4SYNERGY

PdVBV

Notes, abstract

Put Dev-Ops into perspective.

With recent tools, notably Docker, the use of "containers" has become a way to develop and deploy systems quickly and flexible. This presentation will take a look at the use of both application-containers (Docker - anyone?) and Database-containers (PDB). Containers can be used in both "old-fashioned" projects and in new-and-shiny Dev-Ops and cloud-deployed systems. The presentation can help you see through the buzzwords, and start using Containers to your advantage.

Level: All

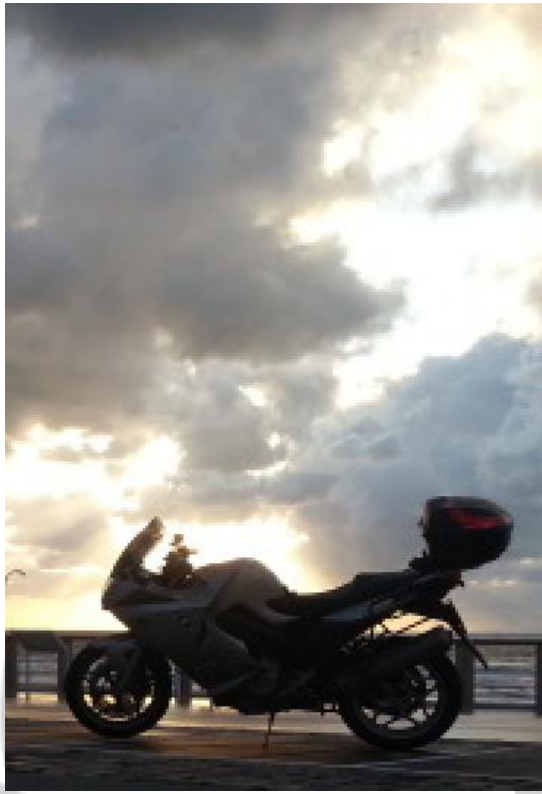
Time: anywhere from 30 to 60 min.

Repeat of history, with different nameing convention.

Dinosaur DBA
vs
Dev Ops

Piet de Visser

The Simple Oracle DBA



Favorite Quotes: “The Limitation shows the master” (Goethe), “Simplicity is not a luxury, it is a necessity. Unfortunately, “Complex’ solutions sell better. (EW Dijkstra).

Logo Cloud

4SYNERGY

PdVBV

- Shell
- Philips
- ING bank
- Nokia
- (dutch gov)
- Insinger, BNP
- Etihad
- NHS
- BT
- Claritas, Niels
- Unilever
- Exxon
- GE

Don't waste time on Self-Inflation… but Hey, this was such a cool Idea (from a marketing guy)…
Logos of my major customers over time. If you want your logo here: Hire me.

What does it look like..

- 


Couldn't resist… after this changing room, not allowed to take pictures anymore..
For travel pictures from Asia: later…

4SYNERGY

Agenda (approx 45 min) PdVBV

History… Data + Code (old stuff…)

Containers and Oracle (Two things)

“Docker” (New)
Oracle PDBs (Newer)

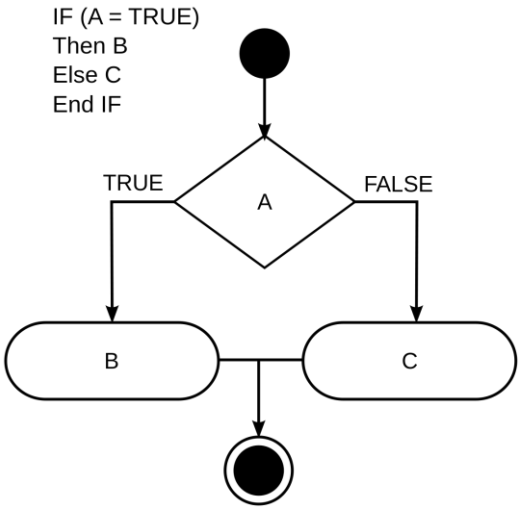
When + Where to use
Lessons ?

10 min Discussion (What do You Do)

Agenda: What is going to happen 1-in this talk and 2-to your job.
20+ slides, 2min/slide.

Data and Code

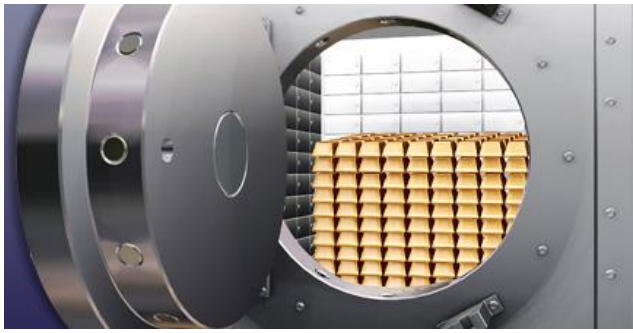
- **Information => Data**
  - Your Salary, Your Bank-stmnt
  - Your Name + Address
  - Your browsing history..
- **Process => Code**
  - Allow you to buy a gadget.
  - Deliver the gadget to your door.
  - Count your visits to FB..
- **Data or Code:**
  - Which is more important?
  - Both – But Different.
  - Don't mix the two ?



IT is about processing "information", We work with data. The things we do with the data are determined by "code"

Data: Information to “Keep”

- **Data is “defined”**
  - Files or Tables. ACID.
  - Schema: on write or on read..
  - Data-collections will grow (never shrink?)



- **Data needs “storage”**
  - Save + Safe to disk
  - Volume will vary (grow)

- **Database (-Containter): “a pet”**
  - Want to keep, cannot afford to loose.
  - Data – Relevant History.

Any place to save data must be like a “warehouse” storage facility.
Safe, able to accommodate, able to grow

Process: Work-instructions..

- **Process must be “defined”**
  - Agreed and Tested actions.
  - Use + Manipulate data
  - Processes will change over time (slowly)

- **Process-definition (the source-code!)**
  - Save + Safe to disk
  - Versions will grow…

- **Source-code (the VCS) is “a pet”**
  - Want to keep, cannot afford to loose.

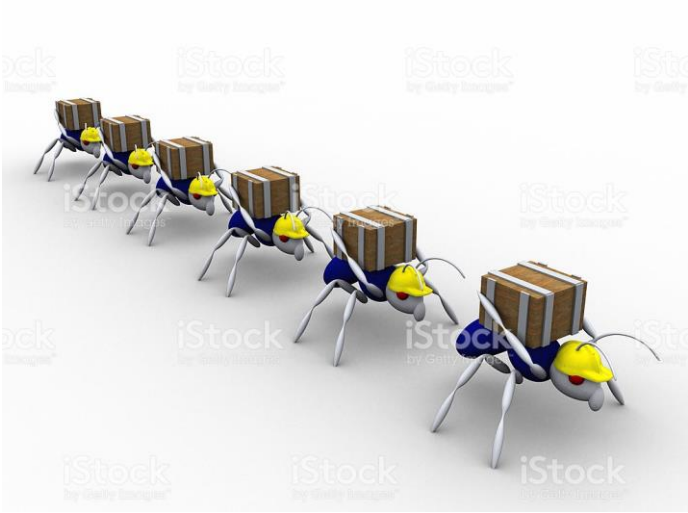
The “process” determines what we can/must/want to with data.
The definition of that process is important,

4SYNERGY

Processing: Execution..

