COSMO CONSULT

Business-Software für Menschen

DevOps

Git

Prerequisite

Use Workshop-VM / Workshop-Computer
follow the instructions of the Boarding Pass

Visual Studio Code

https://code.visualstudio.com

Git

https://git-scm.com

Competence Team DevTech

Microsoft AL

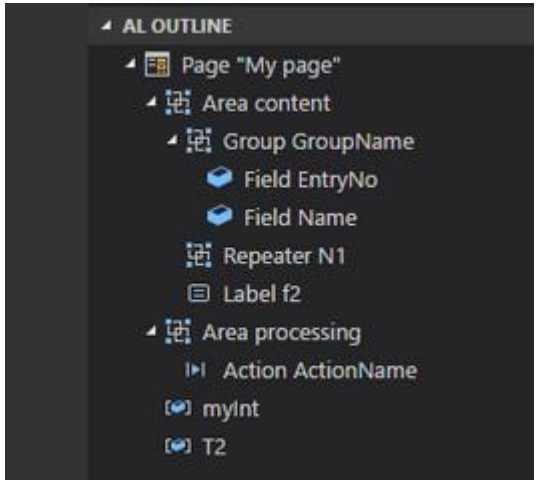
VS Code
Useful Extensions

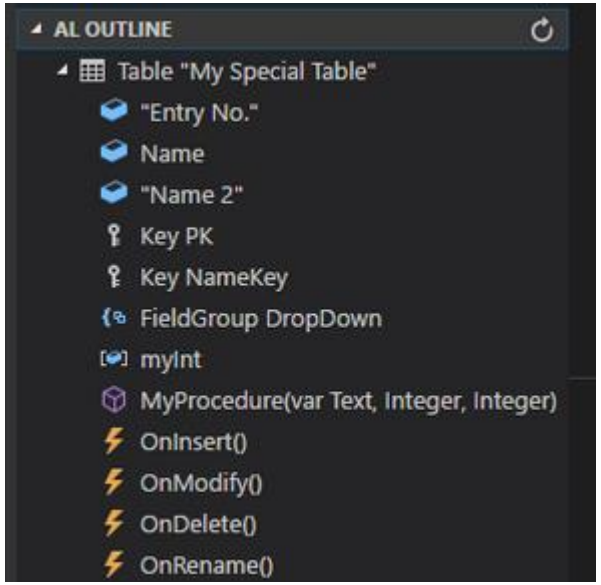
- AL Formatter
  - Indentation
  - Keyword case style
  - Sort variable definitions
  - Readability Guidelines - Spacing and newlines (experimental)

- AL Code Outline
  - Outlining – AL Object Structure
  - *"Create List Page"* from Table
  - *"Create Card Page"* from Table

- al-util
  - *"Rename & Move"* Objects
  - Templates e.g. "Readme.md"

```
Codeunit 70000000 MyCodeunit
{
    var
        Customer : Record Customer;
        Counter : Integer;
        InvalidCustomerErr : TextConst ENU='Invalid Customer
    trigger OnRun();
    begin
        if Customer.FindSet then
            repeat
                if not Customer.HasAddress then
                    Error(InvalidCustomerErr);
            until Customer.Next = 0;
    end;
}
```





