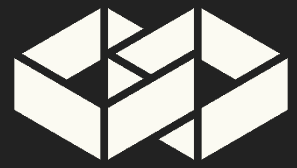
Manual vs Pull Request Release

- Release test are done on pre-merge build
  - More realistic testing
  - Prevent merge conflicts in master
- Less manual, more automation

  But please still communicate!

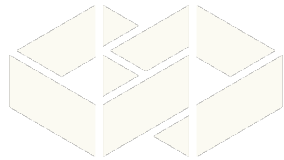


Demo



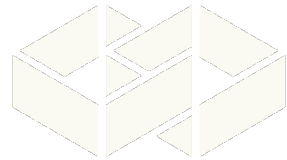
Deliver workflow improvements

- ✓ Test in parallel, just like development
- ✓ Make use of disposable environments
- ✓ Implement Pull Request Release workflow

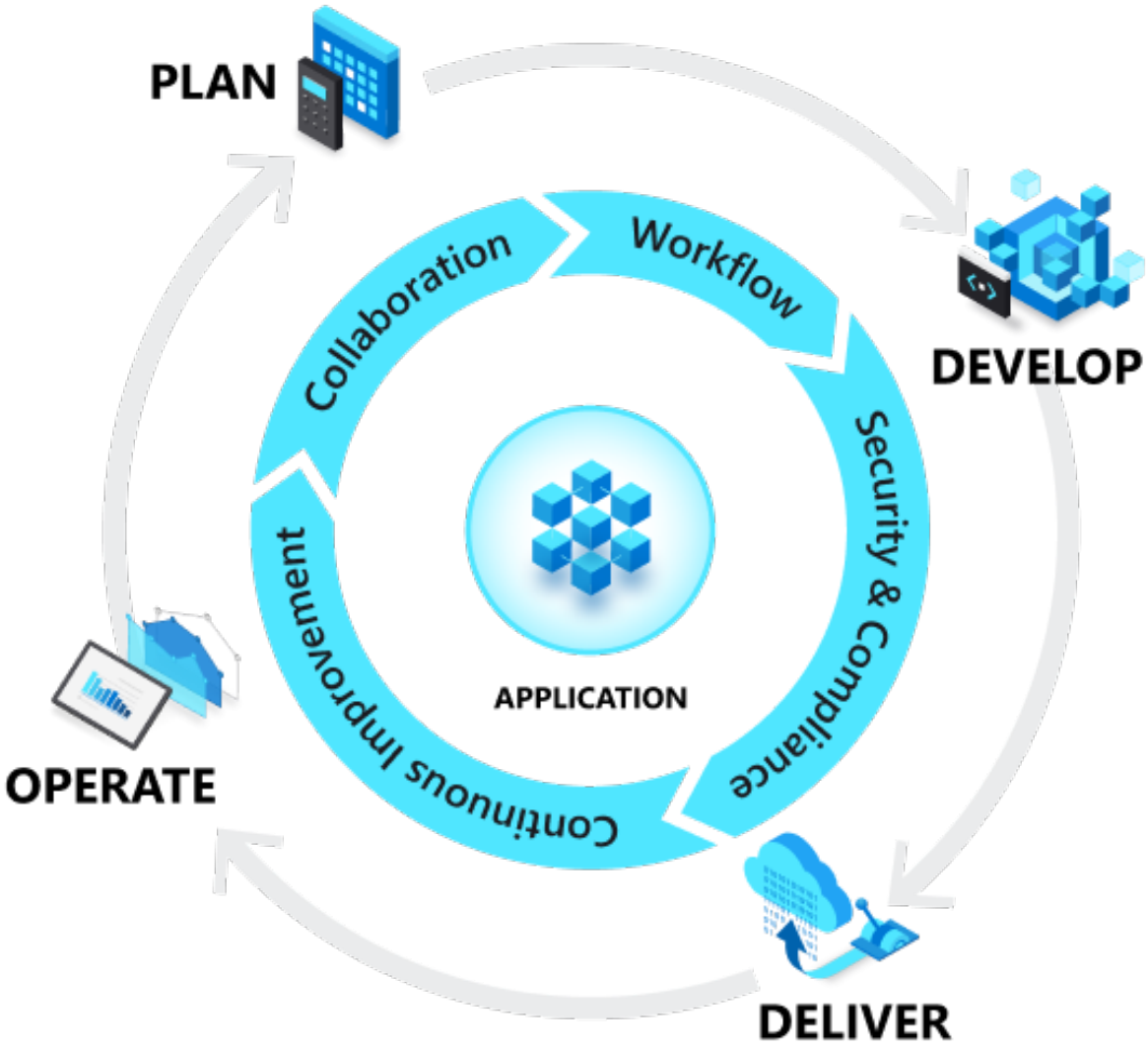


Deliver DevOps practices

- ✓ Automate delivery processes
- ✓ Use release pipelines with approvals
- ✓ Release artifacts to different environments

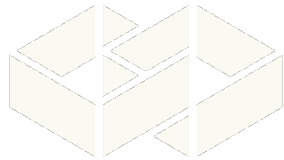


DevOps



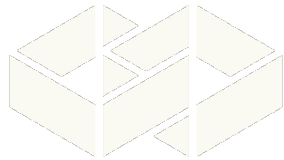
Operate DevOps practices

- Monitor & troubleshoot
- Securing knowledge



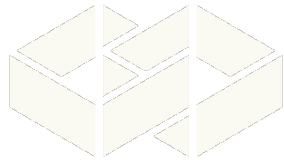
Troubleshoot

- What was changed?
- Who signed off for this?
- Where did the change come from?

