APEX Application Lifecycle

with feature based deployment

Hello!

I am Richard Martens

Welcome to this presentation.

You can find me at

- @rhjmartens
- smart4solutions.nl/blog

smart4 solutions IT's in our nature

Oracle ACE Pro

Oracle ACE Program

500+ technical experts helping peers globally

The **Oracle ACE Program** recognizes and rewards community members for their technical and community contributions to the Oracle community

3 membership tiers

Oracle ACE Director | Oracle ACE Pro | Oracle ACE Associate

For more details on Oracle ACE Program:
ace.oracle.com

Oracle ACE

Nominate
yourself or someone you know:

ace.oracle.com/nominate

Connect: **aceprogram_ww@oracle.com** Facebook.com/OracleACEs @oracleace

Agenda

- Objectives
- Requirements
- Development Methodology
- Solution
- Branching Strategies
- Workflow
- Project Folder Structure
- Deployment
- Assumptions
- Challenges

Objectives

Our customers expect and need a way to deploy an APEX application based on a single feature

Easy deployment throughout the application lifecycle

Development of new features may not be stopped

Objectives

- *Our customers expect and need a way to deploy an APEX application based on a single feature*

- *Easy deployment throughout the application lifecycle*

- *Development of new features may not be stopped*

Requirements

Requirements

- No dependency on specific project-management tools
  - Jira, Wrike, MS-Project
  - Confluence, Sharepoint
  - Bitbucket, Gitlab, Github
- Using our standard development methodology
- 1 core schema
  - for data and business-rules
- 1 apex schema
  - for APEX specific views and packages
    - This is the schema our APEX workspace looks at also known as “parsing schema”
- Optional a VPD layer or other “external” schemas involved
- DB deployment-tool using regular SQL / Command scripts that every DB developer understands (learning curve)

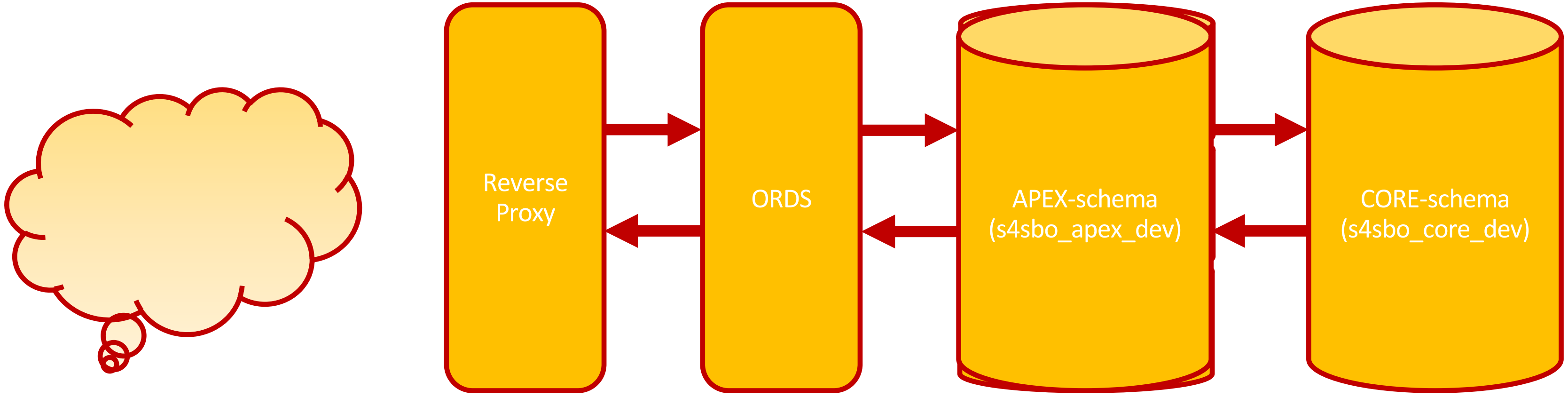
Development Methodology

Standard infrastructure

- The basic idea
  - APEX does not “touch” tables directly
  - A view / package for each individual APEX page
  - A view / package for each individual REST module
- We pushed it a bit further
  - A view for each individual region
  - LOV’s compilable in the database!
  - Naming standards for objects:
    - *prefix*_app_page_description_*suffix*

  - UI and Business should not be mingled, Modularising code
  - Enforces the question: what is business logic, what is not

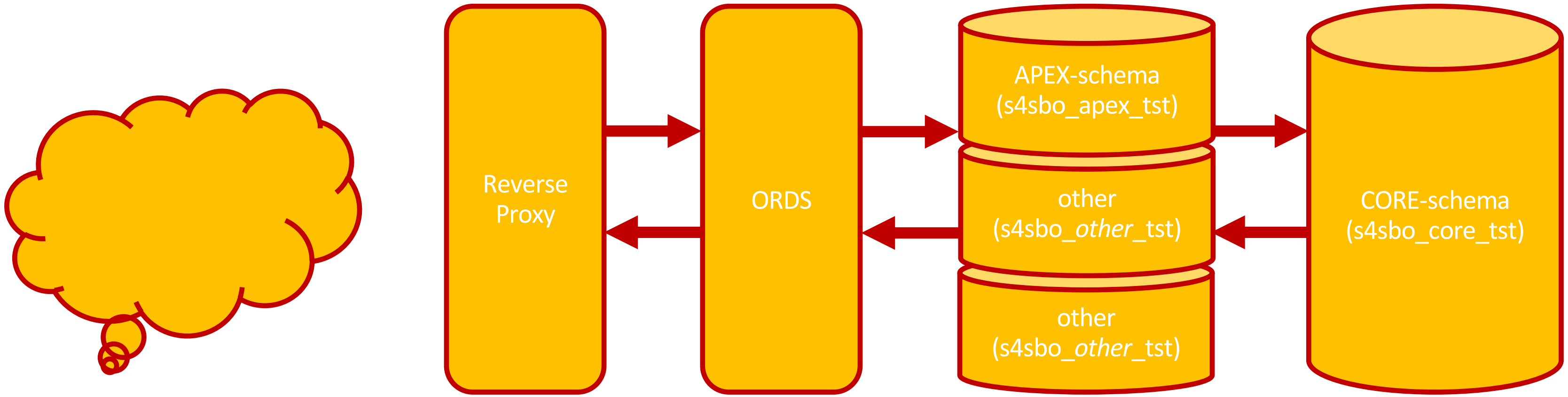
Standard infrastructure dev



static project files
static APEX files (/i)

