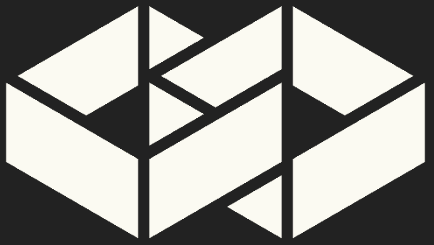


Transforming to DevOps, CJ/CD: Continuous Journey Continuous Disaster?

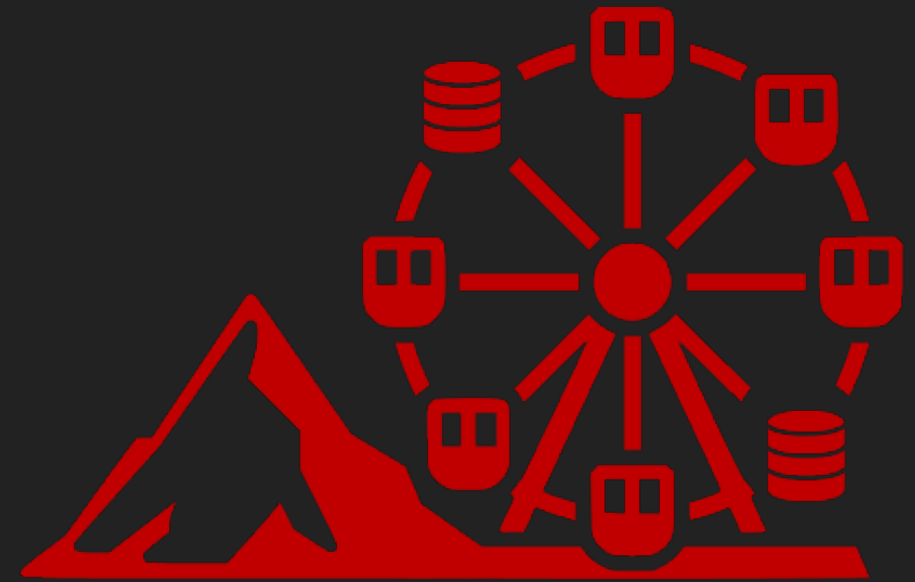


promicro

Tonie Huizer

DevOps consultant

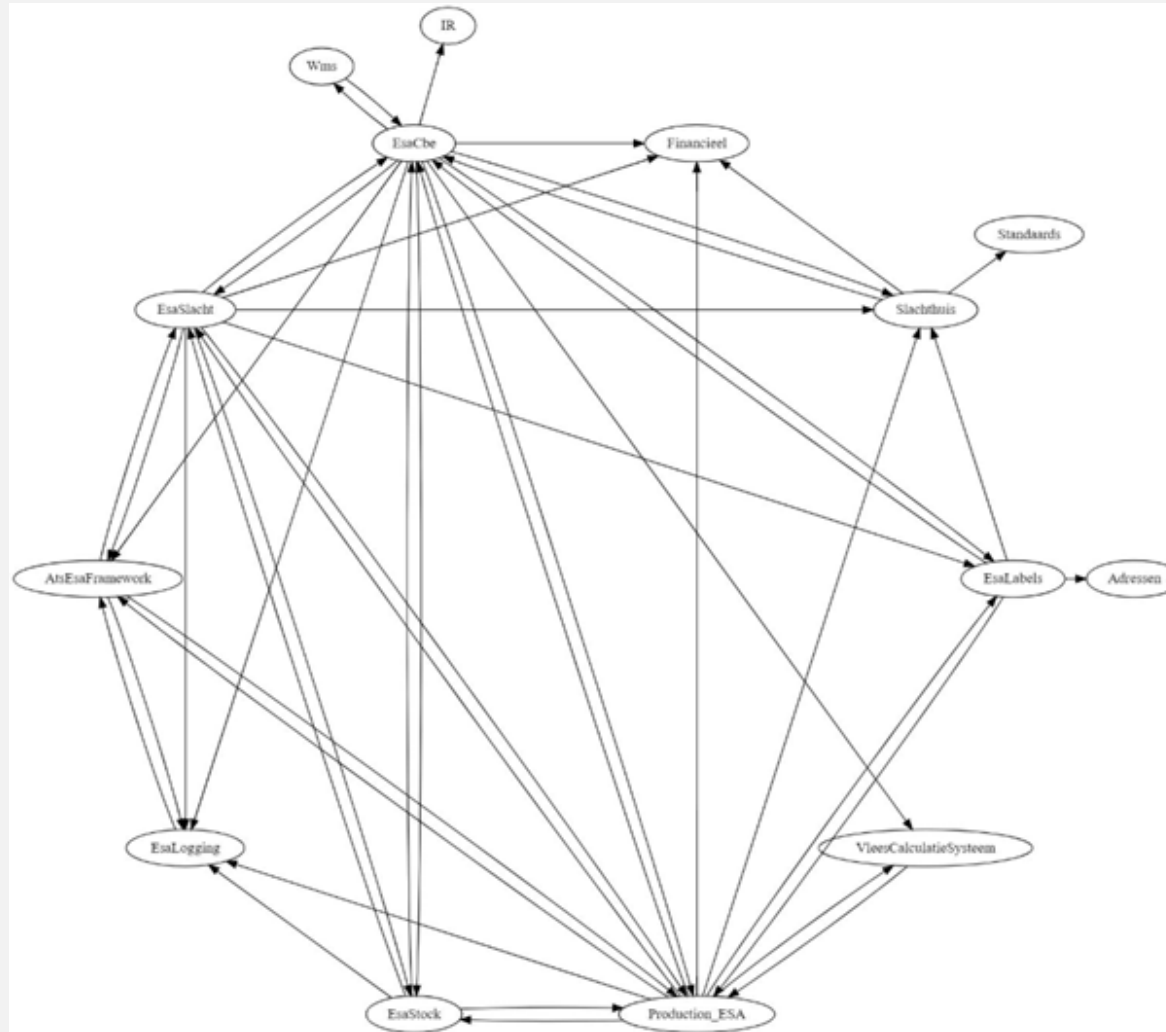
Promicro



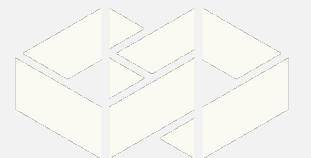
Data Community Austria Day 2025



Database DevOps – the challenge



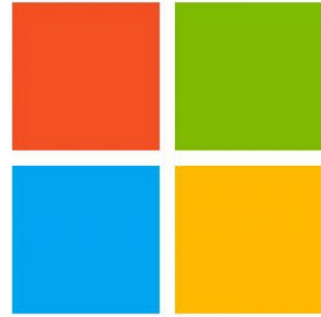
Database DevOps – the challenge



THANK YOU



Awesome Partner



Microsoft

Platinum



redgate



Gold



b.telligent
smart data. smart decisions.

Lucient

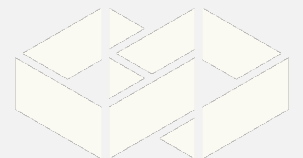
Bronze



Tabular Editor

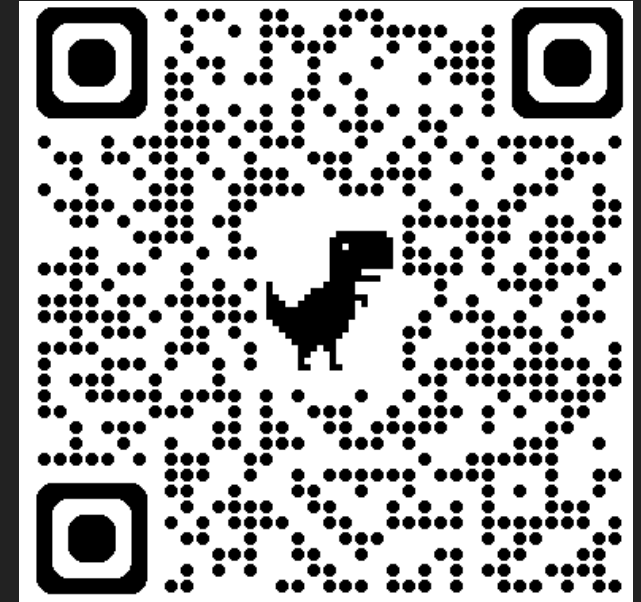


Power BI Camp
www.linearis.at



Tonie Huizer

He/him



Freelance DevOps consultant

Promicro

 linkedin.com/in/toniehuizer

 [@promicroNL](https://twitter.com/promicroNL)

