



Deutsche ORACLE -Anwendergruppe e.V.



# 10. Primavera Community Day München - 15.02.2017

## Project Archivation

# Archivierung von Projektplänen mit Werkzeugen der Software Entwicklung



# Innovative and Smart Way for Archiving, Versioning and Control of ORACLE-Primavera Project Data



GPI Consult – Gunther Pippèrr



3Pworx – Andreas Hock





## Better Archiving and Control of Oracle Primavera Project Plans – Customer Benefits (1)

- Archive, Backup and Restoration of Projects Plans – Use Cases (Examples)
  - 1. Restoring of logical failures in project plans
  - 2. If a user does mistakes - for example schedule a plan with the wrong project data without a new baseline before - you can restore the plan from the day before
  - 3. Excel Import or Global Change fails completely

Smart Restoration and Versioning of Project Plans

Avoid the restore of the full Primavera DB if only one plan was corrupted!



## Better Archiving and Control of Oracle Primavera Project Plans – Customer Benefits (2)

Better Fulfillment of your compliance and project standards and rules, Improvement of Change Management, etc.

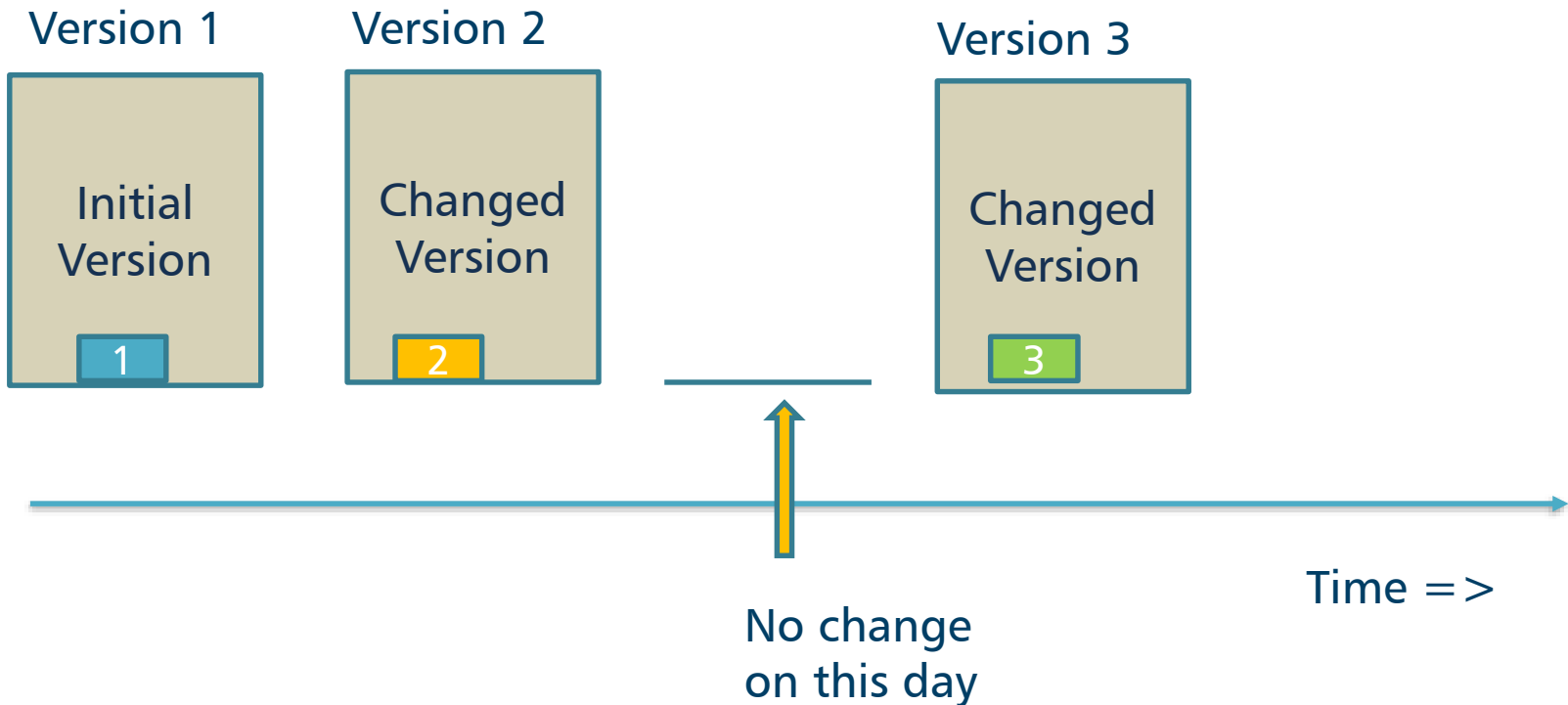
- Easy-to-Use solution to Archive of all changed Project Plans each day
  - Store all historical versions of a project plan
  - Only the changed plans will be saved every day