PdVBV

- **Run, Execution of process on data.**
  - **Use Agreed data + process**
  - **Execute “work” – Anywhere.**
  - **Running of Processes often repeated.**

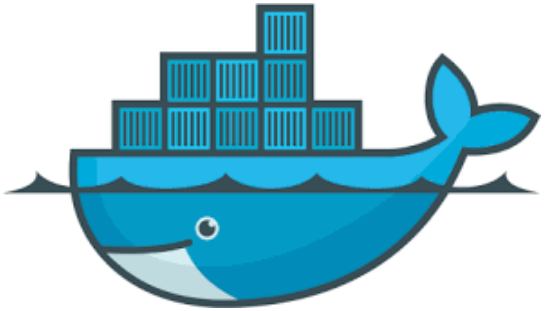


- **Process-Execution = Run Container.**
  - **Save + Safe to disk**
  - **Versions will grow…**

- **Runnable Containter: “Cattle”**
  - **It “only” needs to run.**
  - **Runtime-container, On-Demand, Microservice..**
  - **Cloud !**

Containers: Docker…

- **Relatively Simple (and Popular)**
- **Very Flexible, Scalable.**

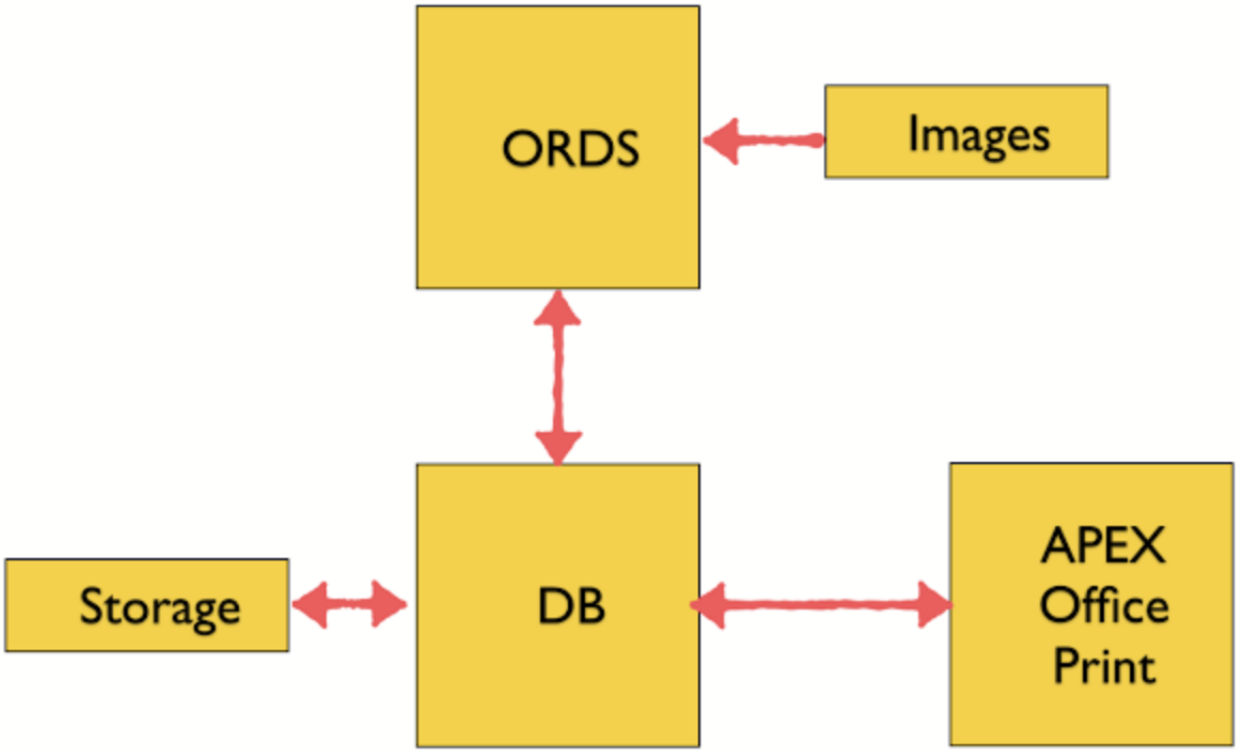


- **Good for:**
  - **Deployment – soo easy !**
  - **Processing – anywhere.**
- **Downsides:**
  - **Persistence (need volume-mapping; +/- NFS)**
  - **Control (decencies on underlying containers…)**
  - **predictability (… it got blown away..)**

Docker… (my opinion): good to deply, run, scale processing. But process-definition needs to be secured elsewhere, and data needs to be saved in database, not in docker-container. (win-2003 example

Docker : Example Roel…

Database & ORDS & AOP in Docker

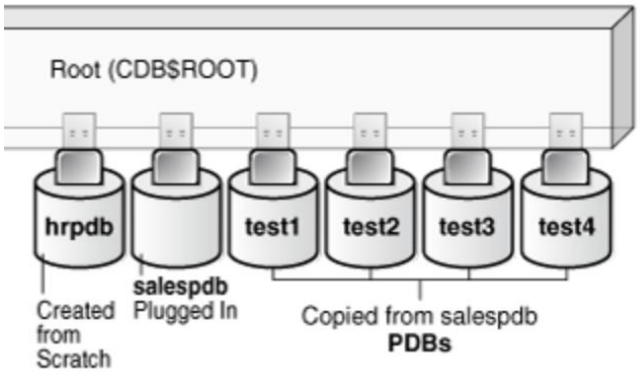


Now go and try it yourself!

Combine 4 containers to do some processing. Simple to understand. Fast-to-build.

Containers: Oracle PDBs…

- Relatively Simple (and Popular)
- Very Flexible, Scalable.