 github.com/promicroNL/events

 www.promicro.nl

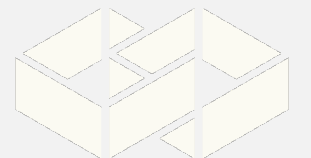


promicro

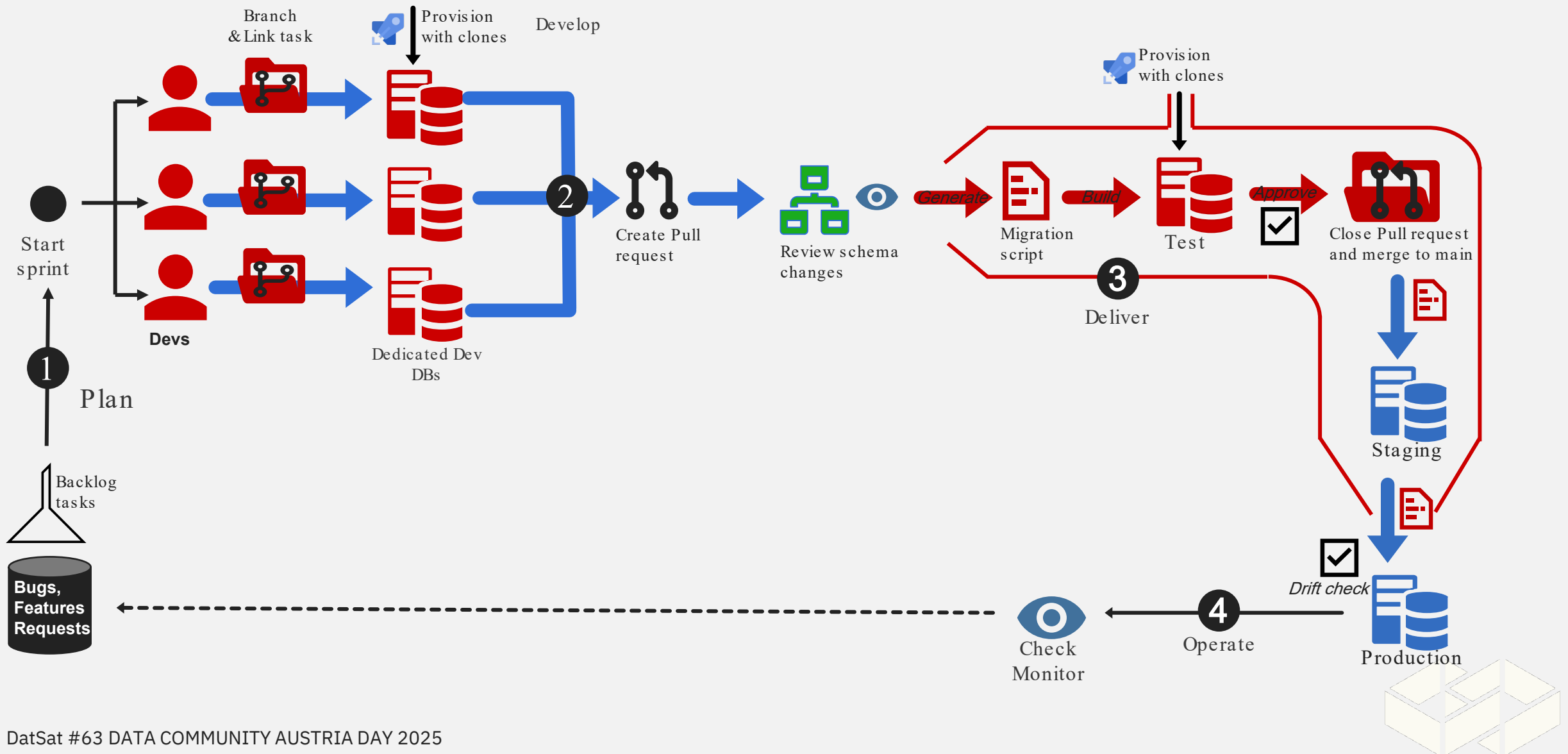


What to expect of today's session

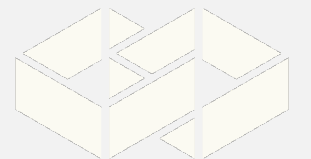
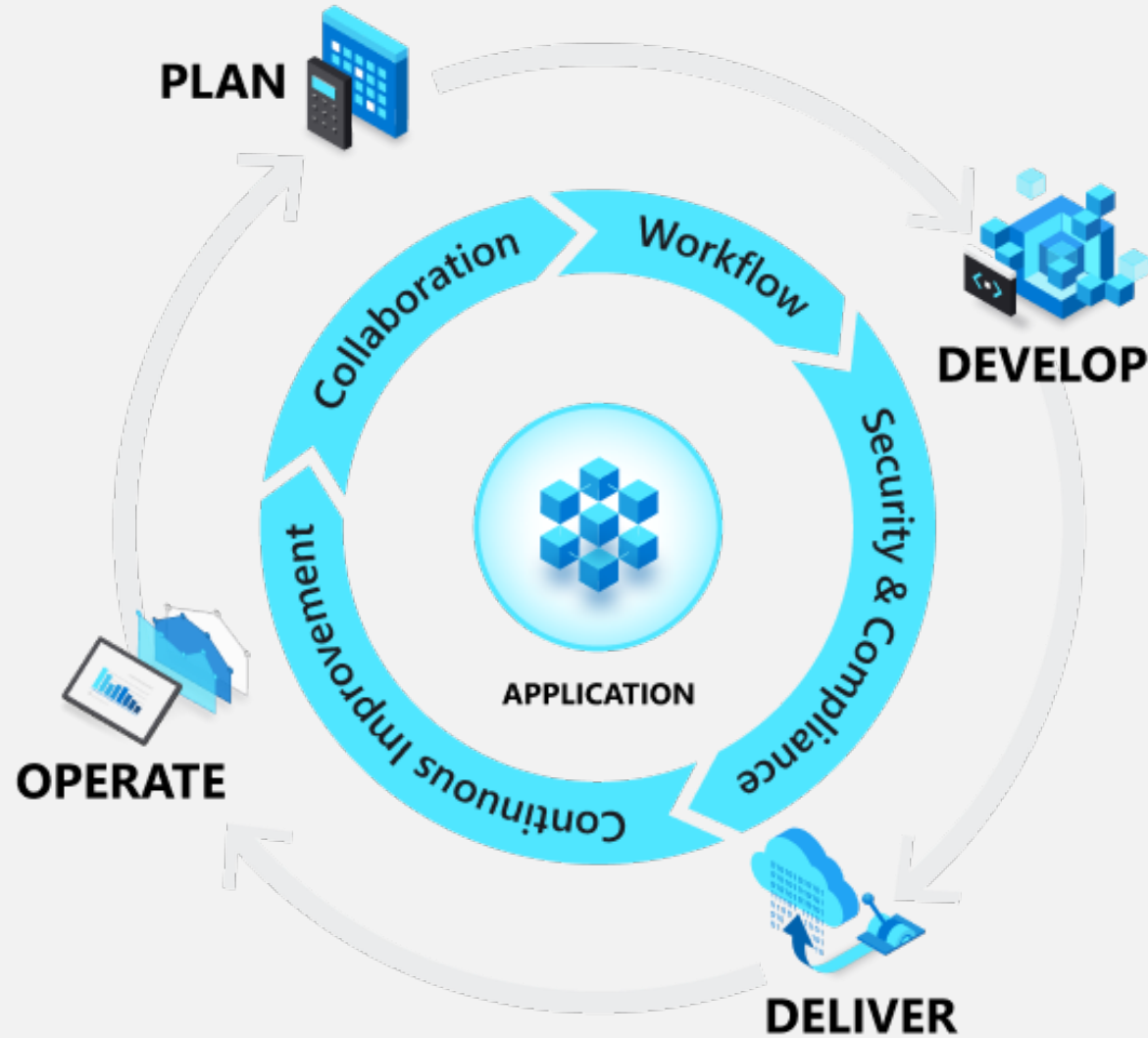
- Transforming to DevOps, CJ/CD?
- The 4 DevOps phases walk through
- Room for discussion & questions