AGENDA

1. Introduction Git
2. Git Commands
3. Git & VS Code
4. Hand On

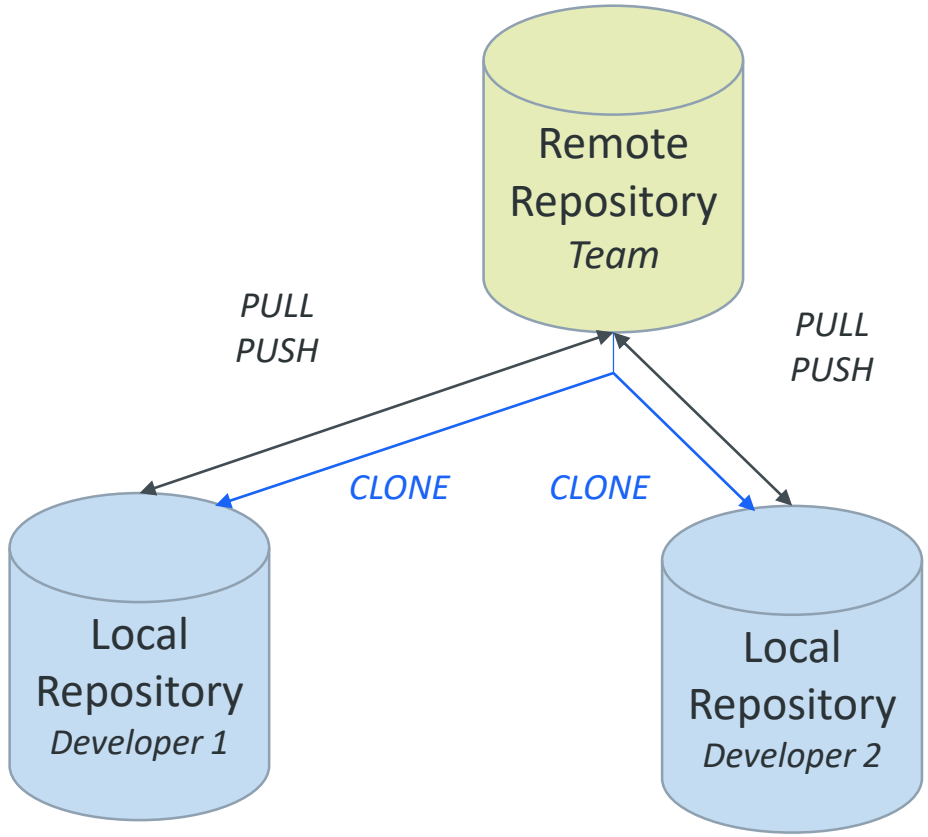
COSMO CONSULT

COSMO CONSULT
Business-Software für Menschen

Git
Introduction

Git ≠ *www.GitHub.com*

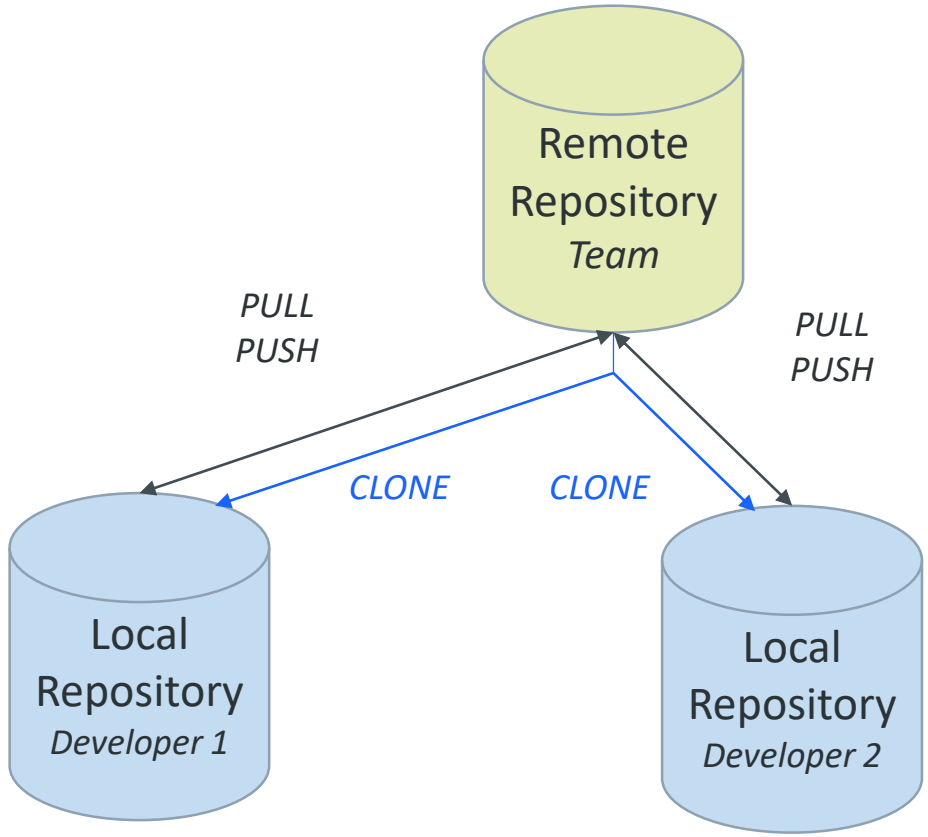
- What is Git?
  - Distributed version control system
  - Tracking changes on files
  - Used for source code management
- Strong support for non-linear development
  - Rapid branching and merging / a branch is only a reference to one commit
  - Help to coordinate work among multiple people
- Distributed development
  - Local copy of the full development history and changes
- Efficient handling of large projects
  - Very fast and scalable