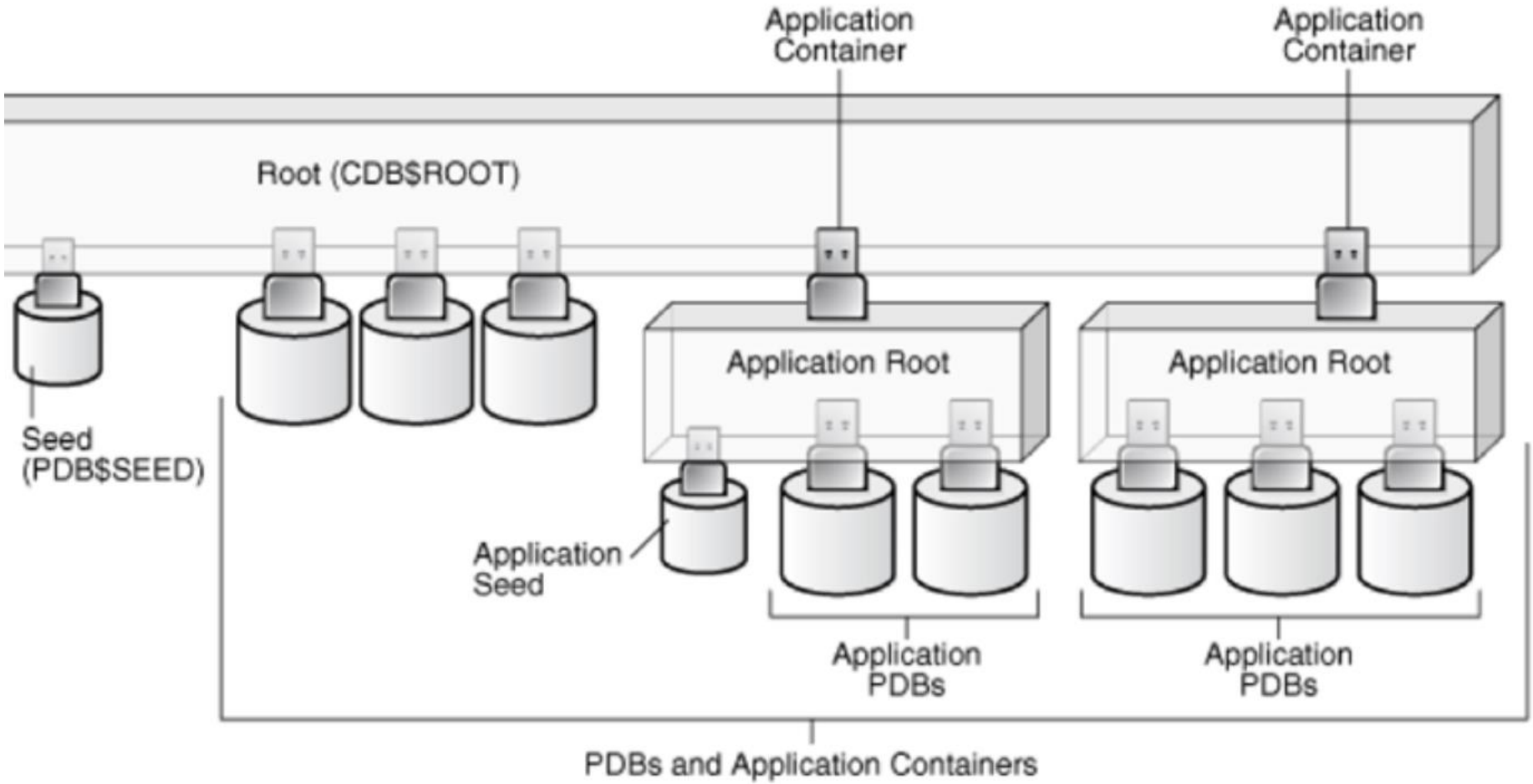


- Good for:
  - Database-work
  - Smart-DB!
  - Shows Real Potential.

- Downsides:
  - Oracle-Only.
  - Complex to administrate. (= work 4 DBA)
  - How to Explain to Dev-team… (= more work 4 DBA)

PDBs Example of “components”

PdVBV



The Oracle –PDB needs more complex admin (and Knowledge) – If time, point out the seed+root+root+instances.. Next slides shows the “syntax”..

PDBs Example of “admin”

Manage Applications

Applications are managed in the application root container using the APPLICATION clause of the ALTER PLUG

```
ALTER PLUGGABLE DATABASE APPLICATION
{ { app_name
    { BEGIN INSTALL 'app_version' [ COMMENT 'comment' ]
    | END INSTALL [ 'app_version' ]
    | BEGIN PATCH number [ MINIMUM VERSION 'app_version' ] [ COMMENT 'comment' ]
    | END PATCH [ number ]
    | BEGIN UPGRADE 'start_app_version' TO 'end_app_version' [ COMMENT 'comment' ]
    | END UPGRADE [ TO 'end_app_version' ]
    | BEGIN UNINSTALL
    | END UNINSTALL
    | SET PATCH number
    | SET VERSION 'app_version'
    | SET COMPATIBILITY VERSION { 'app_version' | CURRENT }
    | SYNC  }
  |
  { ALL SYNC }
}
```

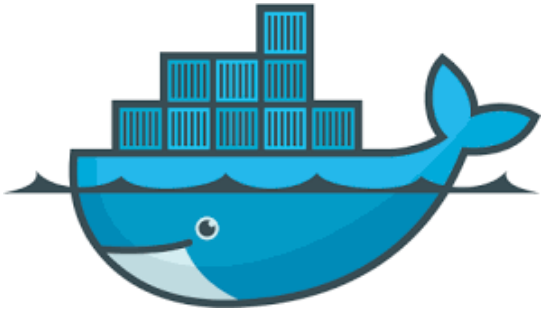

The syntax shows the many options – getting those “right” is not trivial
Note: Docker has similar complexity, but is more “generic”

4SYNERGY

Docker : My current view… 1/4

PdVBV

- **Recap: Docker is Good for:**
  - **Deployment – soo easy !**
  - **Processing – anywhere.**



- **Use Docker for “processing” Only**
  - **Keep No data (or very little data) inside Docker.**
  - **Ensure you can re-built your containers**

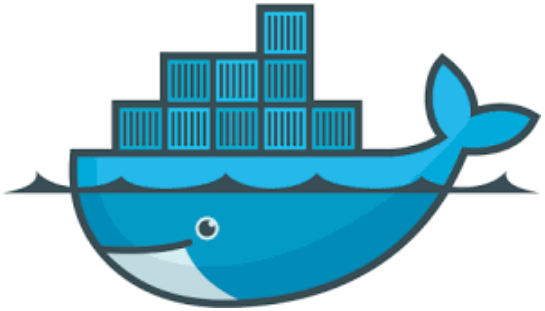
- **Docker-Container is “Cattle” (not pet)**
  - **It can be missed.**
  - **Swarm is appropriate name…**

4SYNERGY

Docker : Running Oracle… 2/4.

PdVBV

- **Oracle in Docker:**
  - Yes, because – soo easy !
  - Gerard Venzl…



- **My “suggestions”**
  - Reduce the container to “sofware only”
  - Map all data to a local (non docker) volume
  - Try storing the $ORACLE_HOME/dbs on local disk

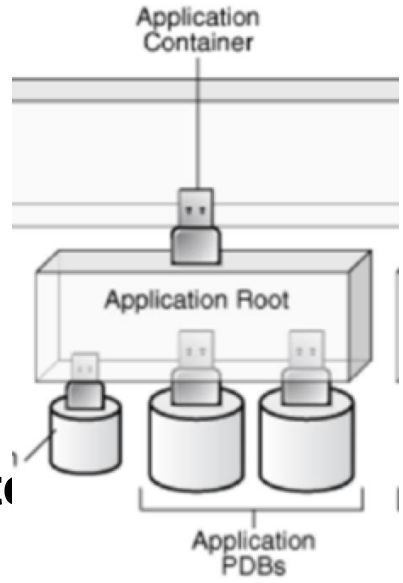
- **Oracle-in-Docker-Container is also “Cattle” (not pet)**
  - Make sure It can be missed.
  - It will “get blown away” a few times…

You can run a DB inside docker.. But don’t count on keeping the data.
Hence: make sure your “data” and your “process” are safely kept outside the containers.