Each new interface gets its own schema

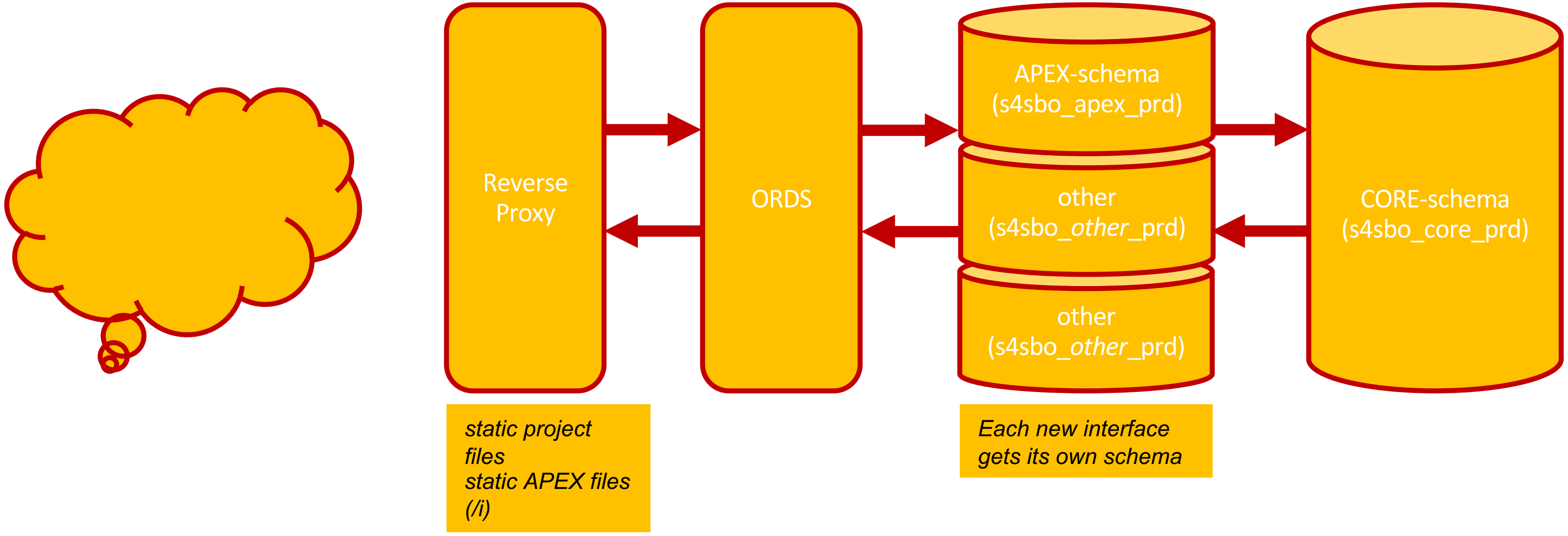
Standard infrastructure tst


Reverse Proxy
ORDS
APEX-schema (s4sbo_apex_tst)
other (s4sbo_*other*_tst)
other (s4sbo_*other*_tst)
CORE-schema (s4sbo_core_tst)

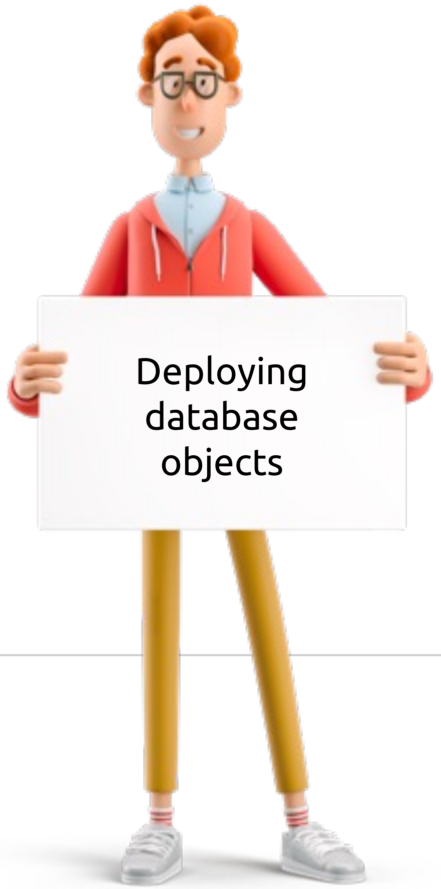
static project files
static APEX files (/i)

Each new interface gets its own schema

Standard infrastructure prd


Reverse Proxy
ORDS
APEX-schema (s4sbo_apex_prd)
other (s4sbo_*other*_prd)
other (s4sbo_*other*_prd)
CORE-schema (s4sbo_core_prd)
static project files
static APEX files (/i)
Each new interface gets its own schema

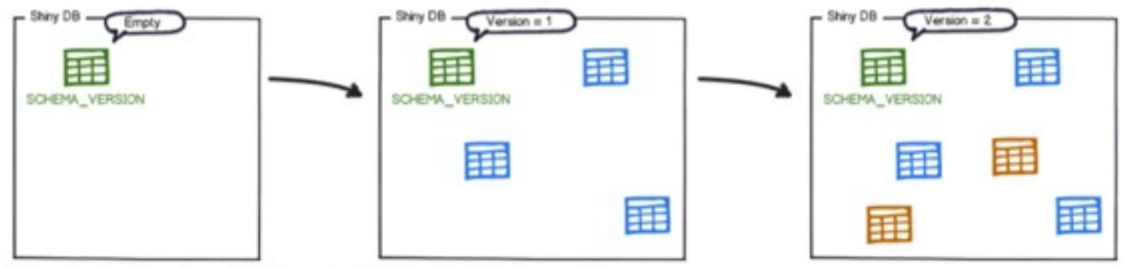
Solution



Introducing Flyway CE

Flyway run

1. Scan folder-structure
2. Execute Versioned files by version increment (ie alphabetically)
3. Execute Repeatable files alphabetically



As each migration gets applied, the metadata table is updated accordingly:

schema_version

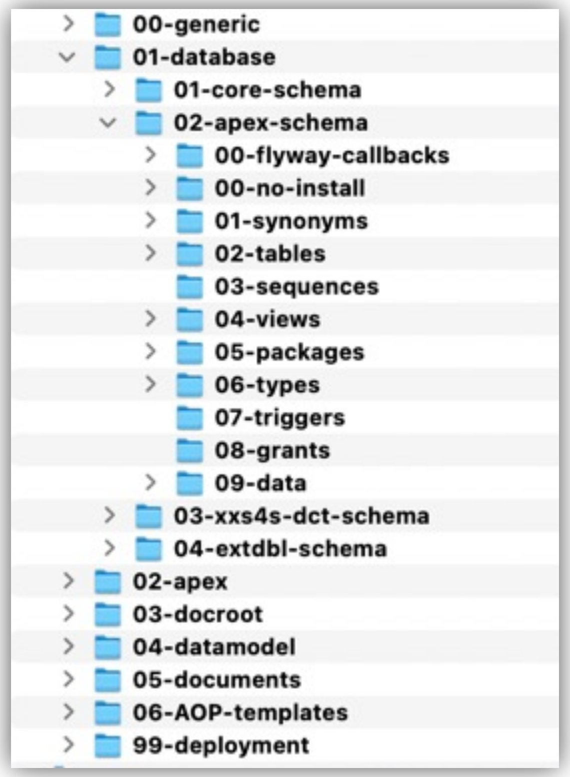
installed_rank	version	description	type	script	checksum	installed_by	installed_on	execution_time	success
1	1	Initial Setup	SQL	V1__Initial_Setup.sql	1996767037	axel	2016-02-04 22:23:00.0	546	true
2	2	First Changes	SQL	V2__First_Changes.sql	1279644856	axel	2016-02-06 09:18:00.0	127	true