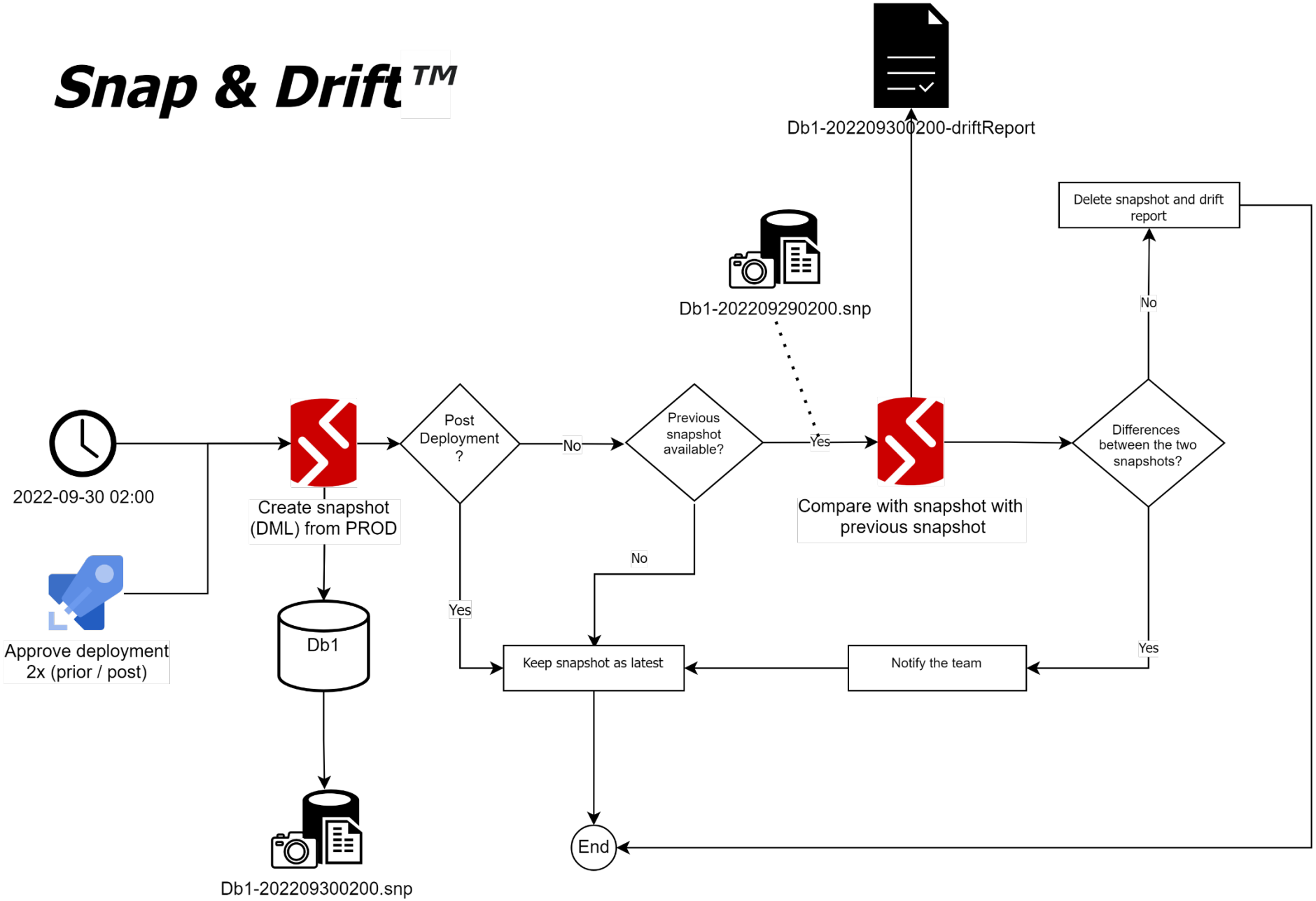


Monitor production databases

- Integrated in every deliver
- Nightly automated check
- Create work items on drift

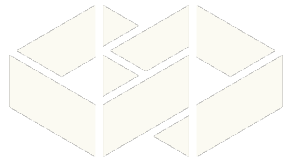


Snap & Drift™


Db1-202209300200-driftReport
Delete snapshot and drift report
Db1-202209290200.snp
No
2022-09-30 02:00
Post Deployment ?
No
Previous snapshot available?
Yes
Differences between the two snapshots?
Create snapshot (DML) from PROD
Compare with snapshot with previous snapshot
Approve deployment 2x (prior / post)
Db1
No
Yes
Yes
Keep snapshot as latest
Notify the team
End
Db1-202209300200.snp

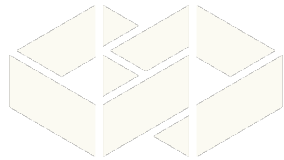
Troubleshoot

- ✓ What was changed?
- ✓ Who signed off for this?
- ✓ Where did the change come from?



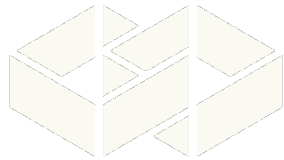
Operate DevOps practices

- ✓ Monitor & troubleshoot
- Securing knowledge



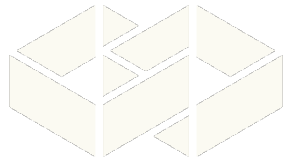
Share and securing knowledge

- Encourage documentation everywhere
- Linking ensures provenance
- SQL Extended properties (SQL Doc)
  - Convert to markdown (wiki) format
- Use Azure DevOps (Elastic)search



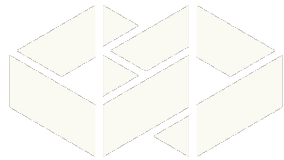
Operate DevOps practices

- ✓ Monitor & troubleshoot
- ✓ Securing knowledge



Key take aways

- People, process and tools (in that order)
- First do DevOps
- Only then Database DevOps can happen



Questions?

Session evaluation

Your feedback is important to us

Thank you

linkedin.com/in/toniehuizer

@promicroNL

github.com/promicroNL/events

www.promicro.nl