Database DevOps – the end goal

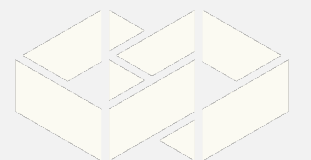


DevOps



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

- Donovan Brown, Microsoft



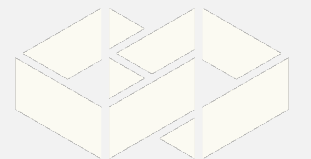
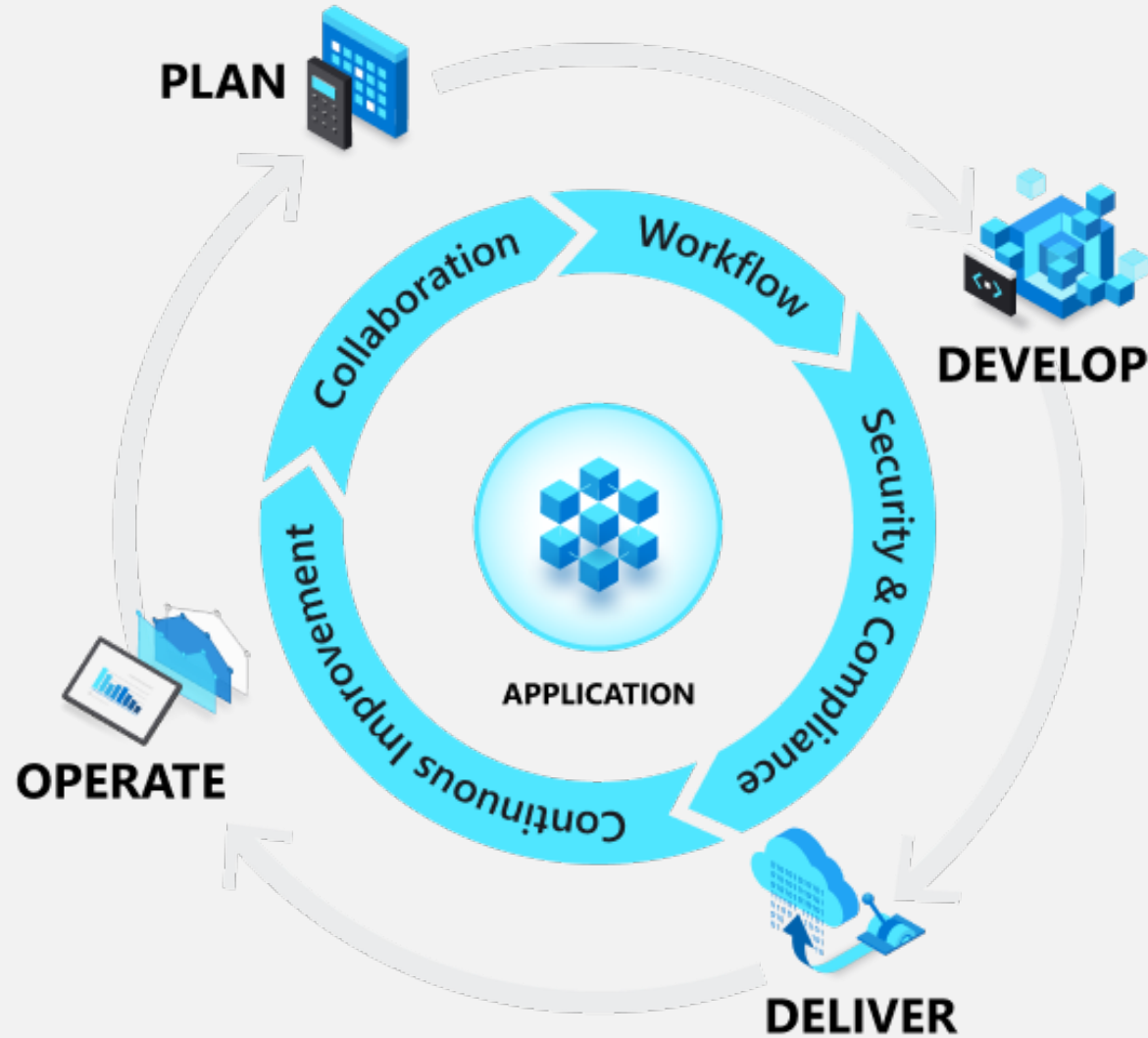


Teams

- 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



DevOps

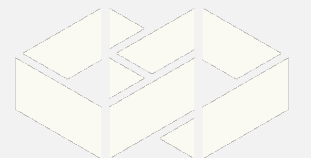
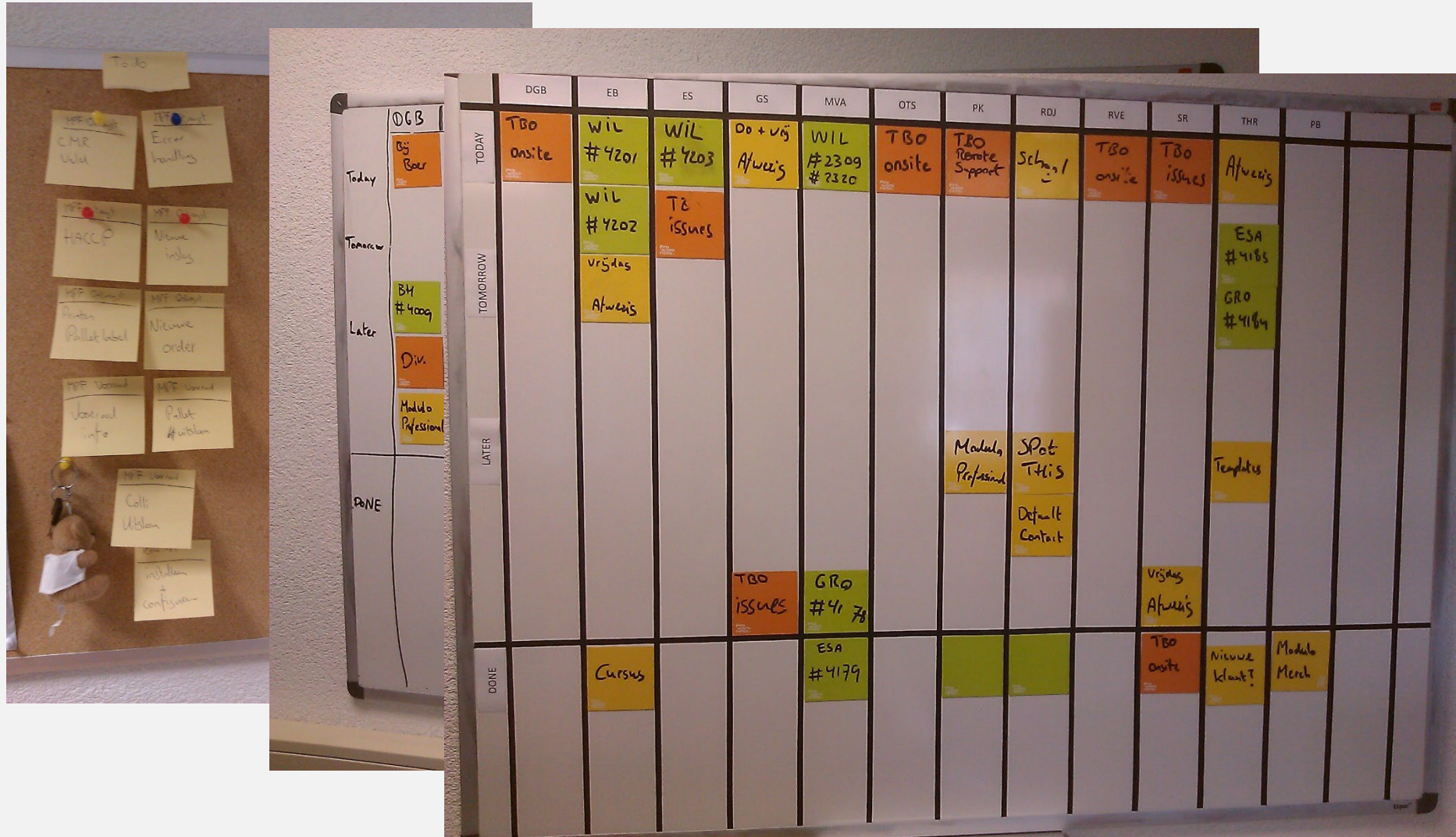


Plan DevOps practices

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



Attempts to plan with boards & sprints

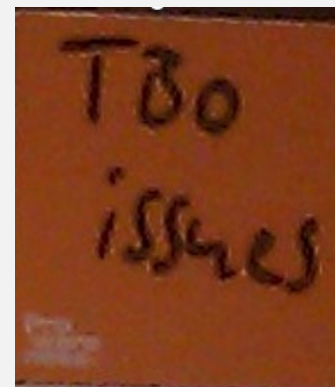


Working in sprints didn't work

Why?