Project Folder Structure

Project Folder Structure

- Each db-object deserves a file
  - Separate package (or type) spec and body
  - Project dependant
    (ie indexes at table creation)
- Predefined structure
- Naming method for Flyway
  - R__xxx
  - VYYYYMMDDHHMI__xxx

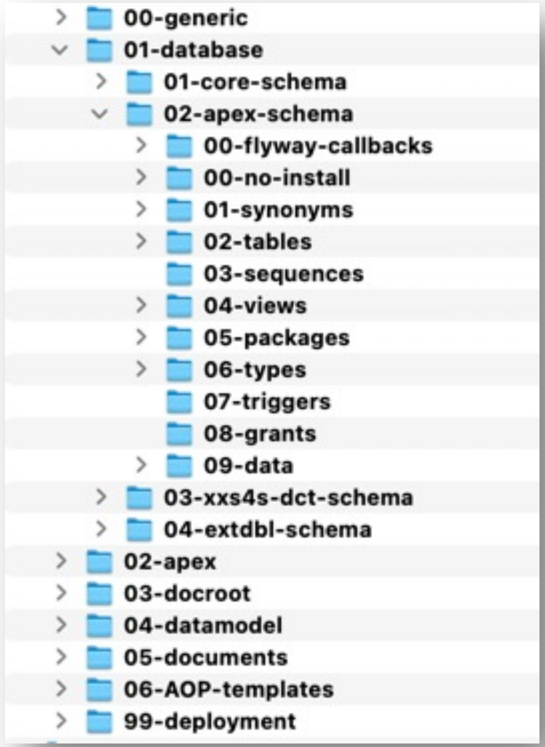
- We're not using an "install_all" script !



```
> 00-generic
v 01-database
    > 01-core-schema
    v 02-apex-schema
        > 00-flyway-callbacks
        > 00-no-install
        > 01-synonyms
        > 02-tables
          03-sequences
        > 04-views
        > 05-packages
        > 06-types
          07-triggers
          08-grants
        > 09-data
    > 03-xxs4s-dct-schema
    > 04-extdbl-schema
> 02-apex
> 03-docroot
> 04-datamodel
> 05-documents
> 06-AOP-templates
> 99-deployment
```

Flyway at work

- Flyway CE
  - only allows ".sql" file-extensions
  - No SQL*Plus commands
- "prep4flyway.sh" renames all files
  - Extensions to ".sql"
  - Prefixes files according to folder-number
  - Enforces order of spec and body
- Flyway only sees the renamed files



```
> 00-generic
v 01-database
  > 01-core-schema
  v 02-apex-schema
    > 00-flyway-callbacks
    > 00-no-install
    > 01-synonyms
    > 02-tables
      03-sequences
    > 04-views
    > 05-packages
    > 06-types
      07-triggers
      08-grants
    > 09-data
  > 03-xxs4s-dct-schema
  > 04-extdbl-schema
> 02-apex
> 03-docroot
> 04-datamodel
> 05-documents
> 06-AOP-templates
> 99-deployment
```

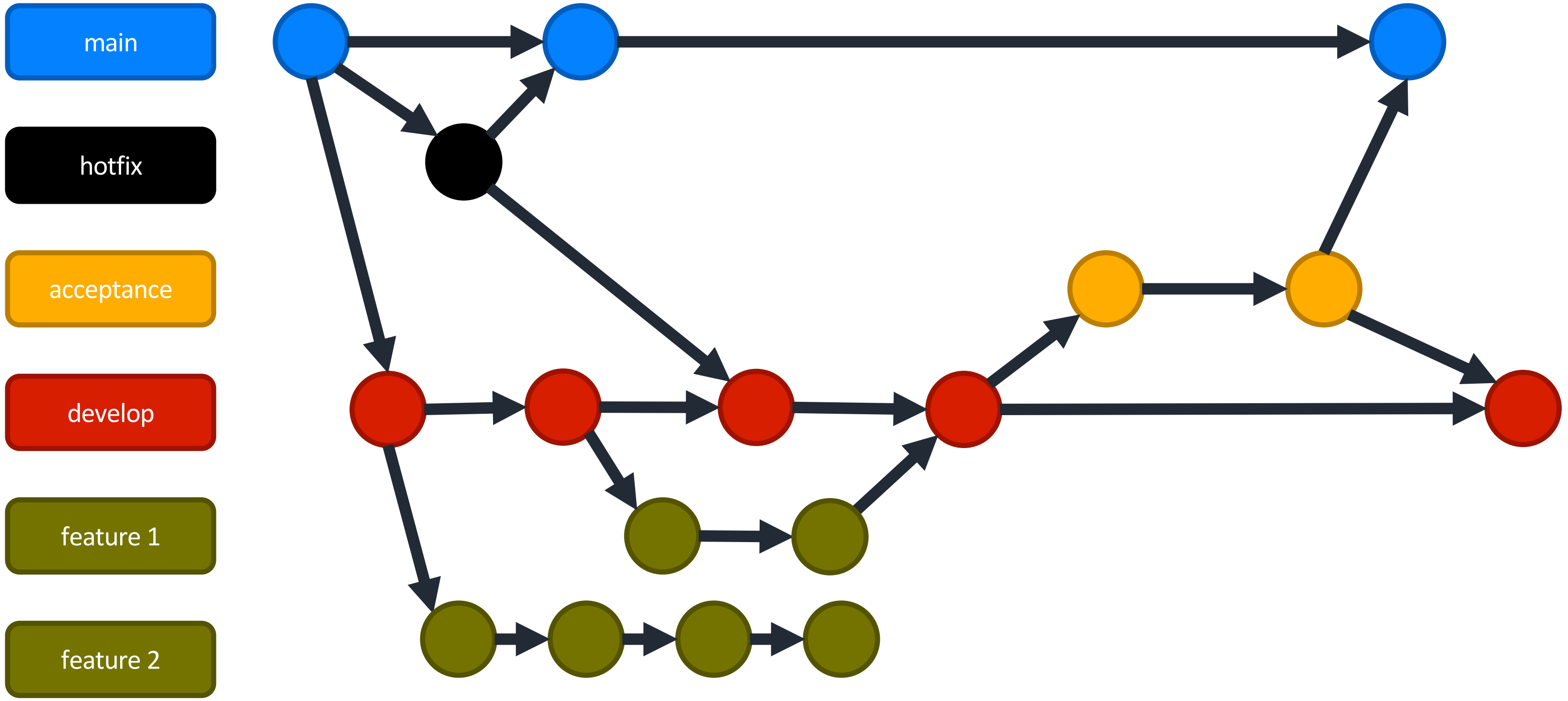
Flyway at work

- We can now build all our DB objects
- What do we feed Flyway to get a proper builds ?