Git & MS Dynamics

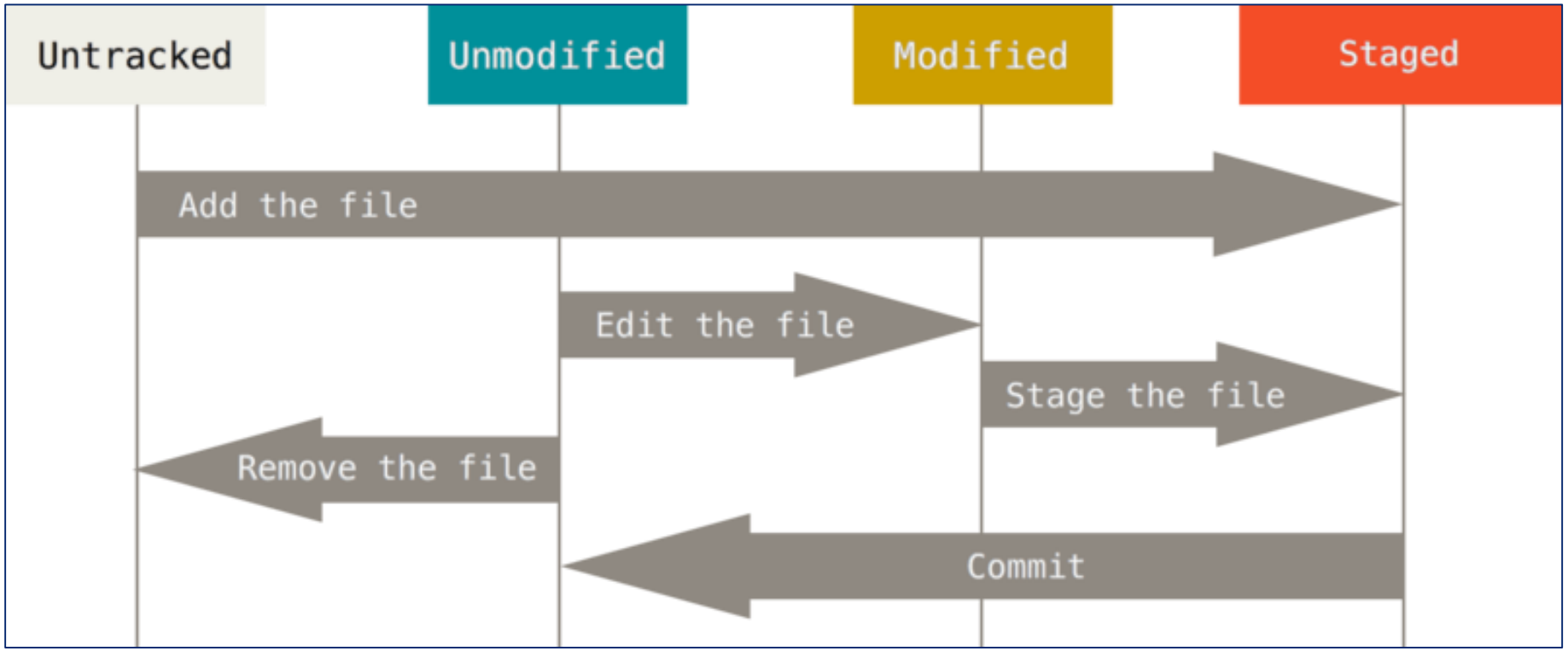
.../Getting-Started-Git-Basics

- Replace In-Db Version Control of Dynamics NAV
  - Tracking of Changes (What / Where / Who)
- File Types
  - Primary Track changes in TXT-Files!
  - Binaries supported **BUT** avoid FOBs
- Directories & Files
  - .git contains the version information metadata of the repository
  - .gitignore exclude files and directories from Git
- Branching
  - Development within same objects at the same time
  - Almost automatic merging of changes



Git – Recording Changes to the Repository

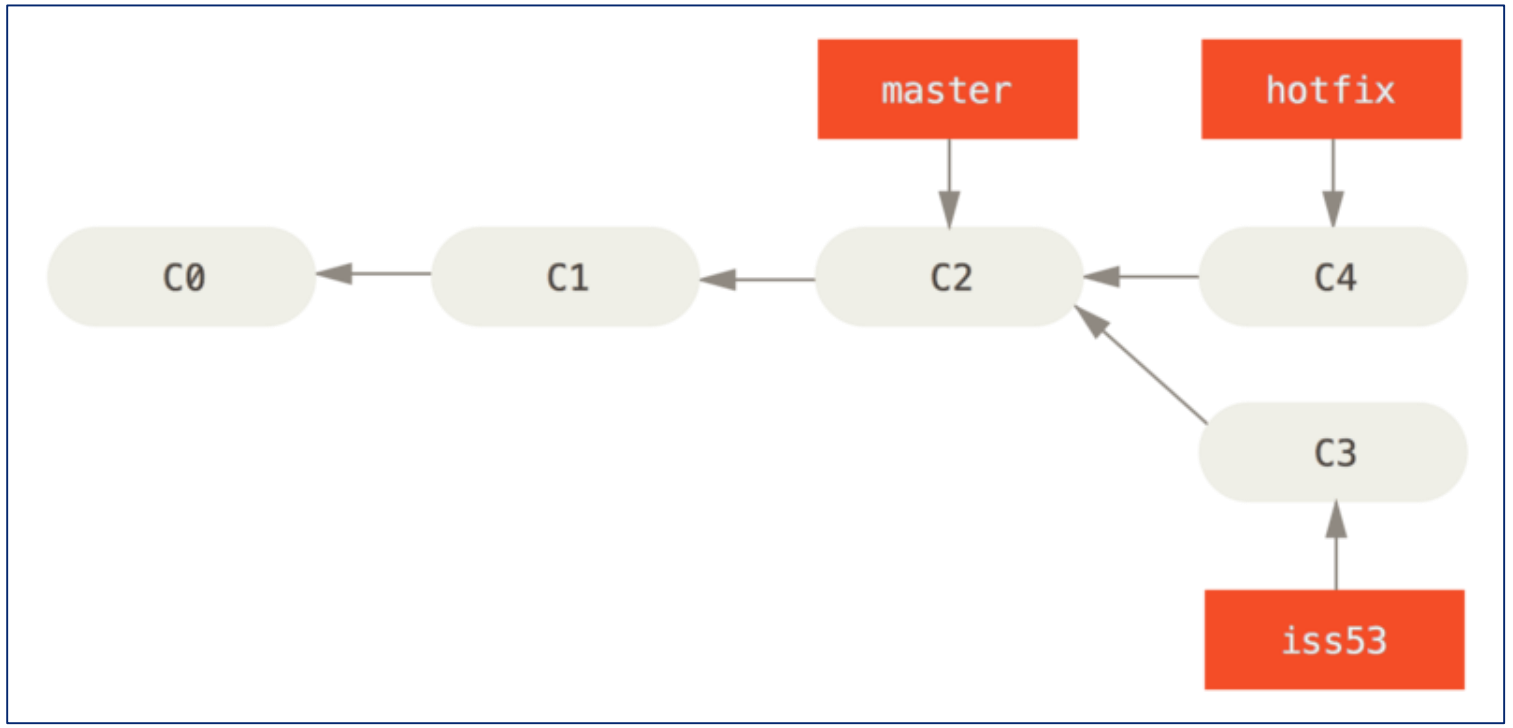
- Create Repository
- Add Files
  - *no empty* Directories
  - Status: ***Staged***
- Commit Changes
  - Unmodified files