PDBs: My current view… 3/4

PdVBV

- Recap: PBD-app-containers.
  - Oracle-specific
  - Complex to admin
  - Relatively “new”, unsure how it will develop.

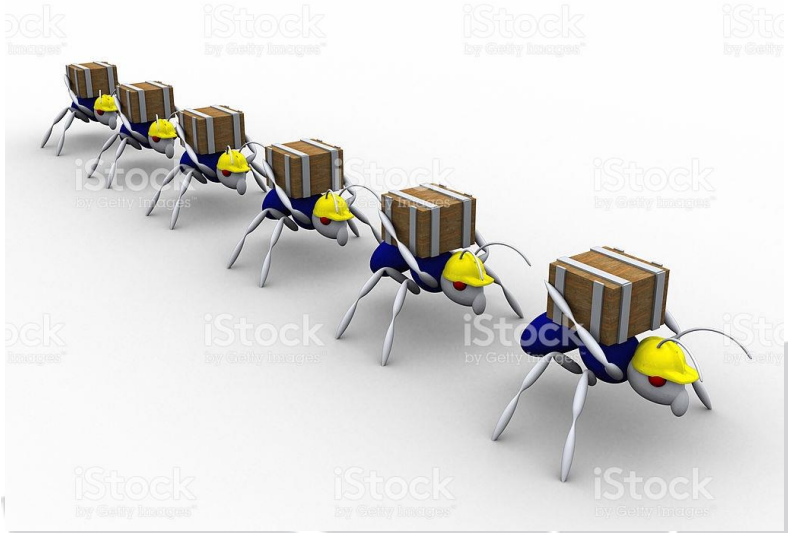


- PDB-Container is “Pet” (e.g. precious)
  - It will contain code – but not necessarily the “mast
  - It will need careful admin.

- Use PDB-app-containers only..
  - If you have the staff and the “knowledge”
  - (e.g. don’t use yet?)

Recap: Code + Data

- **Code, Process:**
  - **Not safe in a container.**
  - **Keep your “definitions” somewhere safe**
- **Data, information:**
  - **Not safe “inside” container**
  - **Store your data in a “database”**
  - **(or on a good filesystem)**
- **Processing:**
  - **Ideal for containers.**



4SYNERGY

Don’t Take my word for it…

PdVBV

Homework: Check your Dev-Ops and our Docker!

- Do you know “data” and “process” ?

Oracle-only / DBA => PDB

Generic IT => Docker (or similar)

(managers … under control?)

Simple Oracle Dba . Blogspot . com (my ramblings)

And keep it Simple!

Goethe: Limitation shows the Master.

Majority of times, I have been WRONG.
So go see for yourself – but don’t complicate life. “In der Beschrankung zeight sich der Meister”

Quick Q & A (3 min ;-) 3 .. 2 .. 1 .. Zero PdVBV

4SYNERGY

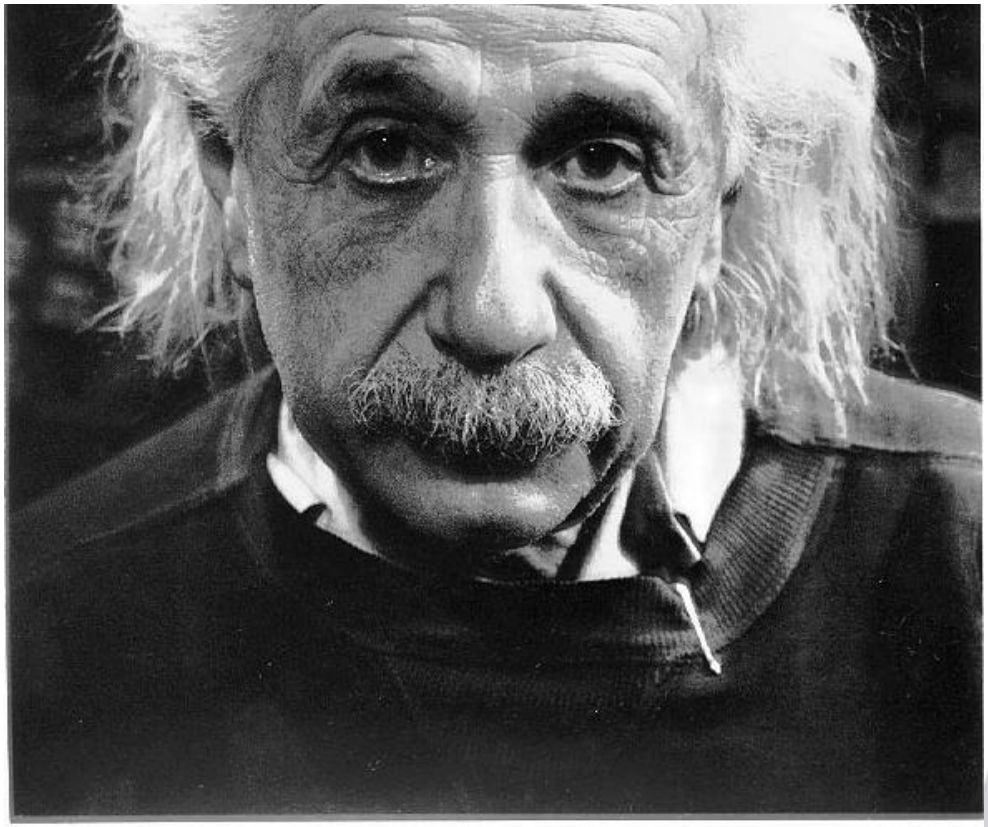
- Questions ?
- Reactions ?
- Experiences from the audience ?

Best part of most meetings: Question and Answer time. Discuss.
Teach me something: Tell me where you do NOT AGREE.

4SYNERGY

He got it …

PdVBV



“If you can't explain it simply, you don't understand it well enough”

As Simple as Possible, but not too simple
Simplicity is a Requirement – but Comlexity just sells better (EWD).