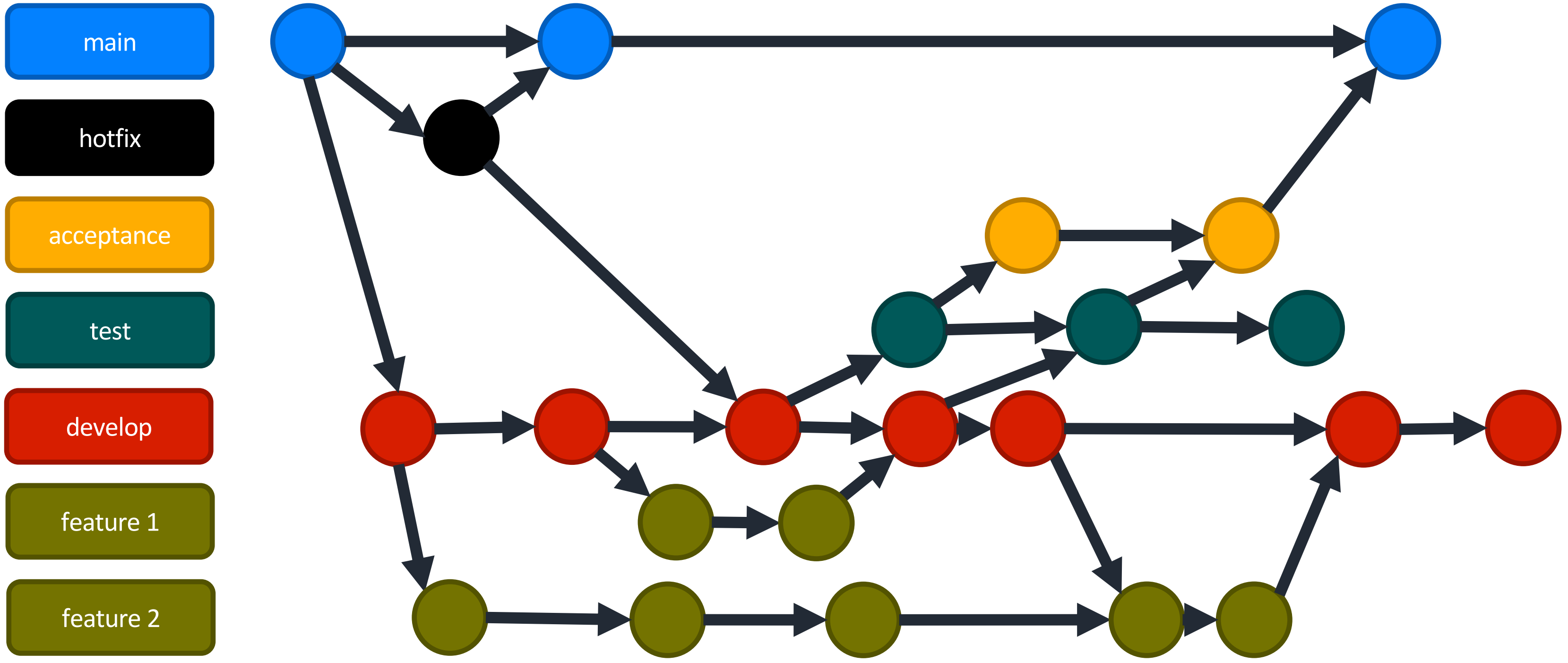
- Introducing: GIT and branching

Branching Strategies

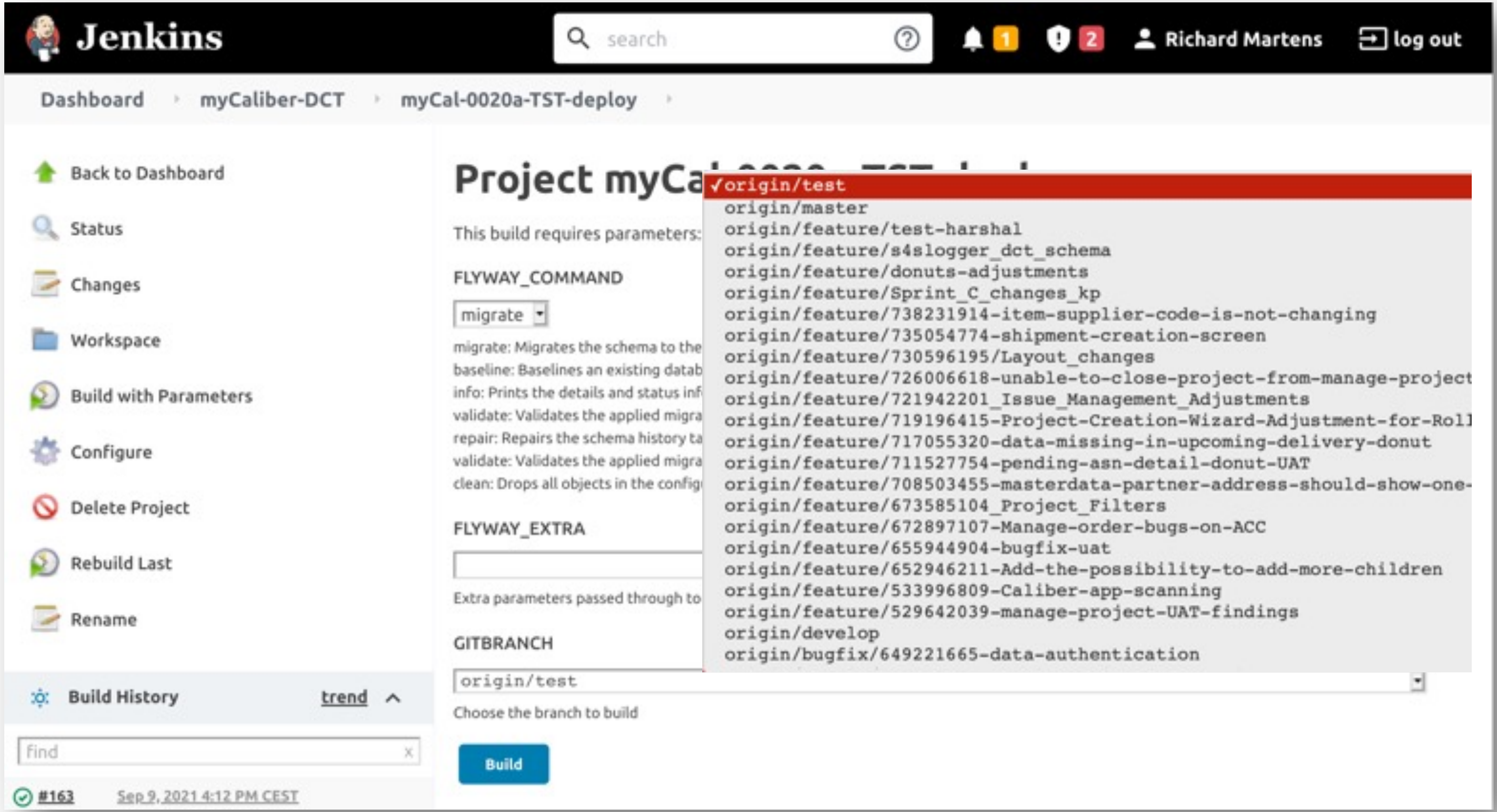
Branching Strategies GITFLOW



Branching Strategies S4S-FLOW



Build Server Jenkins



Jenkins

search

1 2 Richard Martens log out

Dashboard › myCaliber-DCT › myCal-0020a-TST-deploy ›

Back to Dashboard
Status
Changes
Workspace
Build with Parameters
Configure
Delete Project
Rebuild Last
Rename

Build History trend

find

#163 Sep 9, 2021 4:12 PM CEST

Project myCa

This build requires parameters:

FLYWAY_COMMAND

migrate

migrate: Migrates the schema to the
baseline: Baselines an existing datab
info: Prints the details and status inf
validate: Validates the applied migra
repair: Repairs the schema history ta
validate: Validates the applied migra
clean: Drops all objects in the config

FLYWAY_EXTRA

Extra parameters passed through to

GITBRANCH

```
✓origin/test
 origin/master
 origin/feature/test-harshal
 origin/feature/s4slogger_dct_schema
 origin/feature/donuts-adjustments
 origin/feature/Sprint_C_changes_kp
 origin/feature/738231914-item-supplier-code-is-not-changing