Clever Verification of Changes during the Life Time of the Project Plans



## The logical Overview

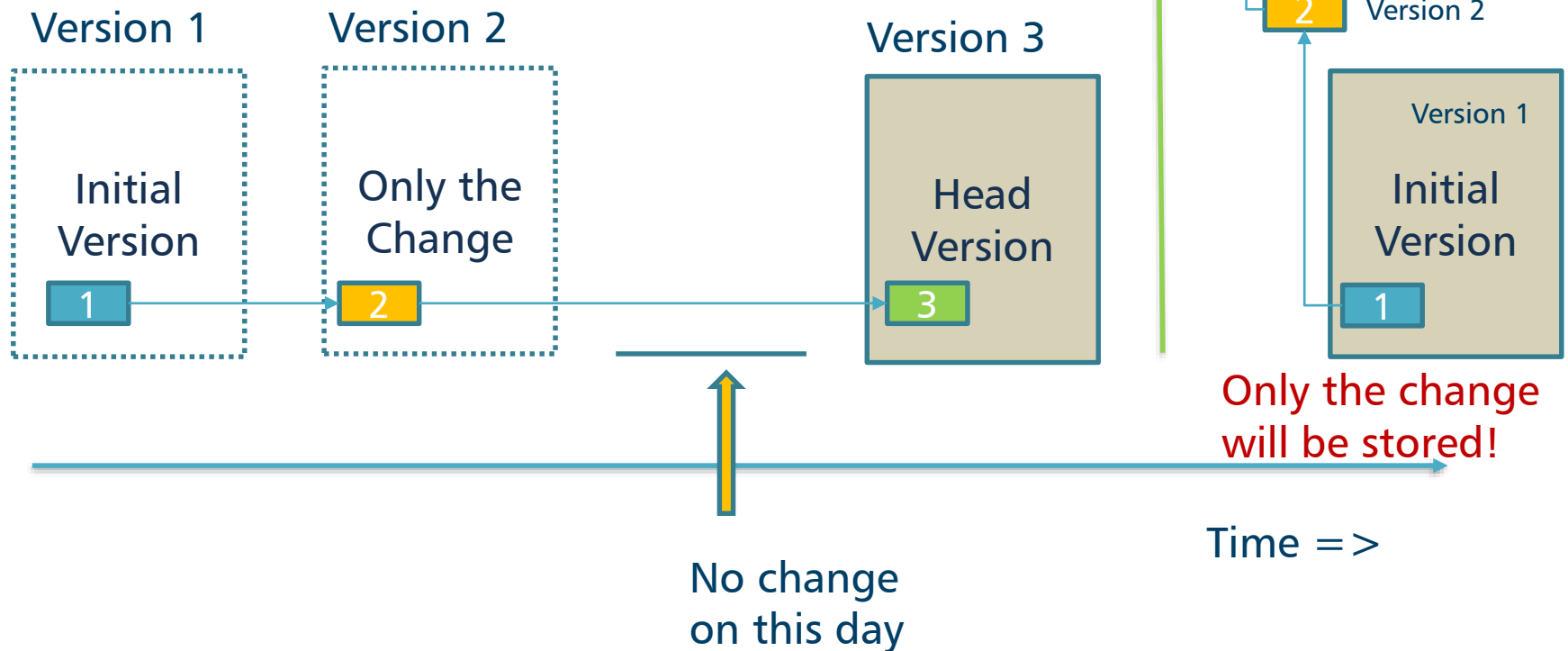
- Export every day the changed plans to a file system to the same file
- Only the last version is on disk
- Each change is in the GIT repository