Source: https://git-scm.com/book/en/v2/Git-Basics-Recording-Changes-to-the-Repository

Git – Branches

- Create a Branch
  - Track isolated changes
- Switch between Branches
  - Change the current View of Branch file(s)
  - Latest tracked Version of Branch file(s)
- Merge Branches
  - Combine the changes of 2 Branches since *Fork Commit*



Source: https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging

COSMO CONSULT
Business-Software für Menschen

Infrastructure
Git Commands

Git
Common Commands

.../common-git-commands.html

Cheat-Sheet

git init	Turns a directory into an empty Git repository `$ git init`
git add	Adds files in the to the staging area `$ git add <file or directory name>`
git commit	Record the changes made to the files to a local repository `$ git commit -m "Commit message in quotes"`
git status	This command returns the current state of the repository `$ git status`
git config	Configure the Git Settings e.g. „user.name“ and „user.mail“ `$ git config <setting> <command>`
git branch	To determine what branch the local repository is on, add a new branch, or delete a branch. `$ git branch <branch_name>`
git checkout	To start working in a different branch, use git checkout to switch branches. `$ git checkout <branch_name>`

Git
Common Commands / Remote Repository Commands

.../common-git-commands.html

Cheat-Sheet

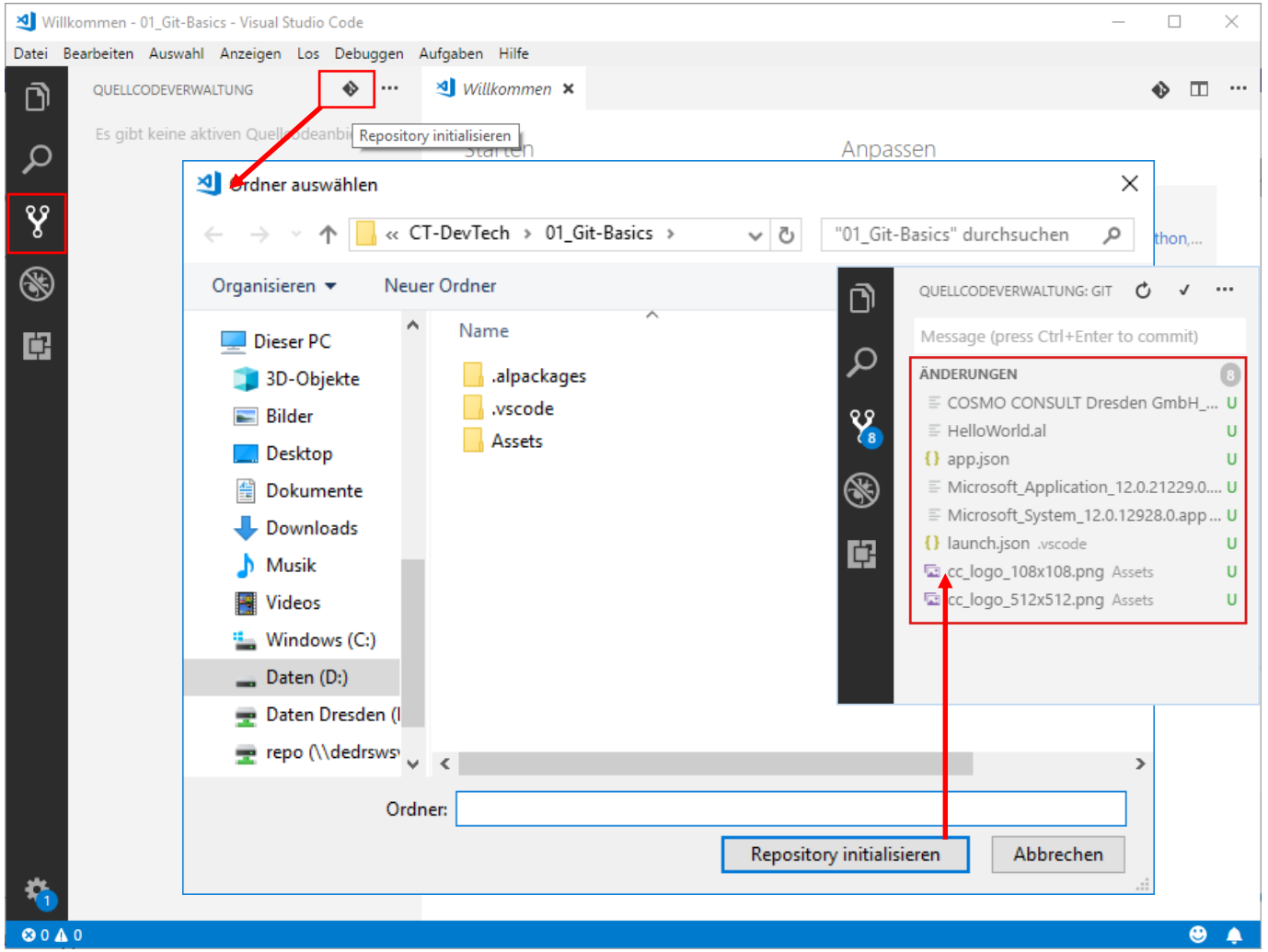
git merge	Integrate branches together. `$ git merge <branch_name>`
git log	Show the chronological commit history for a repository. `$ git log`
git remote	To connect a local repository with a remote repository. `$ git remote <command> <remote_name> <remote_URL>`
git clone	Create a local working copy of an existing remote repository `$ git clone <remote_URL>`
git pull	This pulls the changes from the remote repository to the local computer. `$ git pull <remote_URL/remote_name> <branch>`
git push	Sends local commits to the remote repository. `$ git push <remote_URL/remote_name> <branch>`