 origin/feature/735054774-shipment-creation-screen
 origin/feature/730596195/Layout_changes
 origin/feature/726006618-unable-to-close-project-from-manage-project
 origin/feature/721942201_Issue_Management_Adjustments
 origin/feature/719196415-Project-Creation-Wizard-Adjustment-for-Roll
 origin/feature/717055320-data-missing-in-upcoming-delivery-donut
 origin/feature/711527754-pending-asn-detail-donut-UAT
 origin/feature/708503455-masterdata-partner-address-should-show-one-
 origin/feature/673585104_Project_Filters
 origin/feature/672897107-Manage-order-bugs-on-ACC
 origin/feature/655944904-bugfix-uat
 origin/feature/652946211-Add-the-possibility-to-add-more-children
 origin/feature/533996809-Caliber-app-scanning
 origin/feature/529642039-manage-project-UAT-findings
 origin/develop
 origin/bugfix/649221665-data-authentication
```

origin/test

Choose the branch to build

Build

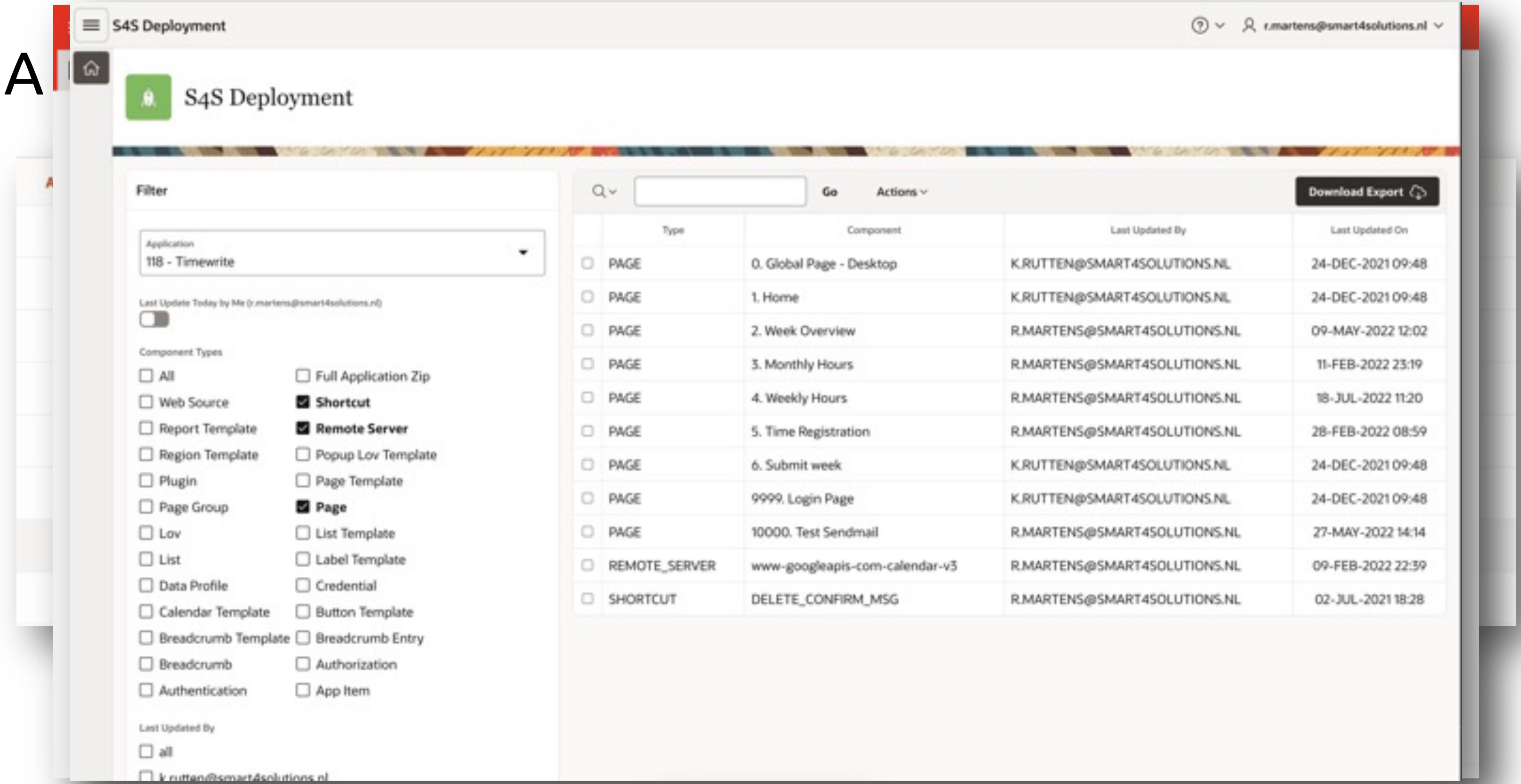
Deploy DB Objects

- We're now able to deploy a specific branch
- Including all DB-objects
- Including data
- The deployment contains the features that were merged into the build branches (Test, Acc, Prod)

- What is missing?