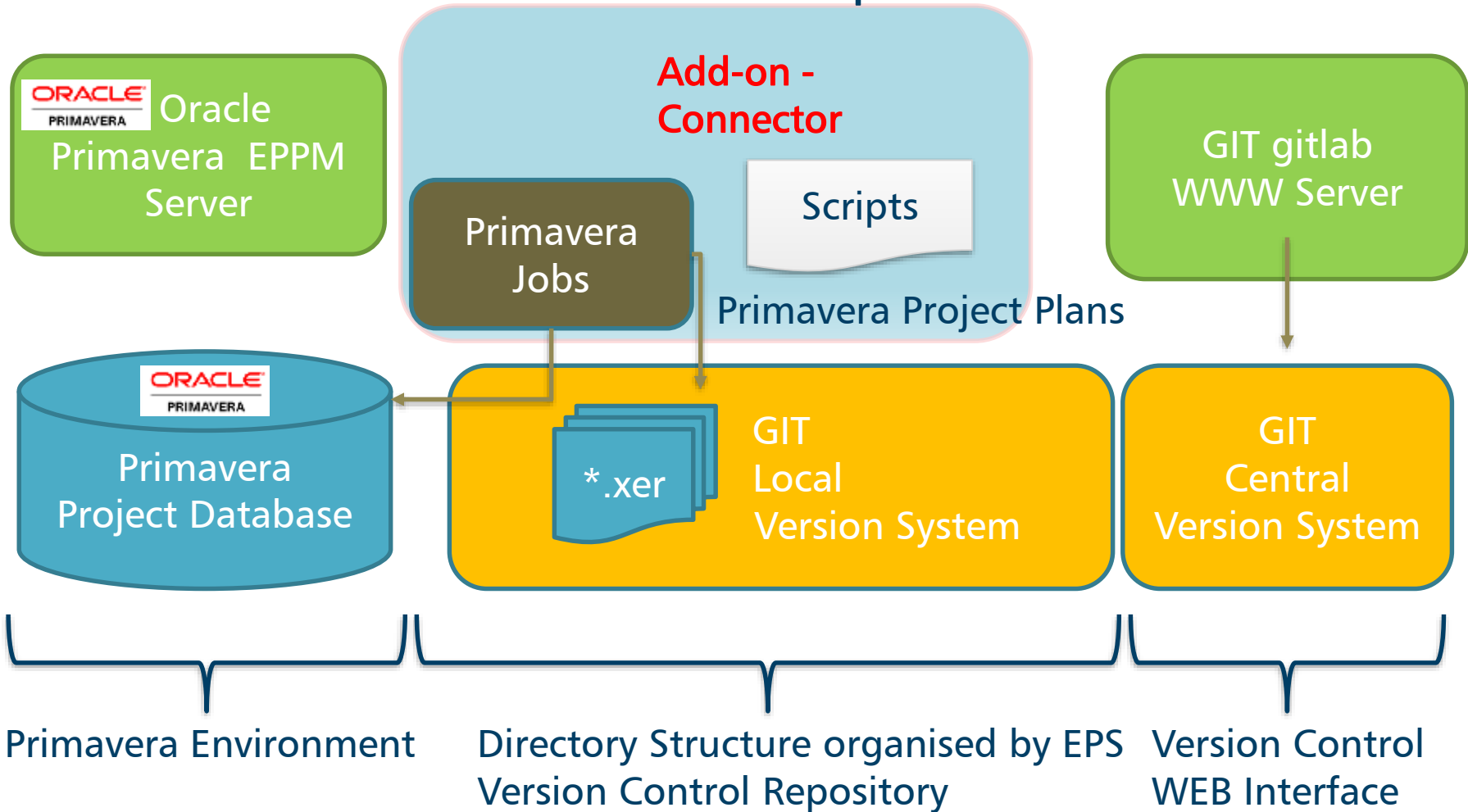
## The logical Overview with a Version Control System

- Export every day the changed plans to a file system to the same file
- Only the last version is on disk
- Each change is in the GIT repository





## The technical Architecture - Solution Footprint





## Use Case - Steps initial load of the Repository

- A script extracts the EPS structure and creates a directory structure with the same hierarchy
- A script exports from Primavera all projects plans as XER or XML files in the created directory structure
  - Project ID + Project name will be normalised to a valid Linux filename
- A script checks in all files to the GIT repository and commit the local GIT repository to the central repository
- The Primavera user can see now as first version all XER or XML files in the web interface





## Use Case - Steps daily load of the Repository

- A script reads the EPS structure and creates directories if missing
- Script export from Primavera of all **changed** projects from last run as XER/XML files in the created directory structure and overwrite the existing file
- A script checks now only all **changed** files to the GIT repository and commit the local GIT repository to the central repository
- The Primavera user can see now all changed versions for all XER files in the web interface



## Use Case - Restore Project Files

- Search over the GITLab Web Interface the last valid version of the plan
- Download the Primavera project plan from the GITLab web interface as \*.XER or \*.XML File
- Import the XER/XML File to a merge the data back into the project in the Primavera environment or create a new Project and
  - OR
- Check the diffs on the Files and fix the changes or issues



## Use Case - Example how to archive (1)

- Export project to your local repository

### Projects

Project ID	Project Name	Actual Nonlabor Units	Schedule % Complete	No
WORKSHOP	Workshop	0.00d	0%	
EX03	Primavera cottage	0.00d	0%	

### Export

#### File Name

What is the name of the .xer file?