The very, very end…

- Intentionally left blank

A “Plummer” can build…

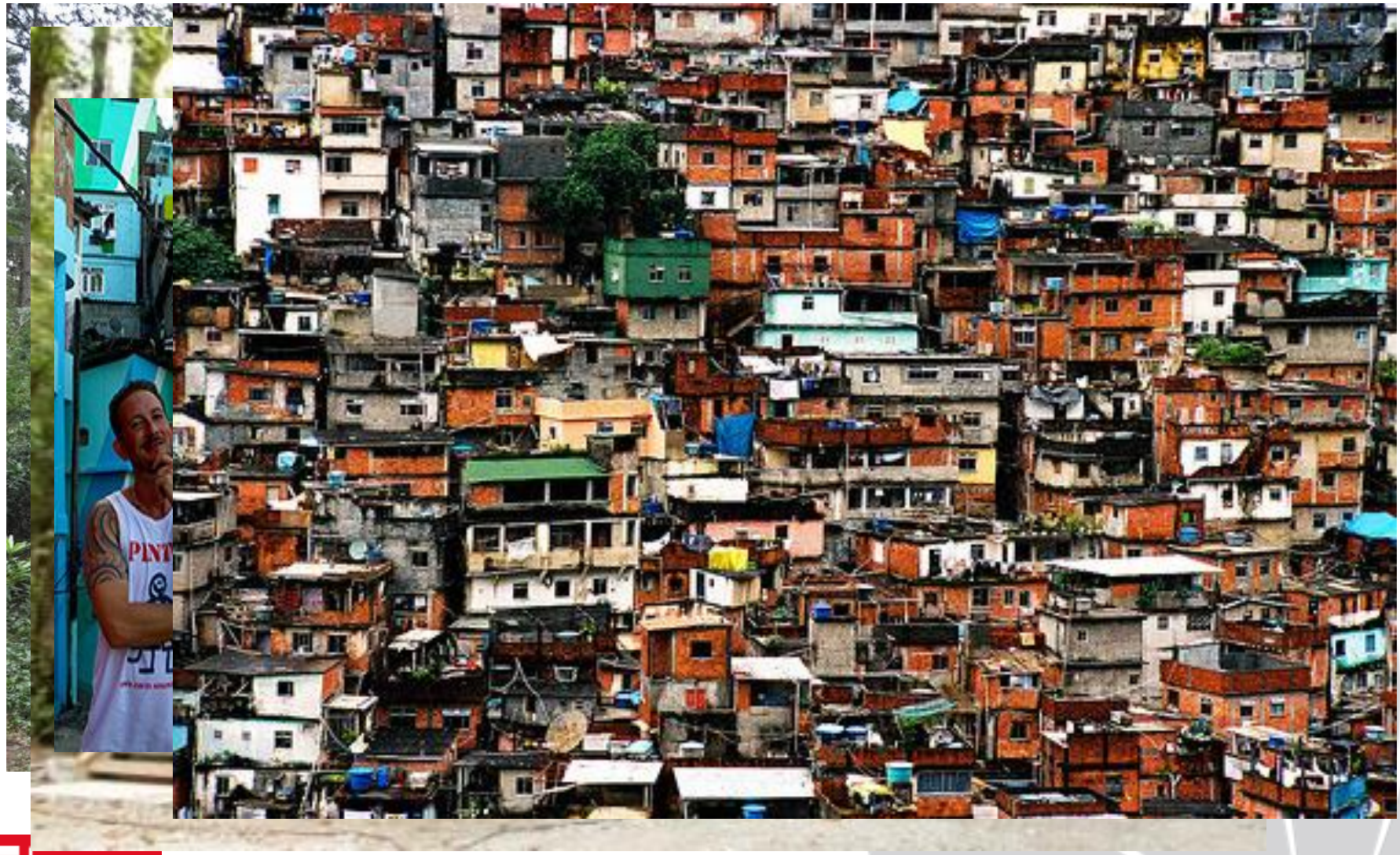
PdVBV

4SYNERGY

I told my boss/client in 1997 or so: we need more planning, more “structure”, we need an architect??
I did not know what would go wrong…

More + Complex houses…

PdVBV



But houses and systems become more complex This is what happened to some IT shops…
And then the plummer (team) becomes overwoked…

We need some “structure”

PdVBV

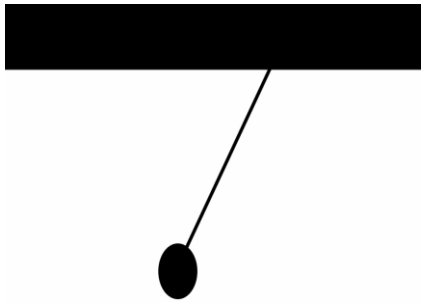
Regulation needed…
Architecture, structure…

4SYNERGY

Specializations … (90s)

PdVBV

- Analysists … + Managers…
- Designers… + Managers…
- Programmers… + Managers…
- Testers…. + Managers…
- DBA's … + Managers…
- System-Admins… + Managers…
- Storage-Admins… + M ….



Kafka comes along…

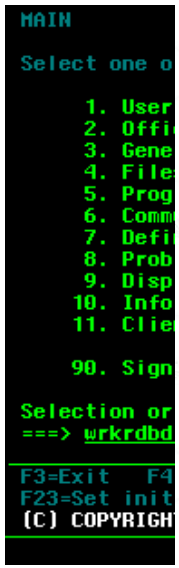
PdVBV

Process… Specialization.. Outsourcing... Offshouring.. Cobol Teams in India, QA-Process...
People Skills became less important, Proces!!

1997: the “Oracle Cowboy”

PdVBV

- **Replace...**
  - **Mainframe Maffia..**
- **By “new tech”**
  - **Unix**
  - **Oracle**
  - **Windows (3.11, 95...)**
  - **Forms 3.0, Forms 4.5, Designer/**
- **Small teams, Rapid Protytyping....**
  - **Users + Developers + DBA working Closely (again)**
  - **Flexible!**
  - **(DevOps?)**



We learned how to “eat the IDMS datasets, and run them on Oracle, Forms, Powerbuilder, and EBS…
Our finest moment: Replacing OTR and GL by EBS 10.7 – Yeah! (anyone remember the patching of EBS...)