Integrating APEX into the build

- A



S4S Deployment

r.martens@smart4solutions.nl

S4S Deployment

Filter

Application: 118 - Timewrite

Last Update Today by Me (r.martens@smart4solutions.nl)

Component Types

☐ All	☐ Full Application Zip
☐ Web Source	☑ Shortcut
☐ Report Template	☑ Remote Server
☐ Region Template	☐ Popup Lov Template
☐ Plugin	☐ Page Template
☐ Page Group	☑ Page
☐ Lov	☐ List Template
☐ List	☐ Label Template
☐ Data Profile	☐ Credential
☐ Calendar Template	☐ Button Template
☐ Breadcrumb Template	☐ Breadcrumb Entry
☐ Breadcrumb	☐ Authorization
☐ Authentication	☐ App Item

Last Updated By

☐ all

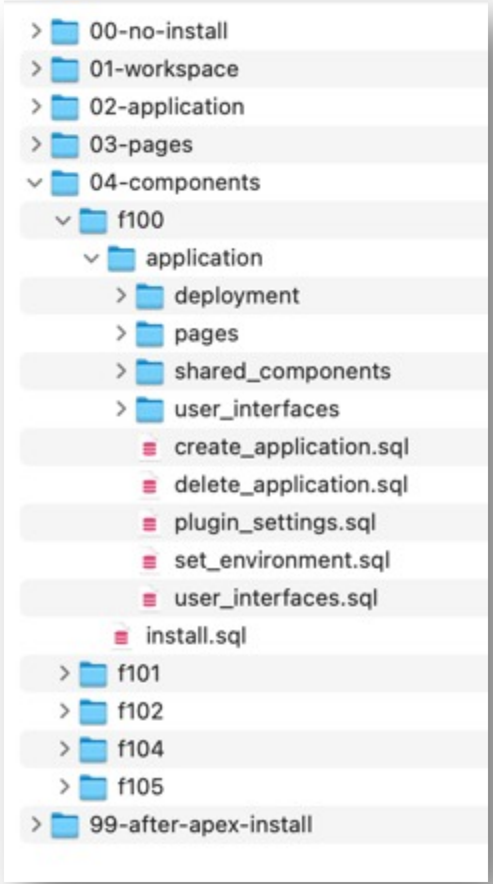
☐ k.rutten@smart4solutions.nl

Go Actions Download Export

	Type	Component	Last Updated By	Last Updated On
☐	PAGE	0. Global Page - Desktop	K.RUTTEN@SMART4SOLUTIONS.NL	24-DEC-2021 09:48
☐	PAGE	1. Home	K.RUTTEN@SMART4SOLUTIONS.NL	24-DEC-2021 09:48
☐	PAGE	2. Week Overview	R.MARTENS@SMART4SOLUTIONS.NL	09-MAY-2022 12:02
☐	PAGE	3. Monthly Hours	R.MARTENS@SMART4SOLUTIONS.NL	11-FEB-2022 23:19
☐	PAGE	4. Weekly Hours	R.MARTENS@SMART4SOLUTIONS.NL	18-JUL-2022 11:20
☐	PAGE	5. Time Registration	R.MARTENS@SMART4SOLUTIONS.NL	28-FEB-2022 08:59
☐	PAGE	6. Submit week	K.RUTTEN@SMART4SOLUTIONS.NL	24-DEC-2021 09:48
☐	PAGE	9999. Login Page	K.RUTTEN@SMART4SOLUTIONS.NL	24-DEC-2021 09:48
☐	PAGE	10000. Test Sendmail	R.MARTENS@SMART4SOLUTIONS.NL	27-MAY-2022 14:14
☐	REMOTE_SERVER	www-googleapis-com-calendar-v3	R.MARTENS@SMART4SOLUTIONS.NL	09-FEB-2022 22:39
☐	SHORTCUT	DELETE_CONFIRM_MSG	R.MARTENS@SMART4SOLUTIONS.NL	02-JUL-2021 18:28

Integrating APEX into the build

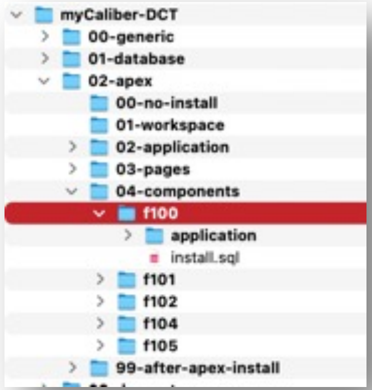
- The generated ZIP contains an installer for the exported components



```
> 00-no-install
> 01-workspace
> 02-application
> 03-pages
v 04-components
  v f100
    v application
      > deployment
      > pages
      > shared_components
      > user_interfaces
        create_application.sql
        delete_application.sql
        plugin_settings.sql
        set_environment.sql
        user_interfaces.sql
      install.sql
  > f101
  > f102
  > f104
  > f105
> 99-after-apex-install
```

Integrating APEX into the build

- The developer unzips the ZIP into the GIT-branch
- f100.zip unpacks in f100 folder
  - Mac users need a tool to merge folders
- Application folder contains entire application but only exported files are “new”
- The included installer cannot be used
- Extra build-step: “create_cmp_install.sh”
  - Generates a new application-installer
- Extra build-step to deploy static files (*.js, *.png etc)

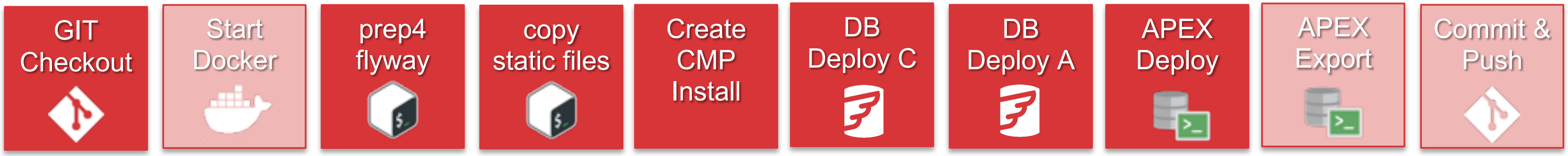


```
myCaliber-DCT
  00-generic
  01-database
  02-apex
    00-no-install
    01-workspace
    02-application
    03-pages
    04-components
      f100
        application
        install.sql
      f101
      f102
      f104
      f105
    99-after-apex-install
```