- Team was too big
- No fixed iteration path
- Little involvement of the stakeholders
- Unrefined backlog items
- No links or integration



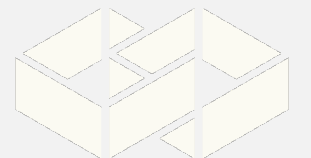
Demo



This time it worked

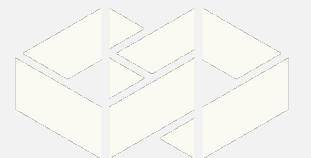
For 157 sprints in a row (and counting)

- ✓ 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

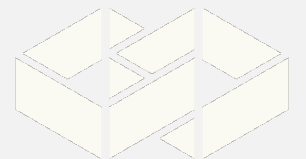
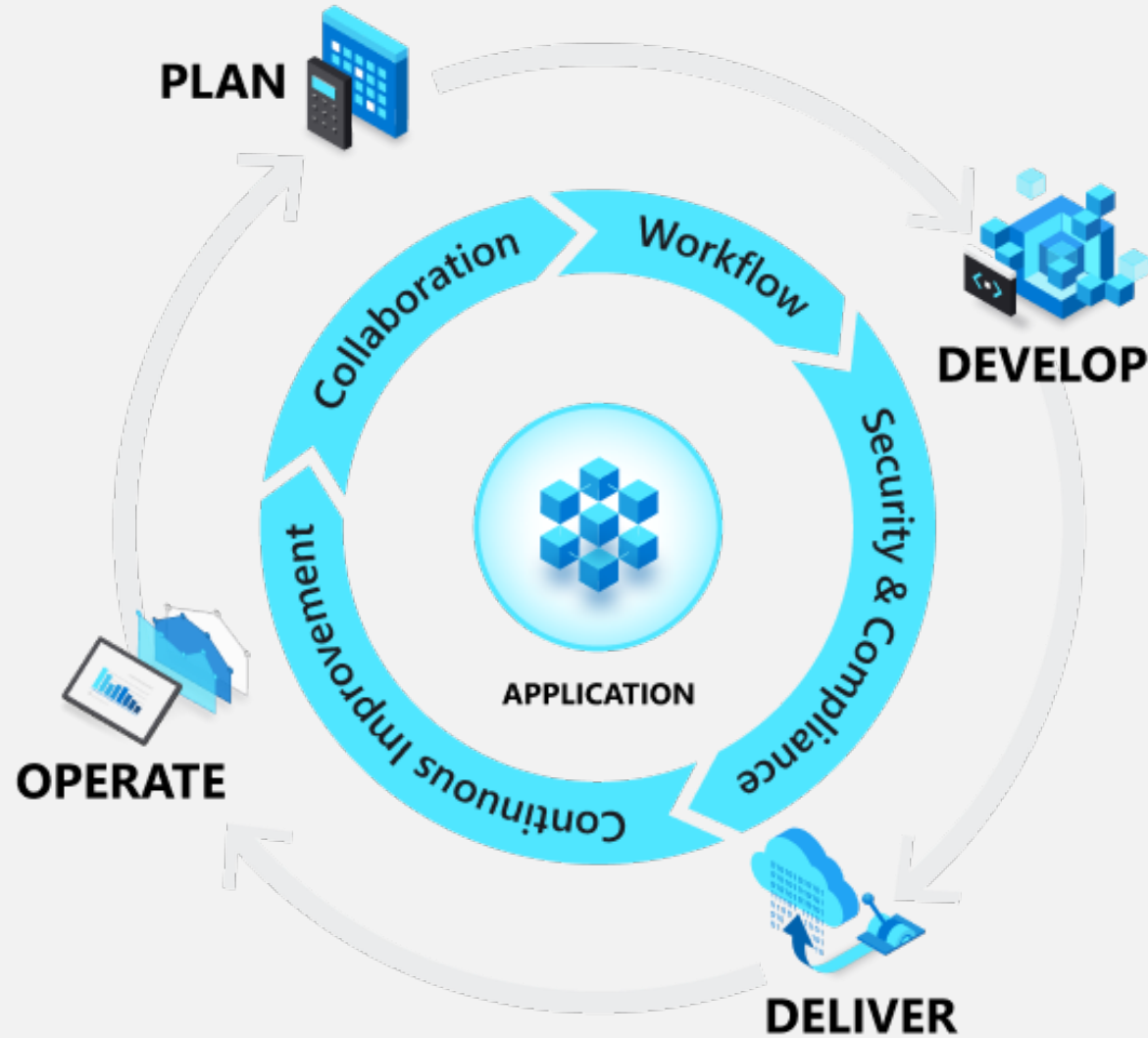


Plan DevOps practices

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

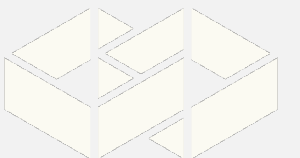


DevOps

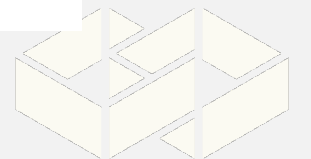
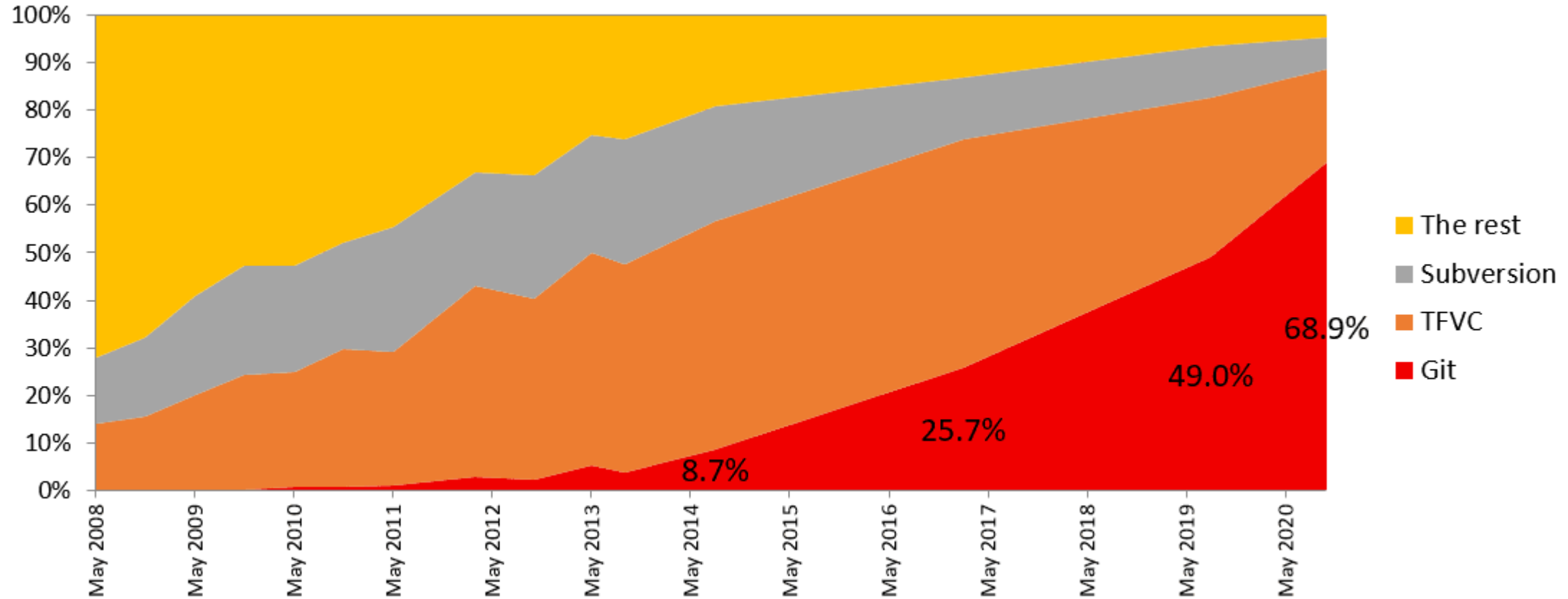