COSMO CONSULT
Business-Software für Menschen

Infrastructure
Git & VS Code

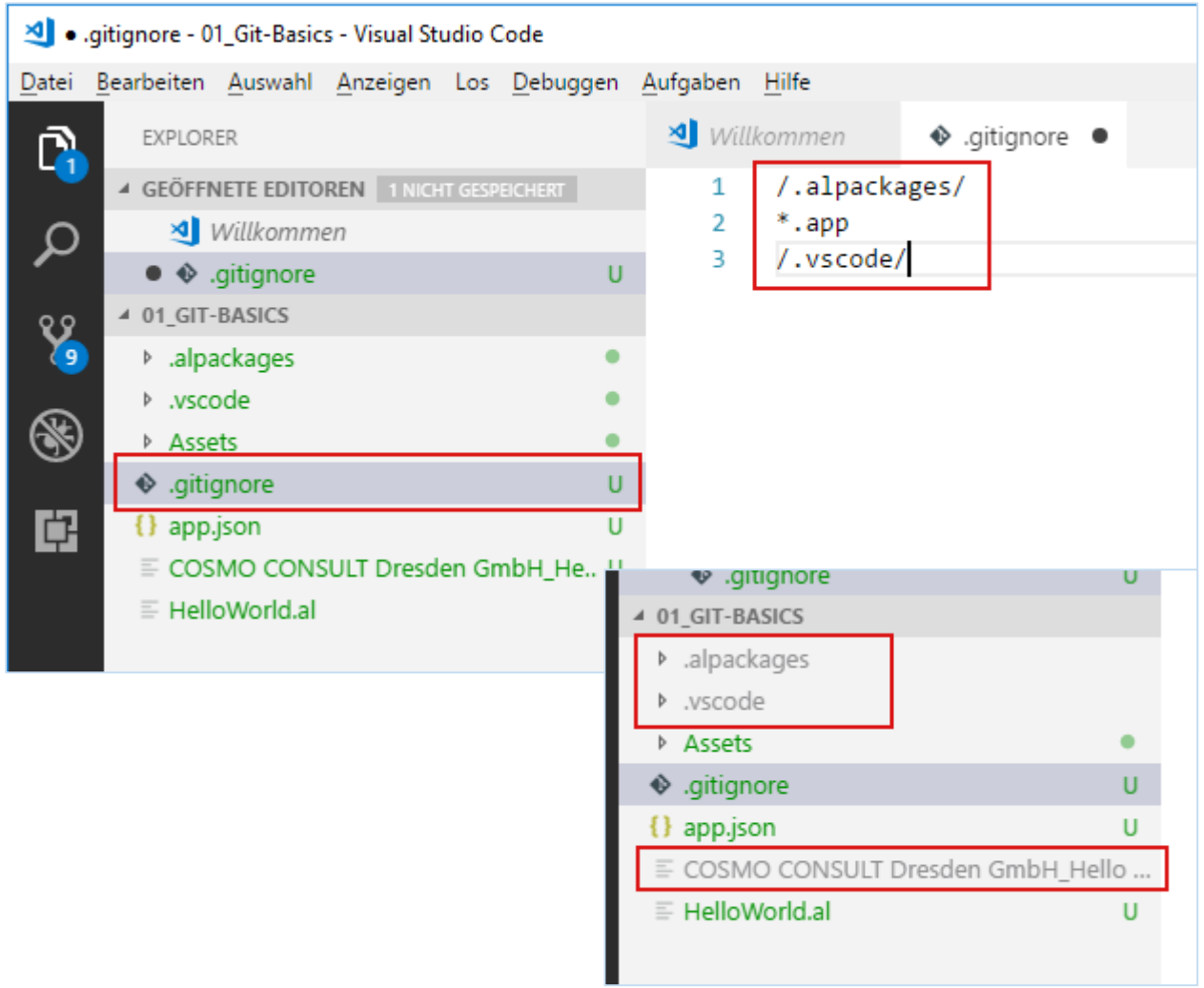
Git with VS Code

- Initialize the local Git repository
  - Select Folder
  - „Initialize Repository“
- Files will be recognized as „Untracked“