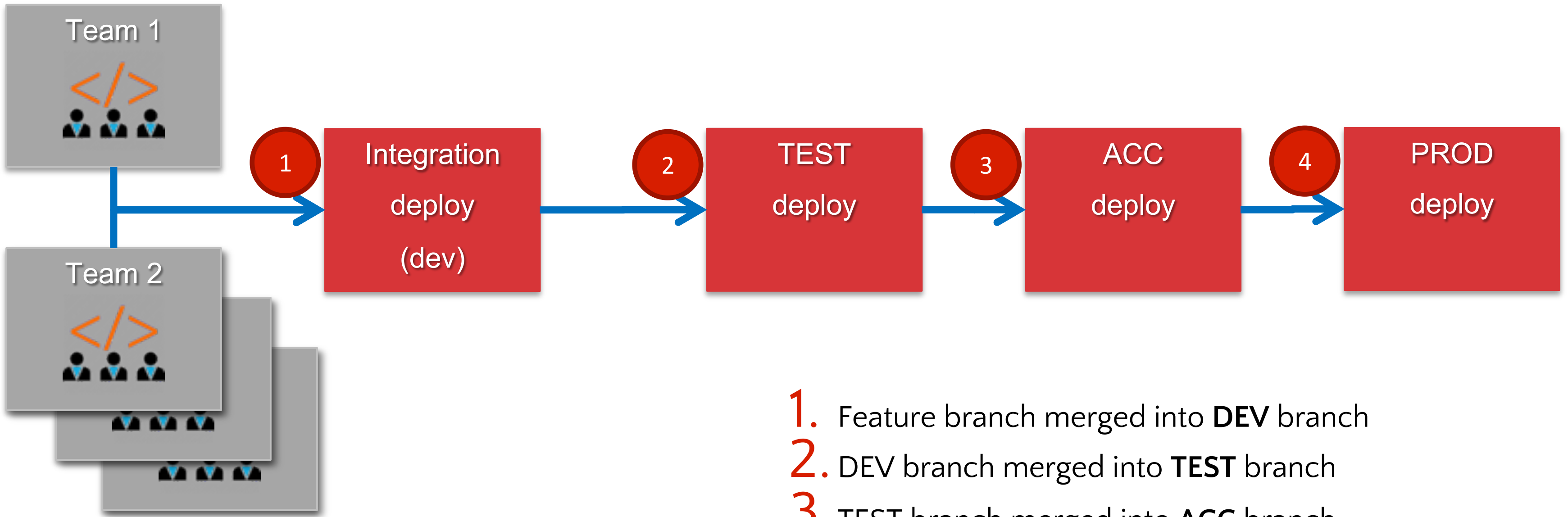
Deployment

- The deployment process is as follows (DTAP)
  - D for Docker, an environment that gets reset (to the latest production version) before each build
  - Because we build all environments automatically we have tested our deployment-scripts 3! Times by the time we deploy to production

1. Checkout the branch
2. Start “prep4flyway” script for CORE (1st) and APEX (2nd) schema
3. Copy static files (rsync)
4. Start “create_cmp_install” script for each application (D & T)
5. Invoke flyway for CORE (first) and APEX (second) schema
    1. Uses callbacks to check for invalids, invalids break the build!
6. Start the generated install script from step 4 (D & T) otherwise full-app install
7. Full App export (T)
8. Place full-app export on the Test Branch