Develop DevOps practices

- 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



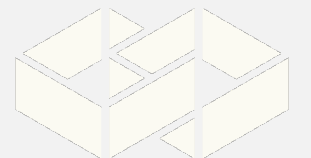
Switching to git

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

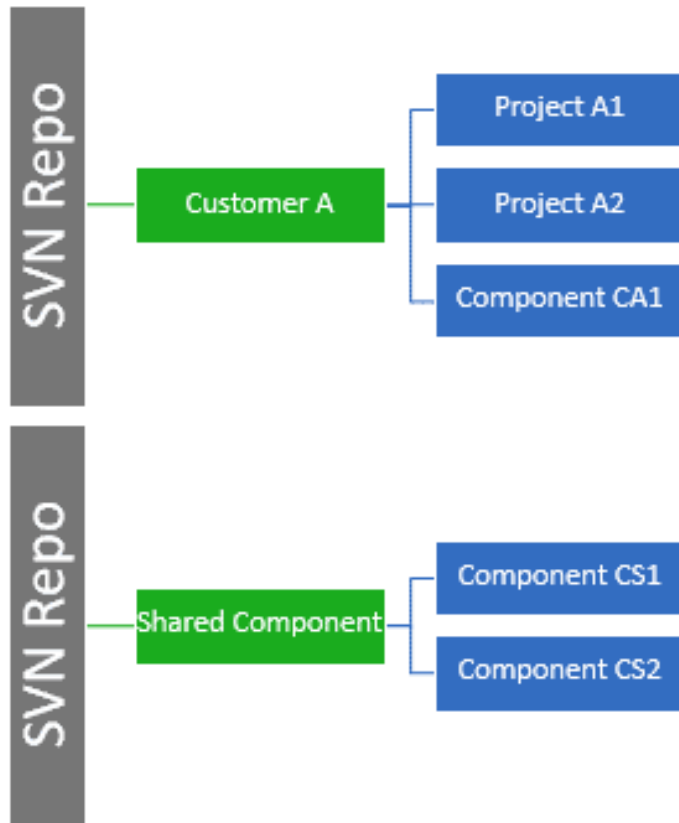


Migrate a VCS

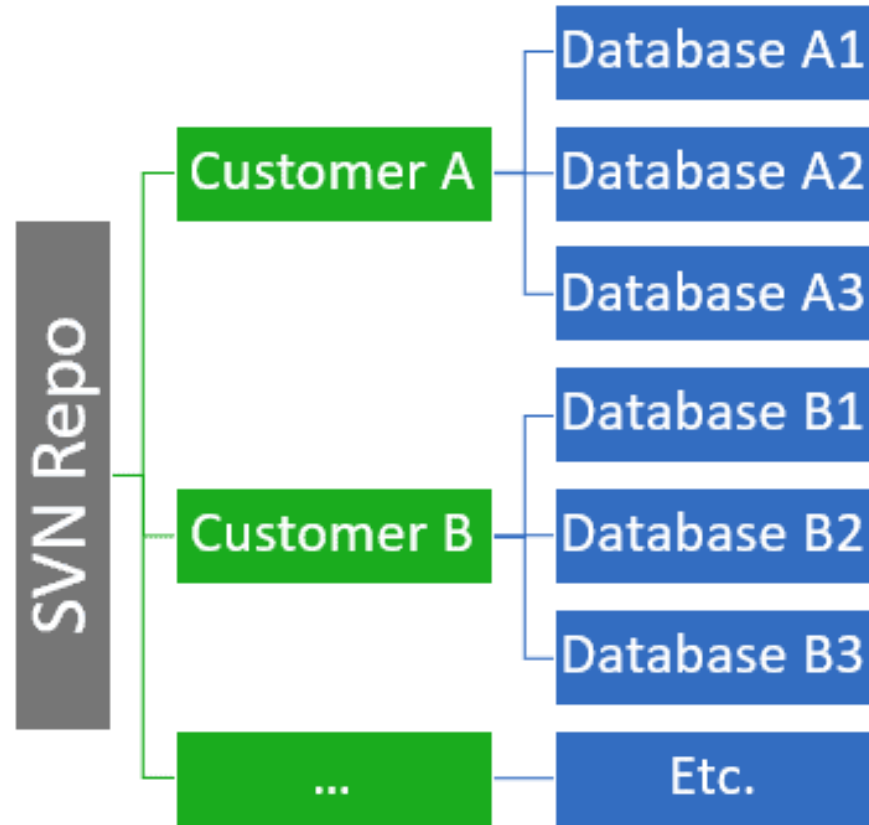
- Functional
 - Split up the big repo per customer / project
 - Separate the components
- Technical
 - SVN → Git → TFVC → Git



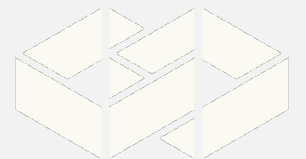
Migrate a VCS (old situation)



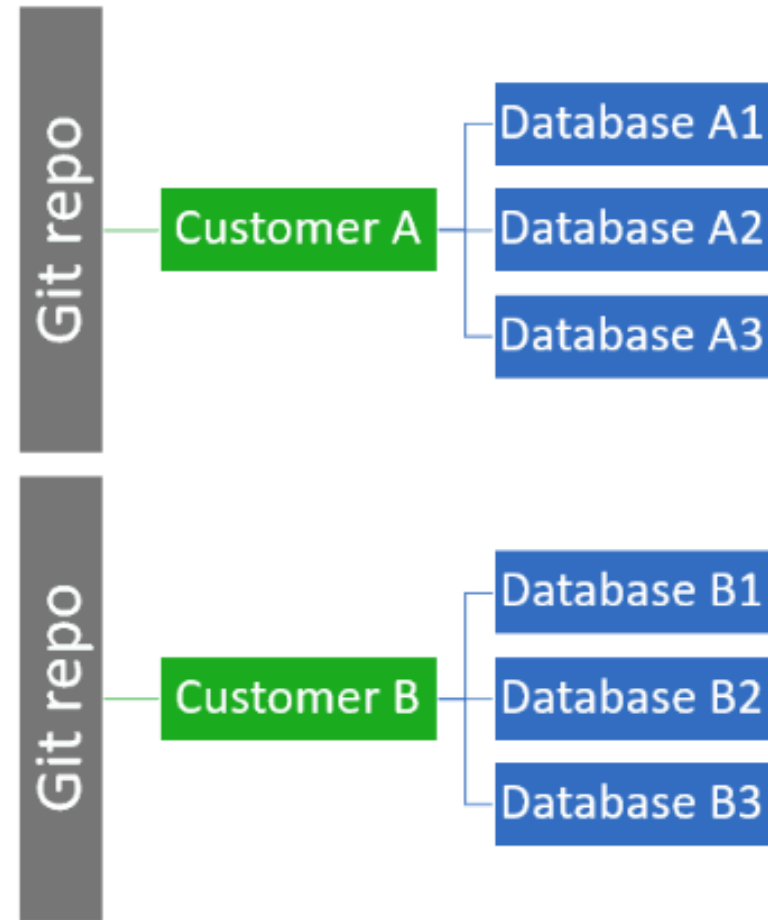
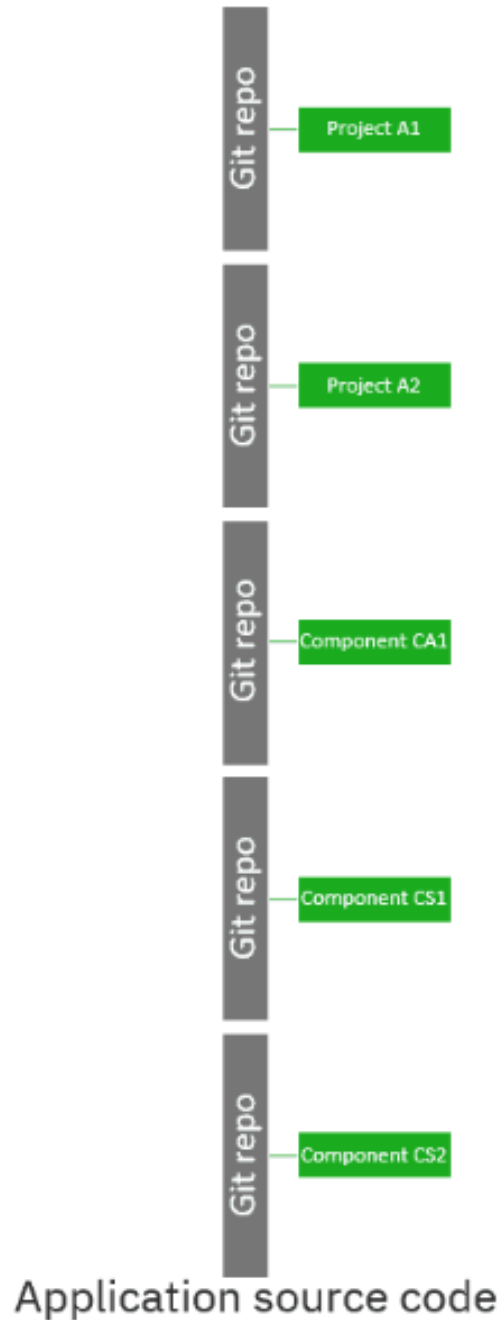
Application source code



Database source code



Migrate a VCS (new situation)