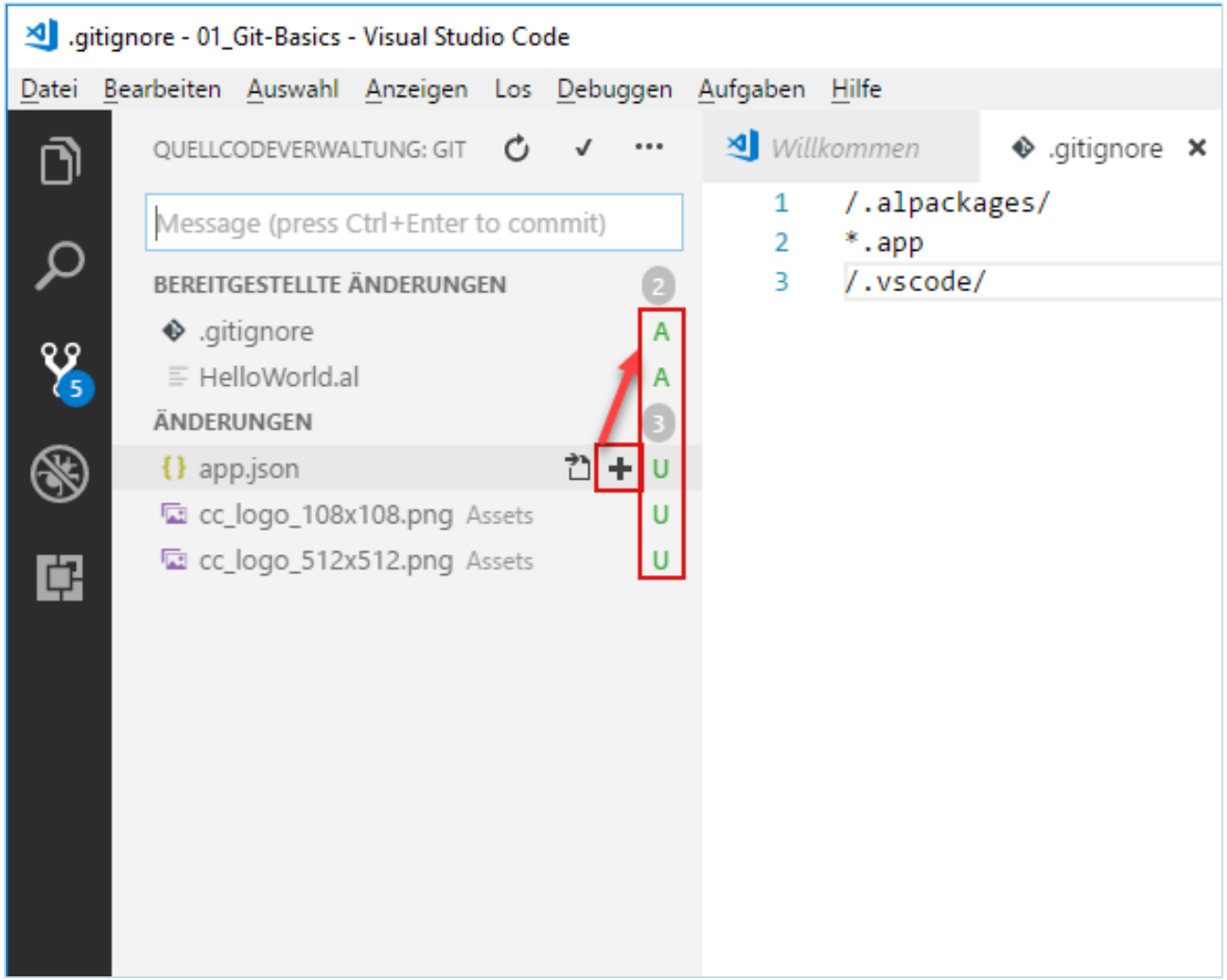
Git with VS Code

- Create the ".gitignore" File
  - Ignore Files & Folders from Repository
- Ignored Files
  - Symbols & Dependencies in ".alpackages"
    use Download from Server
  - Package File "*.app"
    created by compile
  - Workspace Settings in ".vscode"
    e.g. "launch.json"