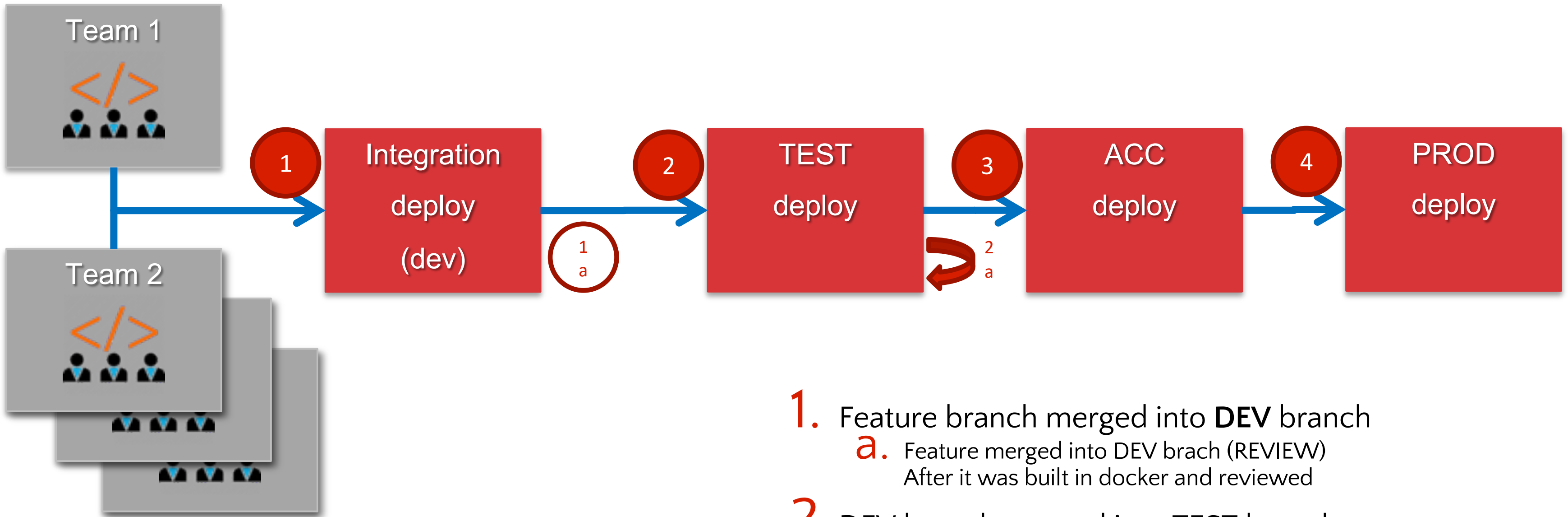


The deployment street



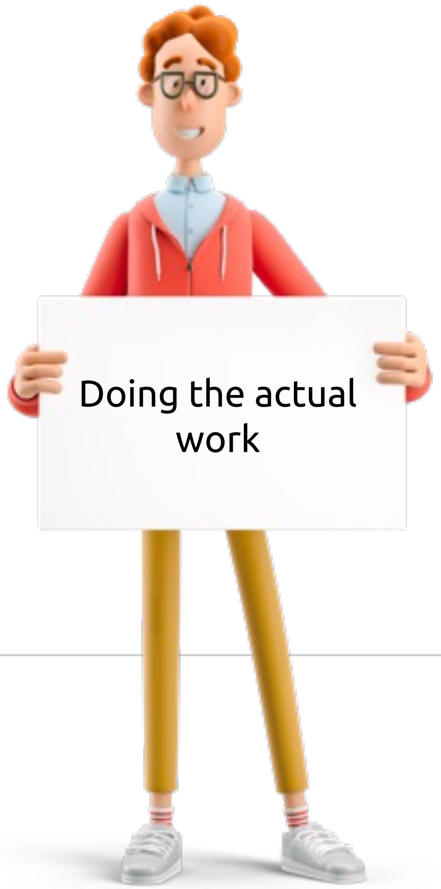
1. Feature branch merged into **DEV** branch
2. DEV branch merged into **TEST** branch
3. TEST branch merged into **ACC** branch
4. ACC branch merged into **MAIN** branch

The deployment street



1. Feature branch merged into **DEV** branch
   a. Feature merged into DEV brach (REVIEW)
      After it was built in docker and reviewed
2. DEV branch merged into **TEST** branch
   a. Full application export pushed into TEST and DEV branch
3. TEST branch merged into **ACC** branch
4. ACC branch merged into **MAIN** branch

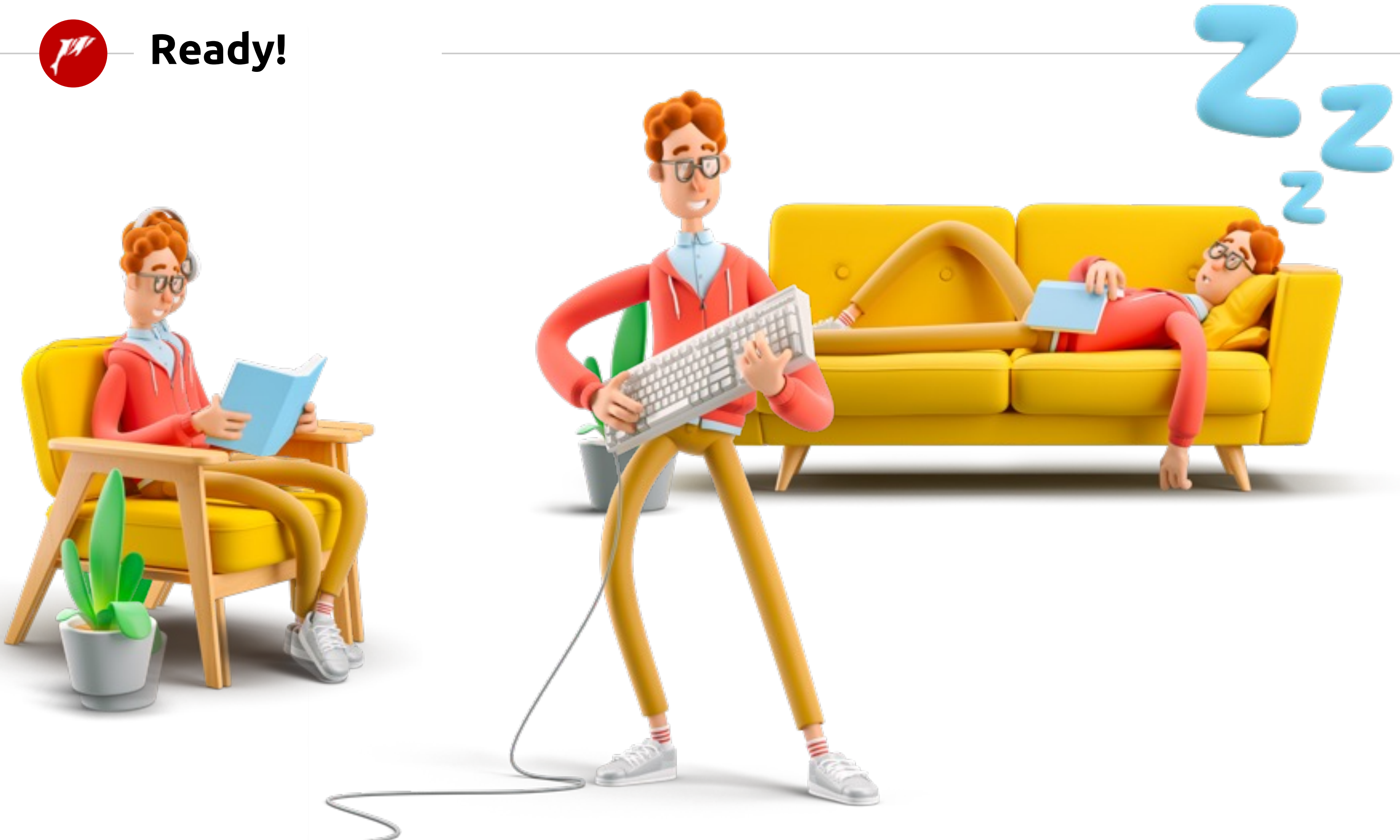
Workflow



Working Flow

- Create Feature Branch (FB)
- Lock page before start
- Save changes on the Feature Branch (DB / APEX)
- Push to Remote Feature Branch
- Build Docker instance to test integration
- Check work on Docker instance
- Create Merge Request
  - Is review step by colleague / QA
- Unlock page when merged into Development Branch

Ready!



Assumptions

Assumptions

- We don’t use the built-in static menu-list in our applications
  - Overcome by using build-options
- NEVER EVER touch a versioned file after it has been merged into DEV branch

- Btw, never ever touch a versioned file after it has been merged into DEV branch!!

Challenges

Challenges

- A shared development instance
  - Simultaneous development on single Database objects
- Corrections for versioned files should go in a new versioned file
- Commits and Merges preferably on the local filesystem push as late as possible
- Get to know your tools
  - IDE (PL/SQL Developer)
  - Gitkraken / Sourcetree / Fork / Tower / TortoiseGit
- **Get to know your process**
  - Git for teams

Recap

- All software is stored in the GIT repository
- A proven branching strategy
- Standard directory structure
- Deployment-tools are stored inside the repo
- Tools used
  - A ticket system (Jira/Wrike etc)
  - A GIT server (Bitbucket/Gitlab/Github etc)
  - A GIT Client (any will do)
  - IDEs for PL/SQL and CSS, images Javascript etc.
  - Flyway Community Edition

Thanks!

Any questions ?

You can find me at

- @rhjmartens
- rmartens@smart4solutions.nl