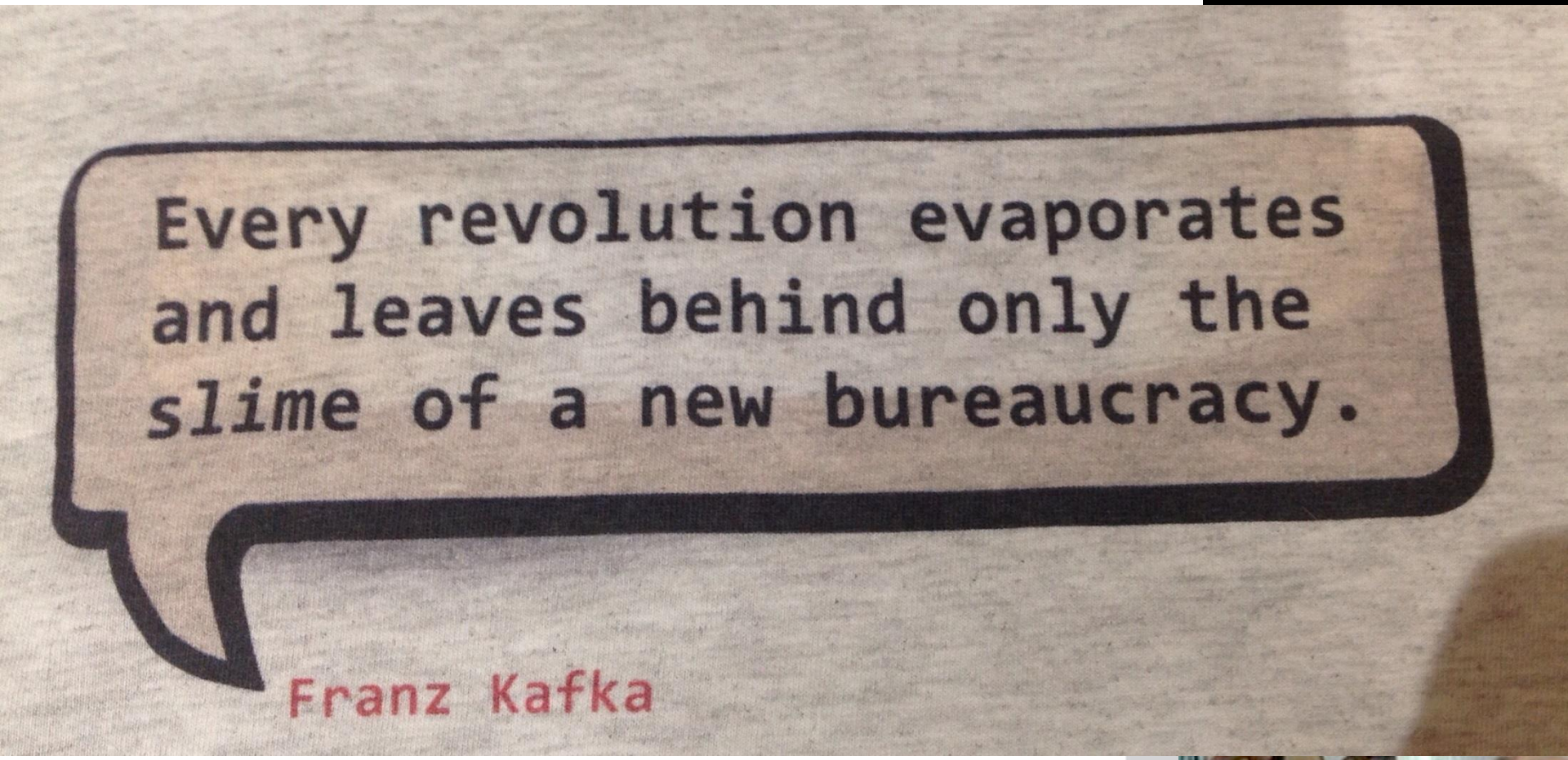
But as “new” systems grow….

- (you make some mistakes)
  - (interface anecdote… later)
- Specializations appear
  - Architects
  - DBA.
  - Windows-Admins...
- CM & QA processes …
  - Testing as a “discipline”
- Requirements + Process…
- Managers ++

So by the time the “new tech” becomes important enough.. It becomes a bureaucracy…
SLA’s, Tickets, Outsource, … Offshore. …

Again: Specialization…(new names)

PdVBV

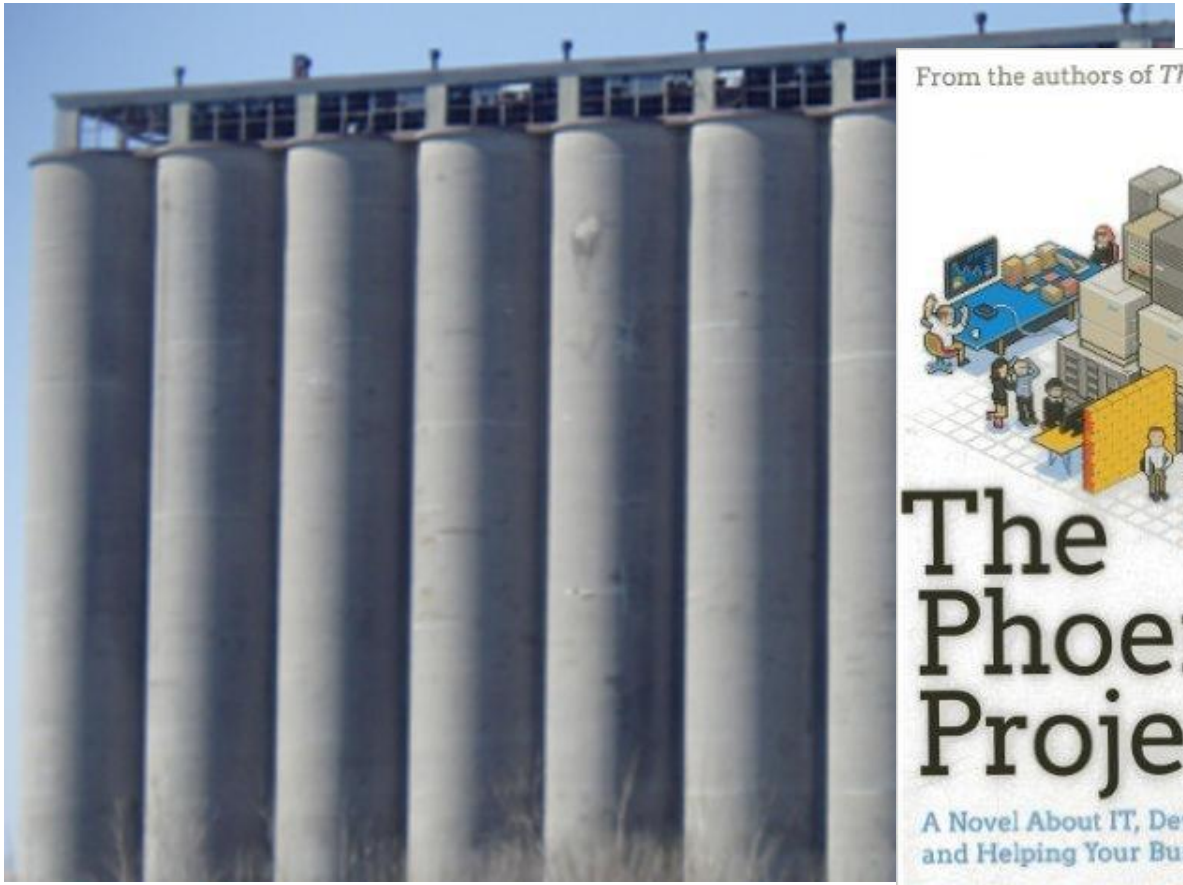


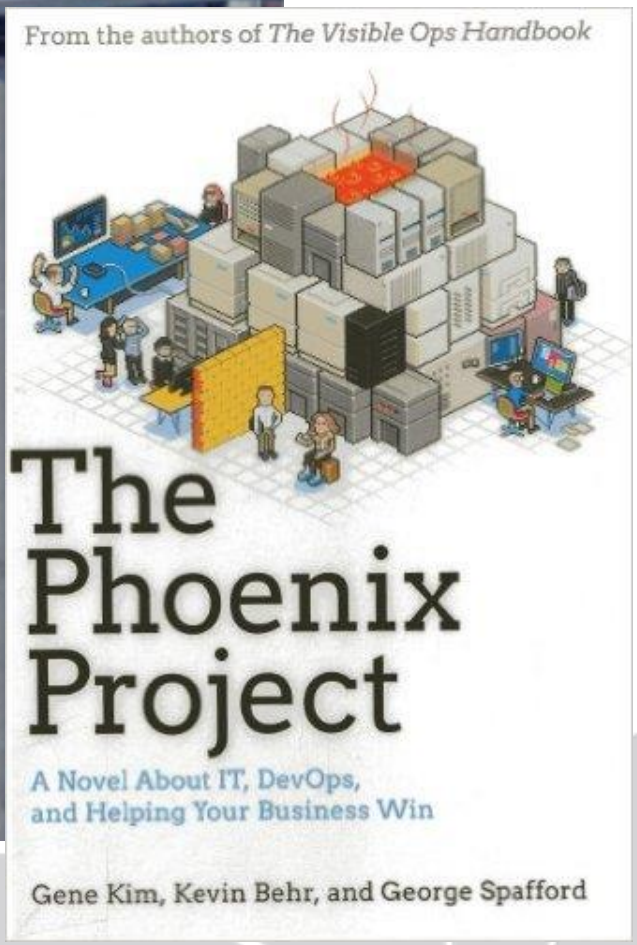
And guess what: Process… Specialization.. Outsourcing... Offshouring..
And this needs …. management…