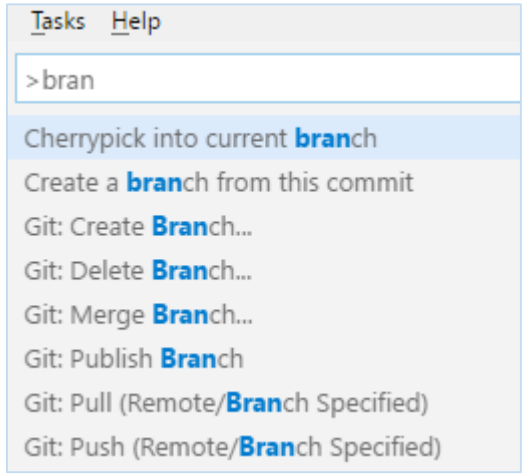
Git with VS Code

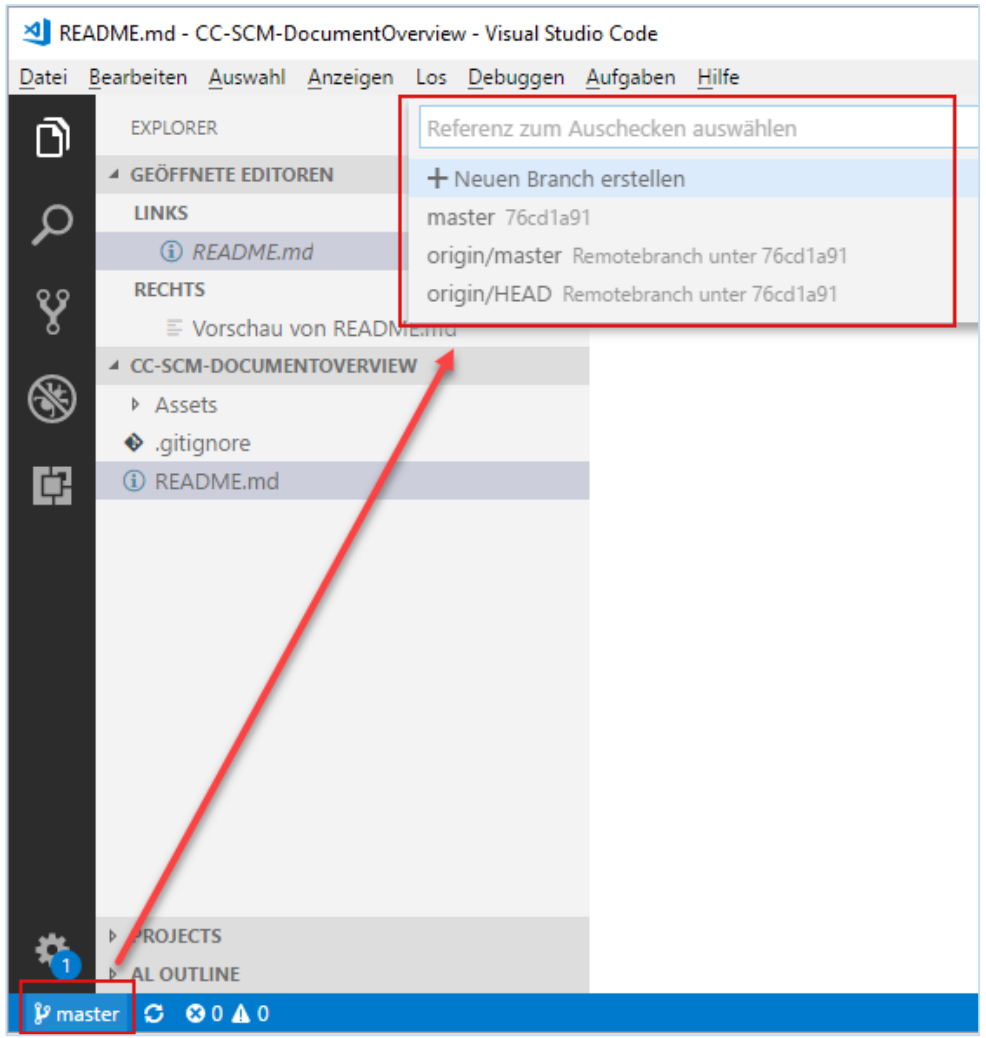
- Add Files
  - Mark as „Staged“
- Commit
  - Commit Message
  - Configure „user.name“ & „user.mail“



Git with VS Code

- Create / Switch Branch
  - Use Command Palett [Ctrl] + [P] or [F1]
    → enter *„BRANCH…“*
  - Use Status Bar
    → click on your current Branch e.g. *„master“*
- Pull / Push
  - Use Command Palett [Ctrl] + [P] or [F1]
    → enter *„PULL…“* / *„PUSH…“*





COSMO CONSULT

Business-Software für Menschen

Hands On

Git
Hands On

- Material
  - COPY the Folder “03-DevOps-Git**Starter**” to a **local folder** e.g. **“C:\Workshop\MyStarter\”**!
    (not Git observed)
  - Project Skeleton „..\03-DevOps-Git**Starter**\COSMO-CONSULT-**1**\”
  - Project Skeleton „..\03-DevOps-Git**Starter**\COSMO-CONSULT-**2**\”
  - Project Skeleton „..\03-DevOps-Git**Starter**\COSMO-CONSULT-**3**\”

- Prerequisites
  - Connect to your Workshop VM → use Instructions on your Workshop Boarding Pass

- Hand On - Tasks
  - Achieve common development related tasks with the Git command line
  - Achieve common development related tasks with the VS Code and the Git Integration
  - Achieve common development related tasks with the a 3rd Party Tool

Git – Hands On
Use the Git command line

- Open folder „…*03-DevOps-Git**Starter***01_Git-Basics**" in Git Bash / Command Line
  - Open the folder "**01_Git-Basics**" also in VS Code
- Init the Repository

```
$ git init .
```

- Add File(s)

```
$ git add *
```

- Commit File
  - Check your Git-Config (Username / User-Email) ☺

```
$ git commit -m "initial commit"
```

```
$ git config user.email "my@emailaddress.com"
$ git config user.name "Jon Doe"
```

- Create Branch „feature"

```
$ git branch "feature"
$ git checkout "feature"
```

- Add or Change File(s) & stage File (git add)

```
$ git add *
```

- Commit File

```
$ git commit -m "comment your changes"
```

- Switch between branches „master" and „feature"
  - see the difference

```
$ git checkout "master"
```