File Name

R:\projectplan\workshop\EX03-primavera\_cottage\_roles.xer

> projectplan > workshop

Name	Date modified	
EX03-primavera_cottage_roles.xer	22.11.2015 17:00	1



## Use Case - Example how to archive (2)

- Add to local repository and commit the changes

```
gpipperr@jupiter MINGW64 /r/projectplan (master)
$ git add -A
warning: CRLF will be replaced by LF in workshop/EX03-primavera_cottage_roles.xer.
The file will have its original line endings in your working directory.

gpipperr@jupiter MINGW64 /r/projectplan (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   fix_gpi.patch
        new file:   workshop/EX03-primavera_cottage_roles.xer

gpipperr@jupiter MINGW64 /r/projectplan (master)
$ git commit -m "Plan V02 22.11.2015"
[master dc81fb6] Plan V02 22.11.2015
warning: CRLF will be replaced by LF in workshop/EX03-primavera_cottage_roles.xer.
The file will have its original line endings in your working directory.
2 files changed, 237 insertions(+)
 create mode 100644 fix_gpi.patch
 create mode 100644 workshop/EX03-primavera_cottage_roles.xer
```

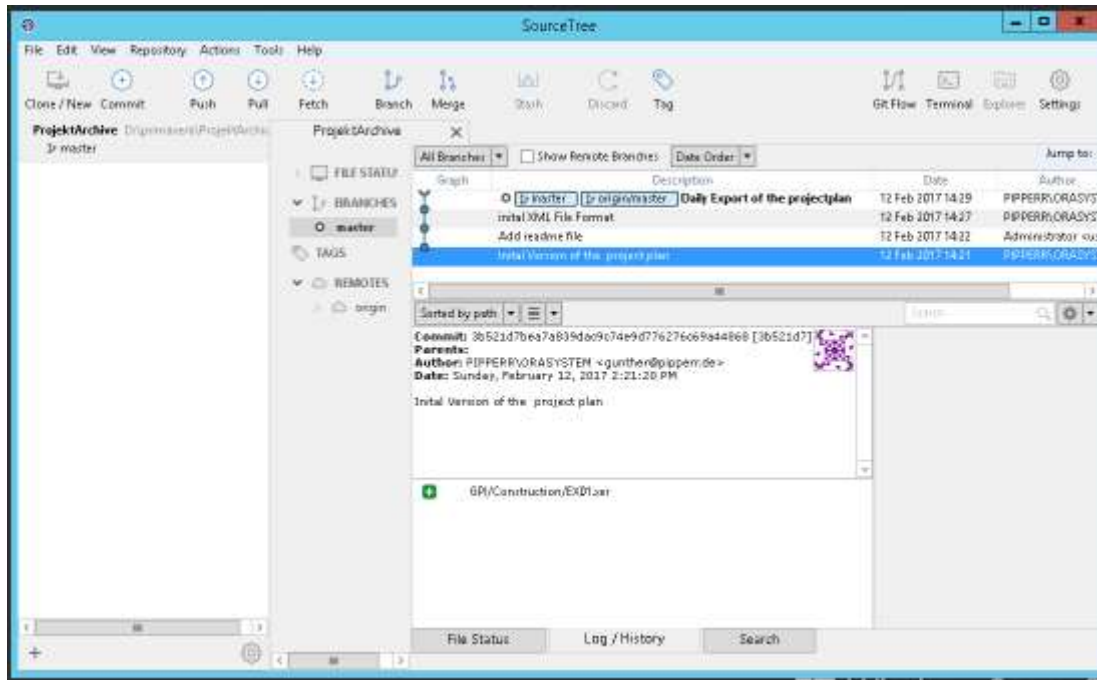
- Add/push the values to the remote repository

```
gpipperr@jupiter MINGW64 /r/projectplan (master)
$ git push gitlab_local2
```

- Change the plan and export/commit/push again to have two versions



## Use SourceTree as graphical Interface



<https://www.sourcetreeapp.com/>



## Use Case - Example how to archive (3)

- Show the different project version of the XER Files over the browser

Administrator / projectplan-repository

Search in this project



Commits 5

Compare

Branches 1

Tags 0

b18cb9dcf138...