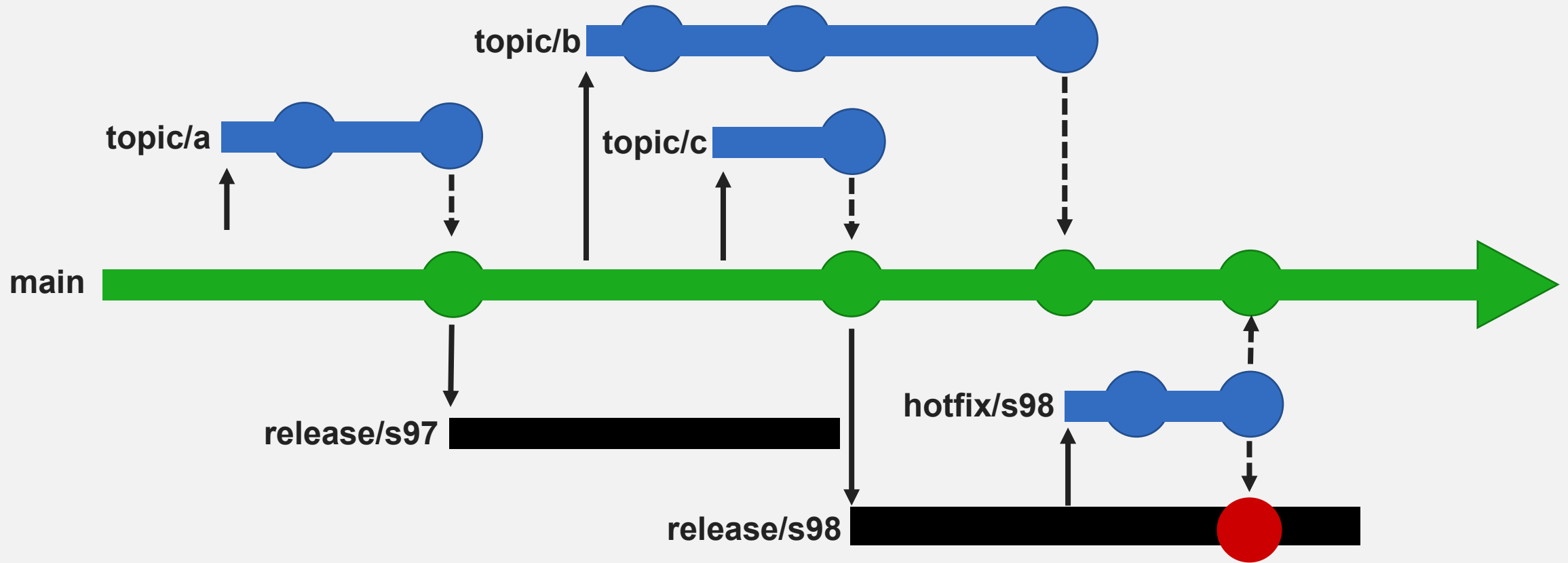


Branching strategy

- 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

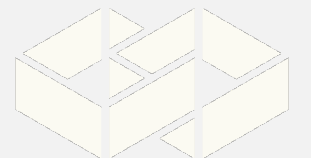
<branch category> / <hot fixed release - >bug<TicketId>-PascalCasingDescription

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

Release branch

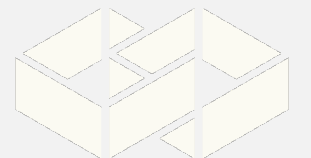
<branch category> / <unique identification>

release/s100



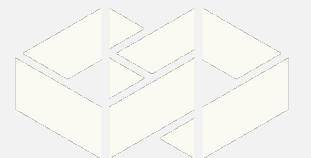
Git compared to SVN / TFVC

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

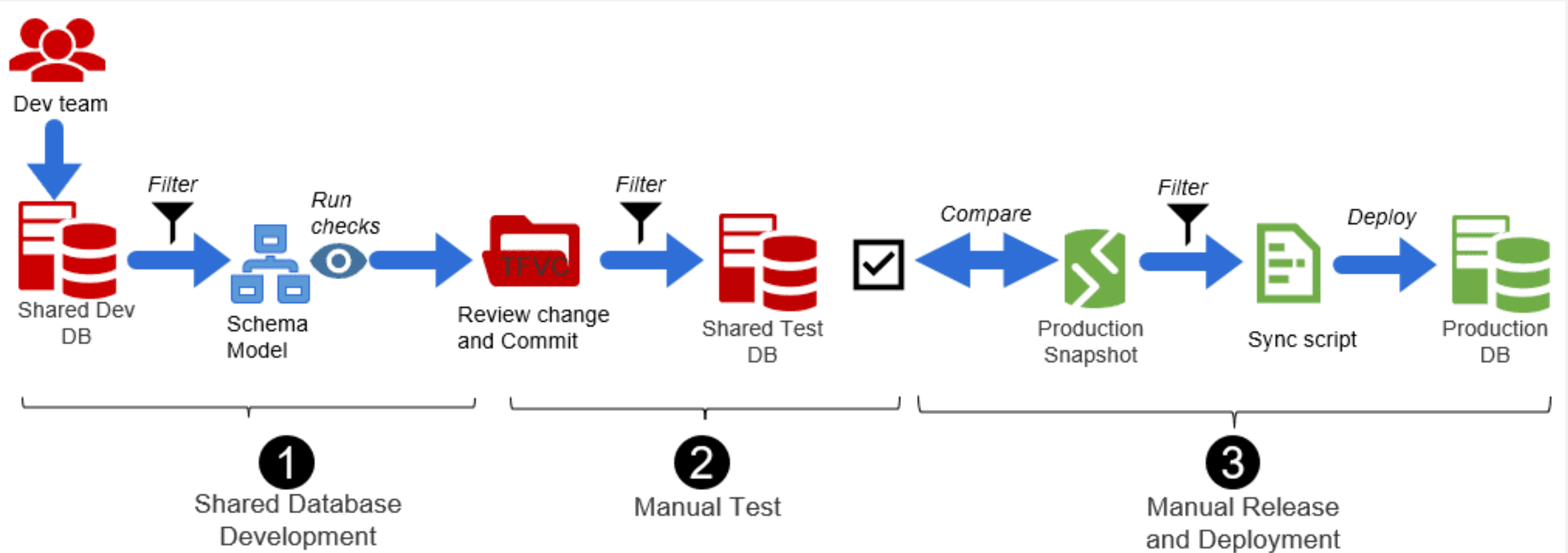


Develop DevOps practices

- ✓ 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 powershell but not inline
 - Use verbose logging

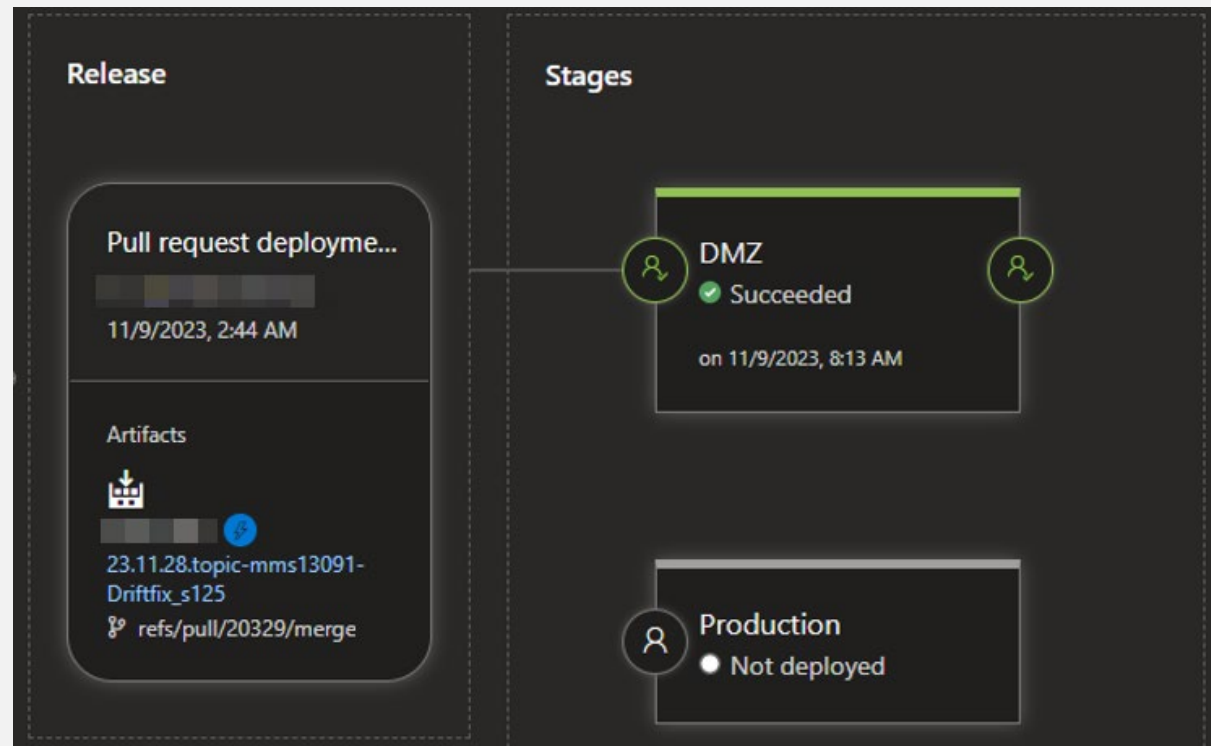
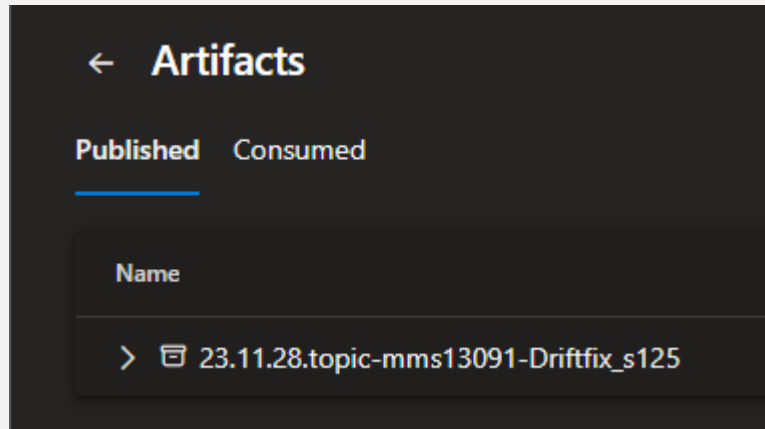