And now … Silos versus DEV-OPS

PdVBV

- **How the Dev-Ops "Cowboys" see the old ways…**





Dev-Ops is about eliminating the Silo's … again… And they have a point…
That book is a recommended Read.. If you have also read "the goal", this is the next book...

Dev-Ops: The 3 Ideas

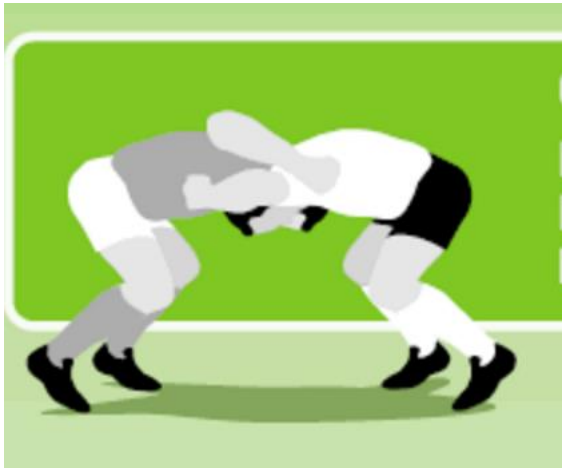
PdVBV

- **Goal: Support the Business..**
  - **Sounds familiar ?**
- **1. Collaboration**
  - **Multi-Disciplined team**
  - **Agile, Lean**
- **Shared Responsabilities…**
- **2. Automation**
  - **Fast turnaround (build, test, deploy)**
  - **Provisioning (cloud..)**
- **3. Continuous Delivery**
  - **Always ready to deploy**

Dev-Ops (wikipedia states ….)
And they have a point…

Dev-Ops: 1. Collaboration

PdVBV



- **Multi-Disciplined Teams**
  - They seem to really mean it now...
- **Educate Eachother !**
  - You must learn from team members
- **From Different Backgrou**
  - Forget hobby-projects (+ fo
  - Focus on Customer + Tean
- **Jargon: Scrum, Standup.**
  - Burndown-chart,
  - Backlog,
  - Sprint...

So what is different this time: serious mulit-discipline, communication..
lots of jargon.. Worst Case: nothing gets done...

Dev-Ops: 2. Automation

- Infrastructure = Code
- Configuration = Code

- Fast turnaround
  - Provisioning – add more of... Anything.
  - (#cloud)

- Tools..
  - Puppet/Chef/Jenkins... Many.
  - “Dockerize”
  - Github

- Push-Button ...
  - Build (what about the design?)
  - Test (... rrrreally ????)
  - Deploy...


```
$ puppet resource custom_package
custom_package { 'GConf2':
  ensure  => 'present',
  version => '2.14.0',
}
custom_package { 'MAKEDEV':
  ensure  => 'present',
  version => '3.23',
}
custom_package { 'NetworkManager':
  ensure  => 'present',
  version => '0.7.0',
}
custom_package { 'NetworkManager-glib':
  ensure  => 'present',
  version => '0.7.0',
}
```

To error is human, but to really mess things up you need a computer…(Debug a Puppet file....!)
Biggest challeng: Autmatic Testing… (test is never complete, and you cannot test Q into a Prod)

Dev-Ops: 3. Continuous

PdVBV

4SYNERGY

- “Continuous” Delivery
  - Pipeline vs tanktruck
  - Pets vs Cattle
- Automated + Permanent Testing..
  - QA = Challenge..
  - Good automated testing..
- Tools..
  - Jenkins/Hudson (workflow + test..?)
- Question: what is “current state” ?
  - And how to stop the flow... ?

Continuous … Allow for quick changes, permanent changes.. … (Test ? QA?)
Biggest challeng: Autmatic Testing… We are stilllearning these tricks...

Dev-Ops : Design…

PdVBV

Bduf is the new on-prem… BDUF = old, bad… But Without minimum Design… Chaos
Probably: Lightweiht “process” and Resonable Data-Model (Key Structure, Data-Life-Cycle (ILM), Archiving …