projectplan-repository / plan / workshop



22 Nov, 2015

2 commits

V02 22.11.2015

Gunther Pippèr authored 3 minutes ago

V01 10.10.2015

Gunther Pippèr authored 5 minutes ago

Showing 1 changed file with 2 additions and 2 deletions

plan/workshop/EX03-primavera\_cottage\_roles.xer

plan/workshop/EX03-primavera\_cottage\_roles.xer

Use a local Editor to visualise the differences in the files:

dr_id	phys_complete_pct	rev_fdbk_flag	lock_plan_flag	auto_compute_act_flag	complet	Jr_id	phys_complete_pct	rev_fdbk_flag	lock_plan_flag	auto_compute
N	N	N	CP_Drtn TT_Mile DT_FixedDrtn	TK_NotStart A1000	Project Start	71	0			
N	N	N	CP_Drtn TT_FinMile DT_FixedDrtn	TK_NotStart A1020	Shell construction comp					
N	N	N	CP_Drtn TT_FinMile DT_FixedDrtn	TK_NotStart A1030	Project complete					
N	N	N	CP_Drtn TT_Task DT_FixedDrtn	TK_NotStart A1040	Clear Building Site	71				
N	N	N	CP_Drtn TT_Rsrc DT_FixedDrtn	TK_NotStart A1050	Excavate for foundation					



## Use Case - Show the difference in the web

- To check the diff of XER Files a plugin for the XER File type must be defined
  - Use XML Format – see this example:

Showing 1 changed file Inline Side-by-side

plan/workshop/EXP03-Primavera\_cottage.xml View file @b53d801

835	<Feedback />	835	<Feedback />
836	<FinishDate>2015-08-19T17:00:00</FinishDate>	836	<FinishDate>2015-08-19T17:00:00</FinishDate>
837	<GUID>{DDDB3413-CED8-4723-9AC5-0FB5B7D74D54}	837	<GUID>{DDDB3413-CED8-4723-9AC5-0FB5B7D74D54}
	</GUID>		</GUID>
838	- <Id>A1021</Id>	838	+ <Id>A1020</Id>
839	<IsNewFeedback>0</IsNewFeedback>	839	<IsNewFeedback>0</IsNewFeedback>
840	<LevelingPriority>Normal</LevelingPriority>	840	<LevelingPriority>Normal</LevelingPriority>
841	<Name>Shell construction completed - Topping-out ceremony</Name>	841	<Name>Shell construction completed - Topping-out ceremony</Name>



## IT Requirements (Example Configuration – Only for Illustration)

- Hardware Requirements
  - Linux VM Server for the GitLab Server with enough disk space for the remote repository
  - Windows Server for the scripting of the export with enough disk space the local repository
- Software Requirements
  - PPM – Primavera SDK and pure SQL on the Primavera Database
    - New User with read rights to the primavera tables + some admin tables
  - EPPM - P6 Integration API – Call via Java RMI Server Methods
    - Configuration of the Primavera Application Server necessary





## Customer Implementation Plan (Draft) - (1)

- Check which Software technologies can be used in the customer environment
  - Primavera Windows SDK or Java P6 Integration API
- Outline and prototyping of needed functional & technical design (e.g. via interviews, workshops), create customer solution footprint
- Calculate the necessary size for the repository from the existing plan volumes
- Deploying Server and Client hardware
  - Setup of the GIT repositories and the GITLab Webserver
  - Setup of the SDK or the Java Api



## Customer Implementation Plan (Draft) –(2)

- Implementing and testing the export scripts, interfaces or modules
- Initial Load and implement solution / first system set-up
- Add Use Cases e.g. Setup of the daily archiving process
- Documentation, prototyping in pilot project to adapt to customer requirements and needs, roll-out and hand over to the local IT, Project operation team, PMO for go-live



## Questions?



# ■ Contact Information:



## **GPI Consult – Gunther Pippèr**

Database software development

Schwanthalerstr. 82  
80336 Munich, Germany

Telephone: +49 (0)89 - 53 026 418

Mobile: +49 (0)171 – 80 65 113

E-Mail: [Gunther@pipperr.de](mailto:Gunther@pipperr.de)

<http://www.pipperr.de>



## **3Pworx GmbH – Andreas Hock**

Consulting for Process and Project Management

Landshuter Allee 8-10  
80637 Munich, Germany

Telephone: +49 (0)89 - 44 23 86 52

Mobile: +49 (0)163 - 8 20 60 17

E-Mail: [Andreas.Hock@3Pworx.com](mailto:Andreas.Hock@3Pworx.com)

<http://www.3Pworx.com>