Demo



Build artifact & automate repetitive tasks

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

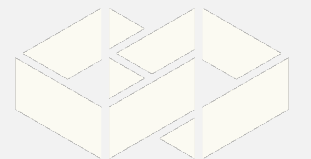
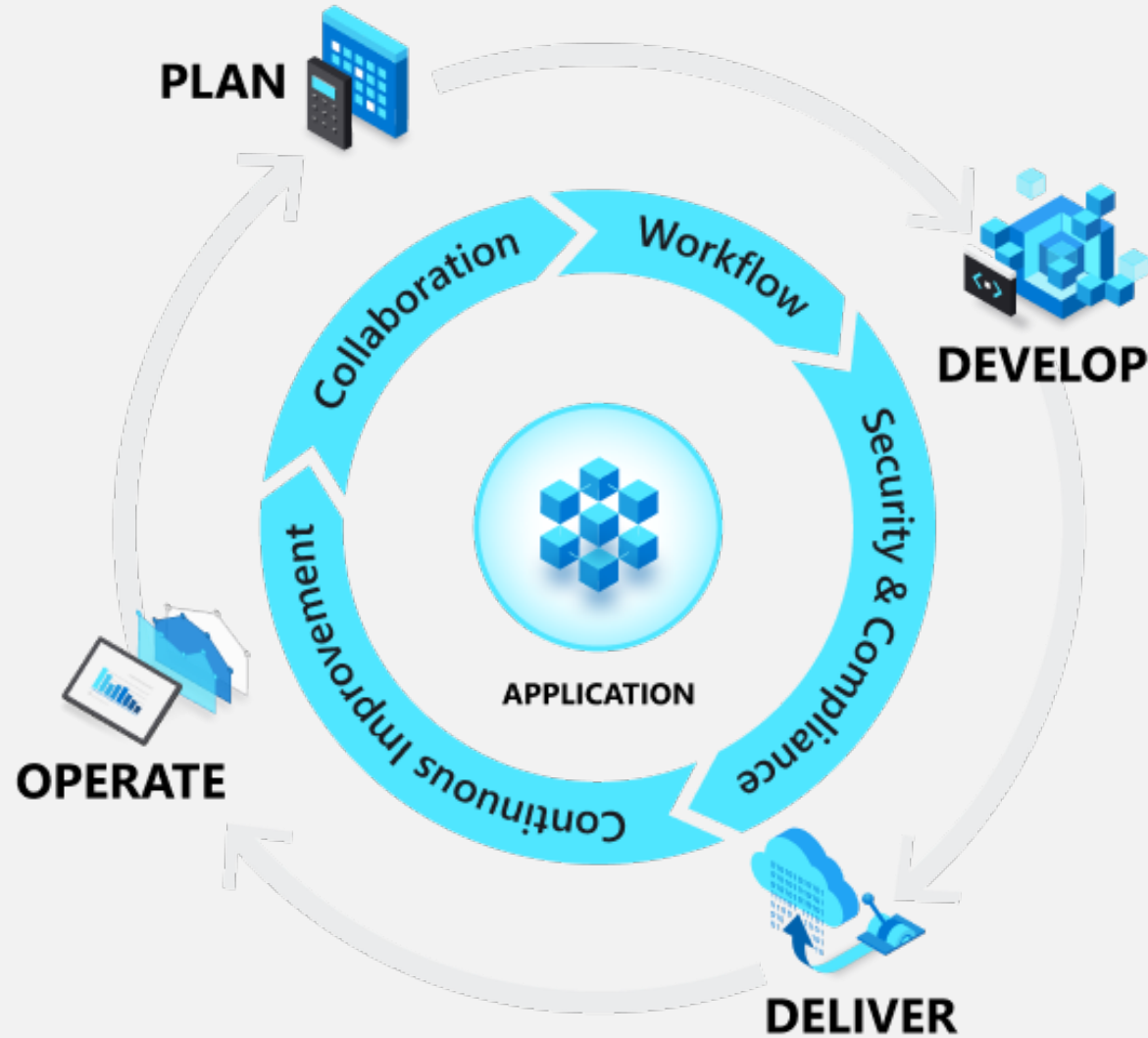


Develop DevOps practices

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



DevOps



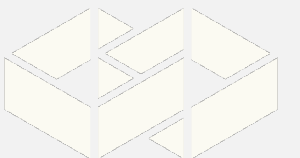
Deliver DevOps practices

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



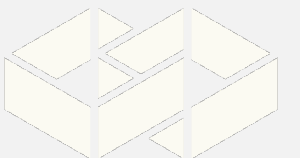
Previous deliver workflow

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



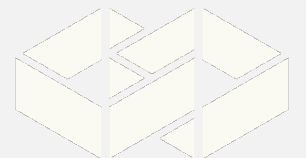
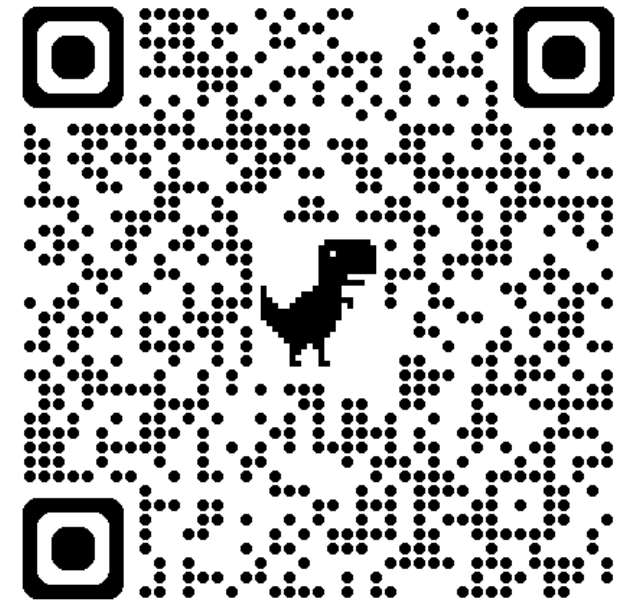
Deliver workflow improvements

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



Introduction of automated database clones

- Using cloning technology is smart
- Automating this usage is even smarter
 - The creation, use but also the house keeping
- Personal instance but through the AzDo Process
 - Including “Database stashing”