Dev-Ops : Knowledge, Skills…

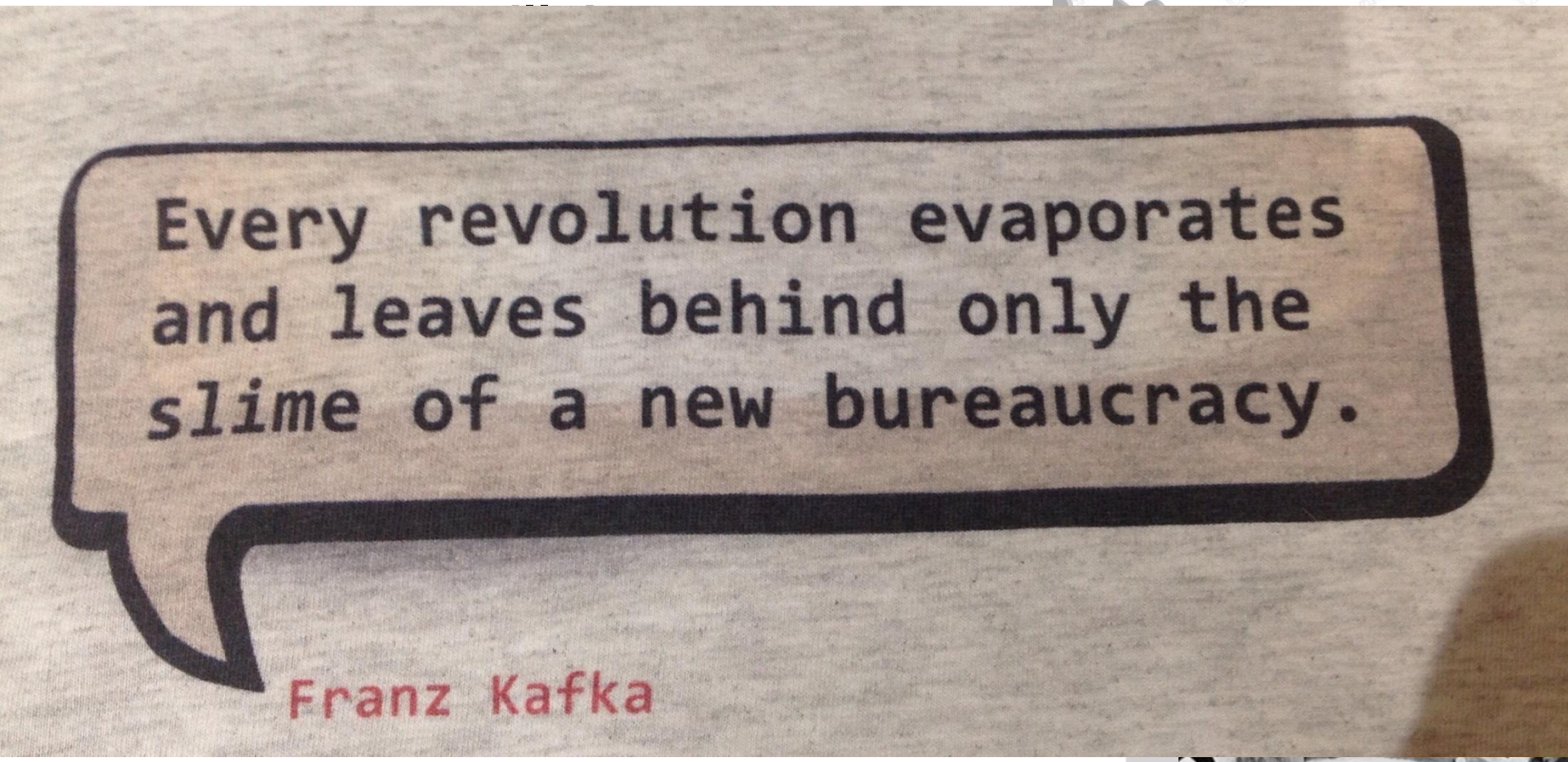
PdVBV

- **Use of Concepts and Tools**
- **Specialization: Supposedly …Bad..**
  - **Specialists don't communicate well.**
  - **For "the hammer", everything is a nail…**
- **Without a Knowledge and "skills"**
  - **Clumsy stuff results.**
- **I think: SKUF = probably needed..**
  - **Some Knowledge/Training Up Front…**

Dev-Ops doesn't favour "specialists", but you need a minimum level of knowledge and skill to accomplish anything. Probably: Lightweiht knowlegea and some "experience" is needed.

You. Will. Do. Dev-OPS…

PdVBV



The positive sides : Keep them.. The potential problems: Try to fix.
Make sure teams had "knowing", and make sure automation doesn't control you..

Short : Be like him …

PdVBV

Be Flexible, helpful, communicative.
(but mind : Architecture, structure…)

Short : Avoid this…

PdVBV

Don't become a slave… and Avoid bossy-bosses.
(mind: some Architecture, some structure, is needed)

Then: Create this…?

PdVBV

Regulation needed…
Architecture, structure…

4SYNERGY

PdVBV

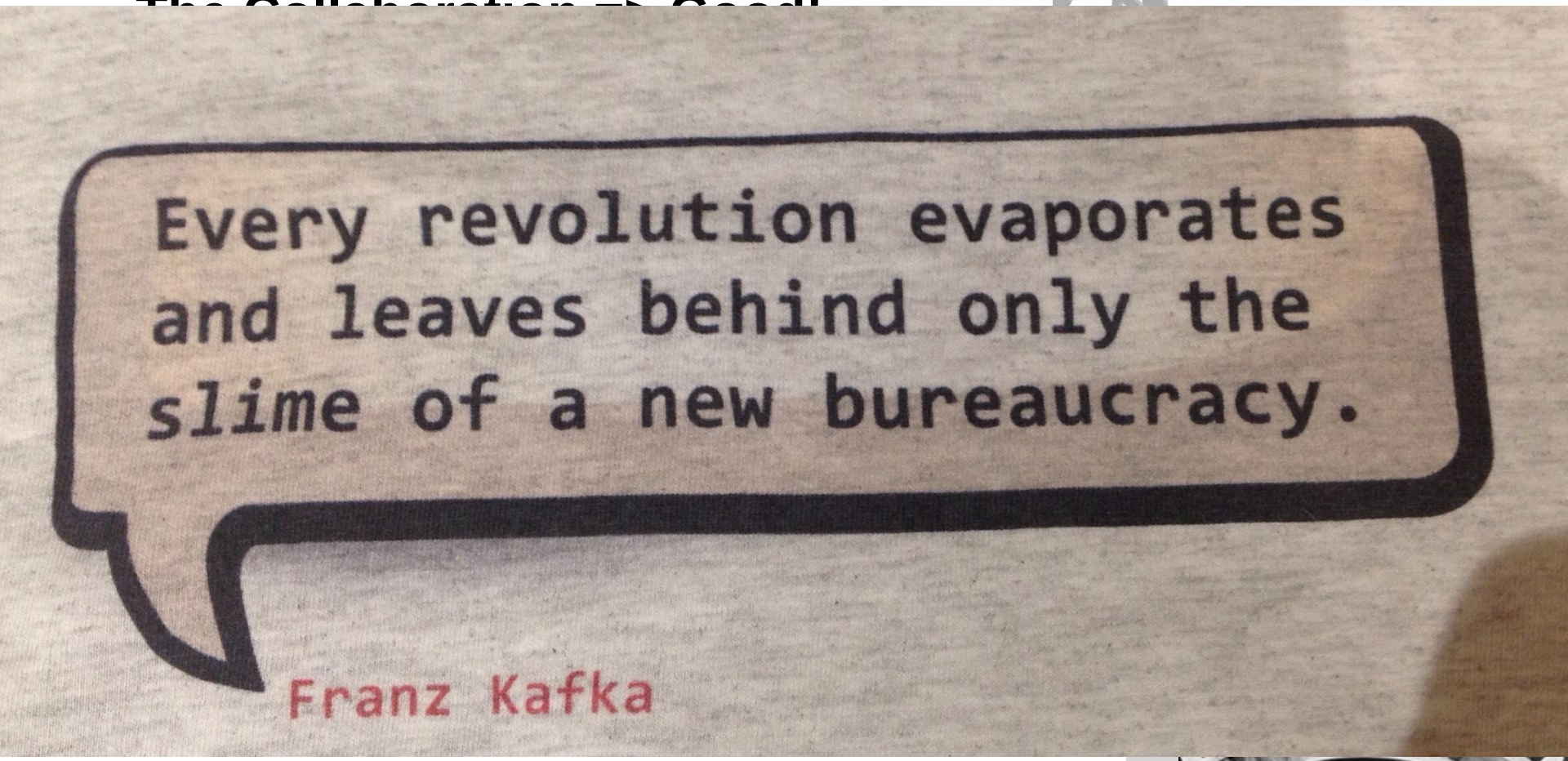
The very, very end…

- This tech is still ahead of any pg-wal or barman, but oracle want too much…

- #poug17 speaker:
- let me exp… -hic- explllllain.
- (bad joke again)

Dev-Ops : Good !

PdVBV



The positive sides : Keep them.. The potential problems: Try to fix.
Make sure teams had "knowing", and make sure automation doesn't control you..