- Merge branch „feature" into „master"

```
$ git checkout "master"
$ git merge "feature"
```

Git – Hands On
Use VS Code git integration

- Open folder „…*03-DevOps-Git**Starter*\02_Git-Basics" in VS Code
- Init the Repository
- Create a ".gitignore" file to ignore Symbols (".alpackages") and the ".vscode" folder and the ".app" file
- Add the other project File(s)
- Commit File
  - Check your Git-Config (Username / User-Email) ☺

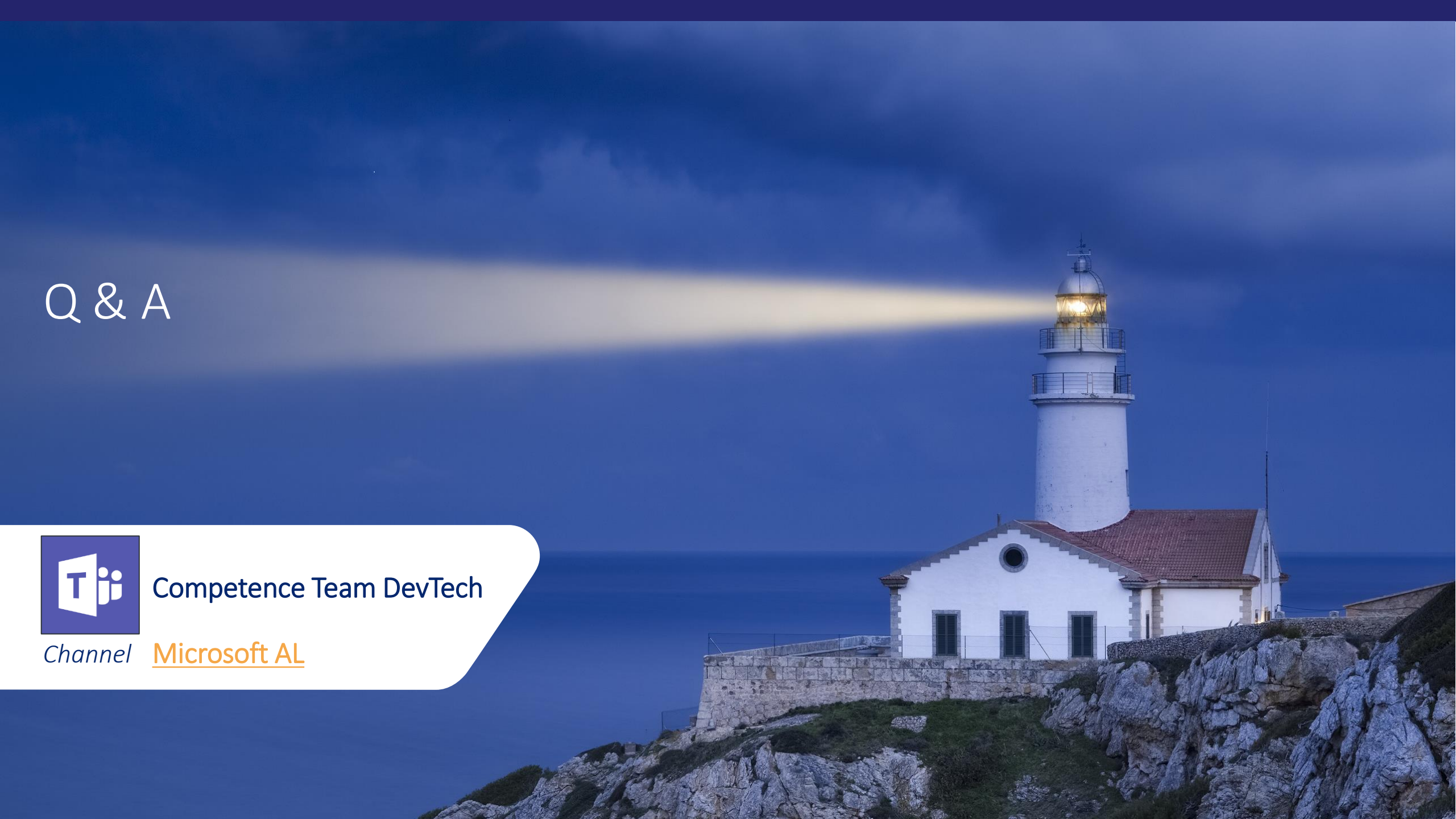
- Create Branch „feature"
- Add or Change File(s) & stage File (git add)
- Commit File
- Switch between branches „master" and „feature"
  - see the difference
- Merge branch „feature" into „master"

Git – Hands On
Use a 3rd Party Tool (e.g. Smart GIT form syntevo) for your Git workflow

- Open the folder „…*03-DevOps-Git**Starter***03_Git-Basics**" in VS Code
- Add and Init the Repository („…*03-DevOps-Git**Starter***03_Git-Basics**")
- Ignore Symbols (".alpackages") and the ".vscode" folder and the ".app" file
- Add the other project File(s)
- Commit File
  - Check your Git-Config (Username / User-Email) ☺

- Create Branch „feature"
- Add or Change File(s) & stage File (git add)
- Commit File
- Switch between branches „master" and „feature"
  - see the difference
- Merge branch „feature" into „master"

Q & A



Competence Team DevTech

Channel Microsoft AL