Demo



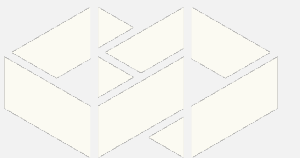
Manual vs Pull Request Release

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



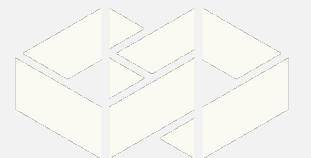
Deliver workflow improvements

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

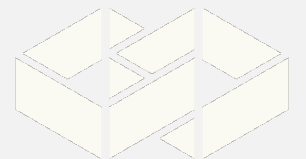
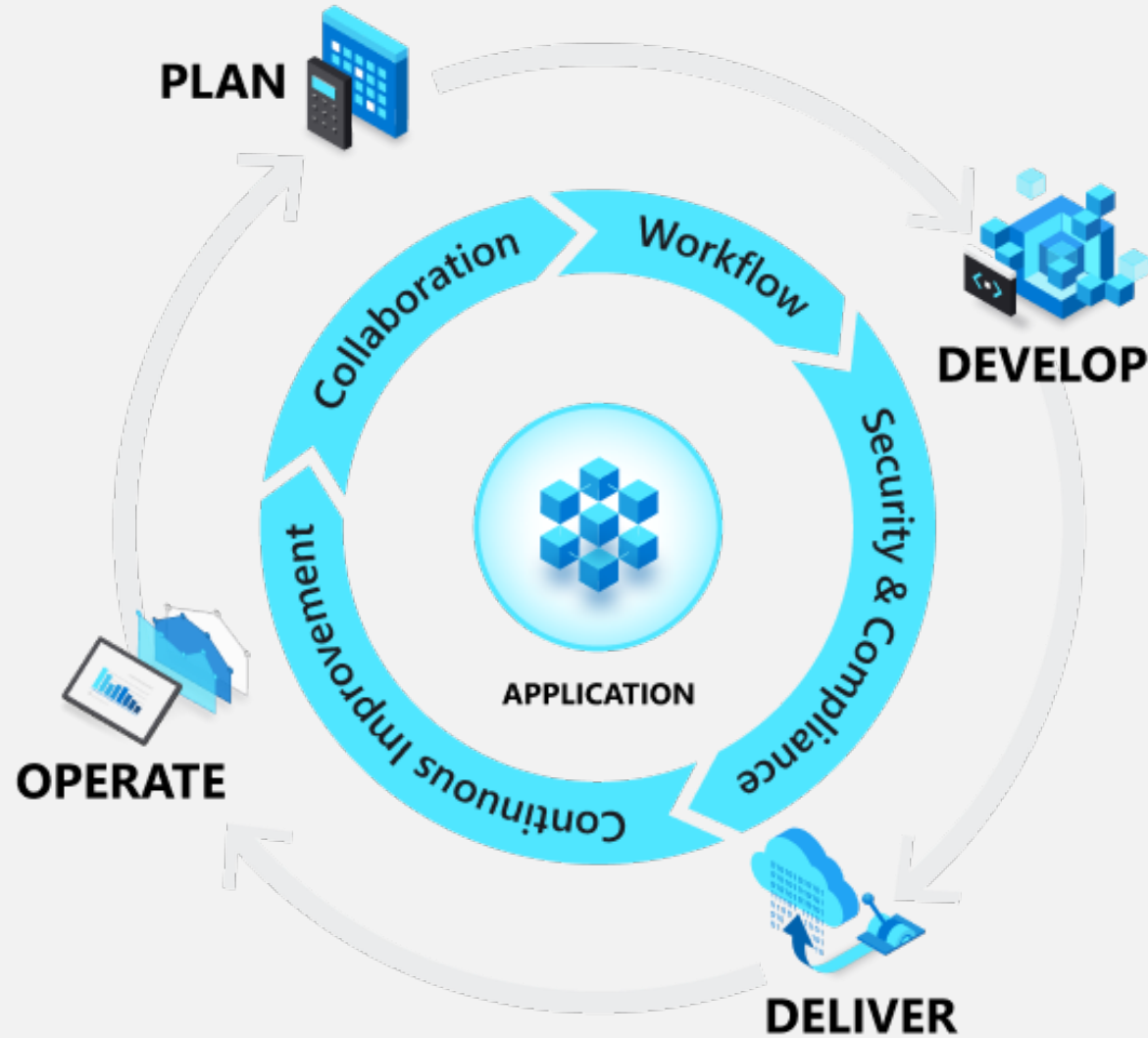


Deliver DevOps practices

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

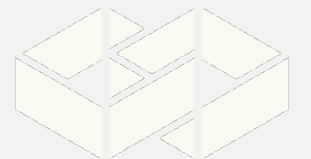


DevOps



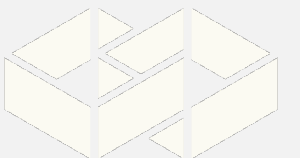
Operate DevOps practices

- Monitor & troubleshoot
- Securing knowledge



Troubleshoot

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

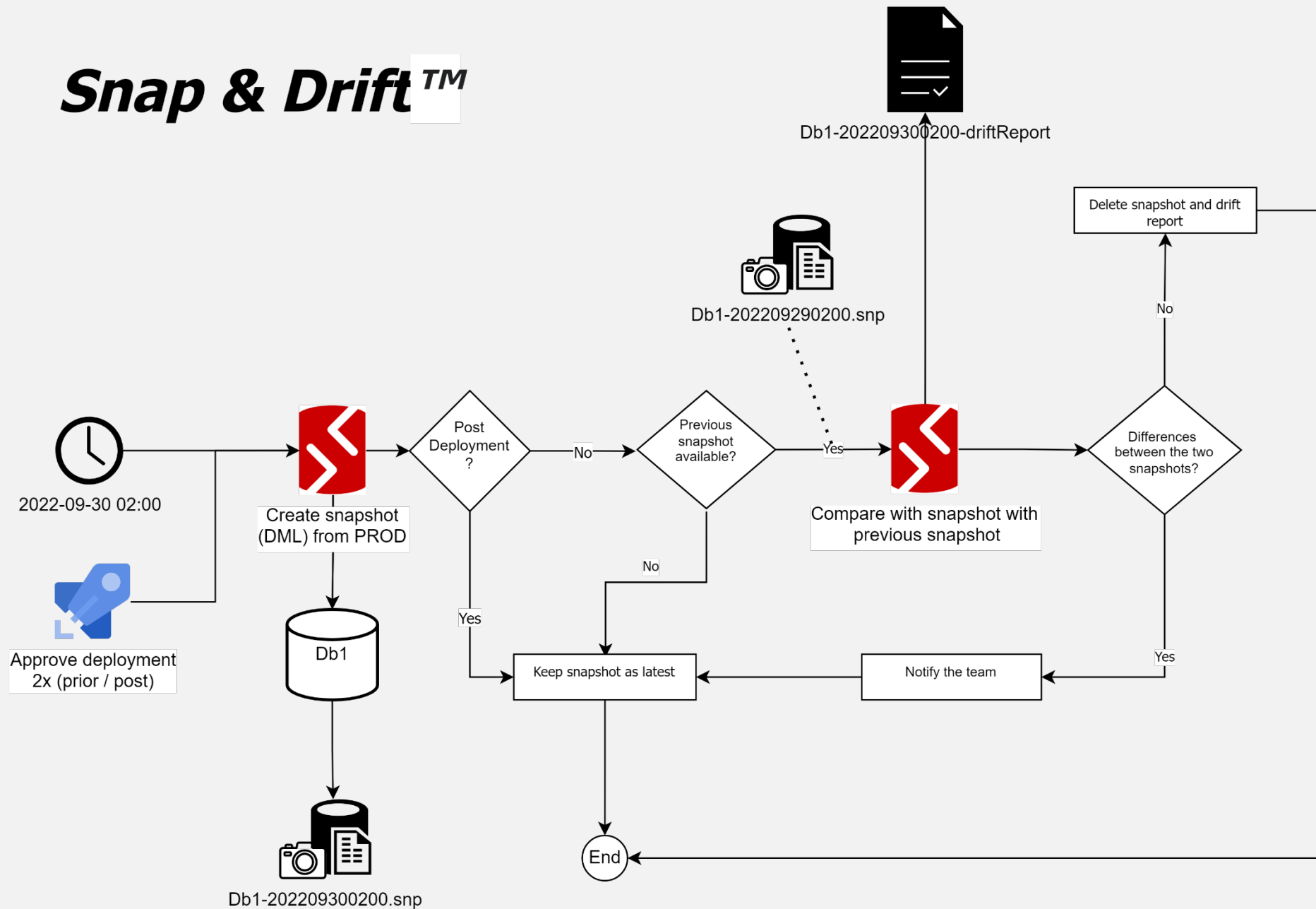


Monitor production databases

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



Snap & Drift™



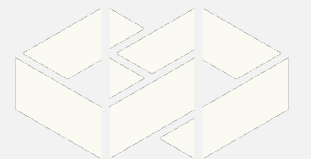
Troubleshoot

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



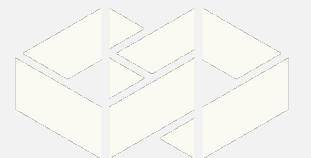
Operate DevOps practices

- ✓ Monitor & troubleshoot
- Securing knowledge



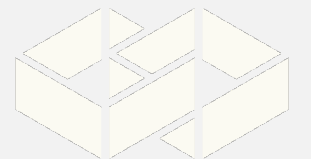
Share and securing knowledge

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



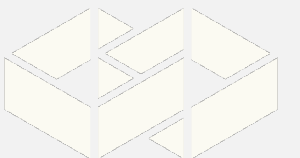
Operate DevOps practices

- ✓ Monitor & troubleshoot
- ✓ Securing knowledge



Key take aways

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



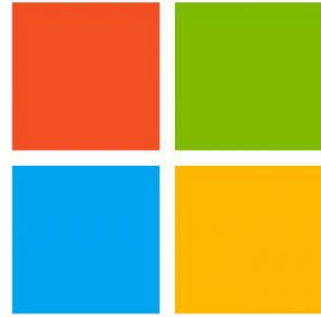
Questions?



THANK YOU



Awesome Partner



Microsoft

Platinum



redgate



Gold



b.telligent
smart data. smart decisions.

Lucient

Bronze



Tabular Editor



Thank you

 linkedin.com/in/toniehuizer

 [@promicroNL](https://twitter.com/promicroNL)

 github.com/promicroNL/events

 www.promicro.nl

