

A DYNAMICSCON PRESENTATION

POWERED BY  DUG

# DYNAMICSCON VIRTUAL

MARCH 2023

C:\Users\klemen.novak\OneDrive - Docentric  
d.o.o\Events\2023-03 DynamicsCon Virtual

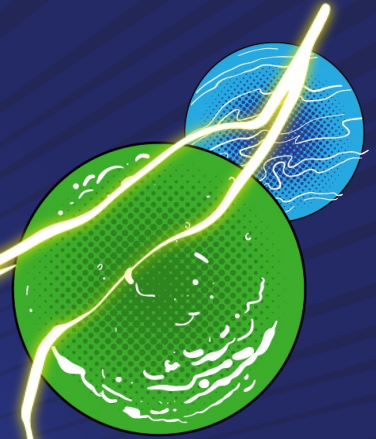
FINANCE &  
OPERATIONS

[DYNAMICSCON.COM](https://DYNAMICSCON.COM)



# Multiple Projects on D365FO OneBox using Git

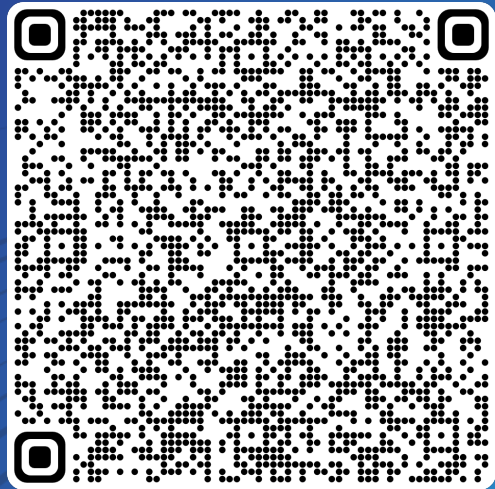
KLEMEN NOVAK



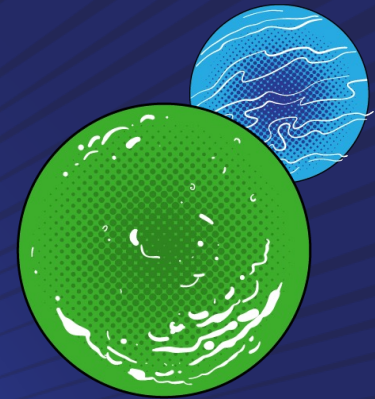
# KLEMEN NOVAK

## CTO @ DOCENTRIC

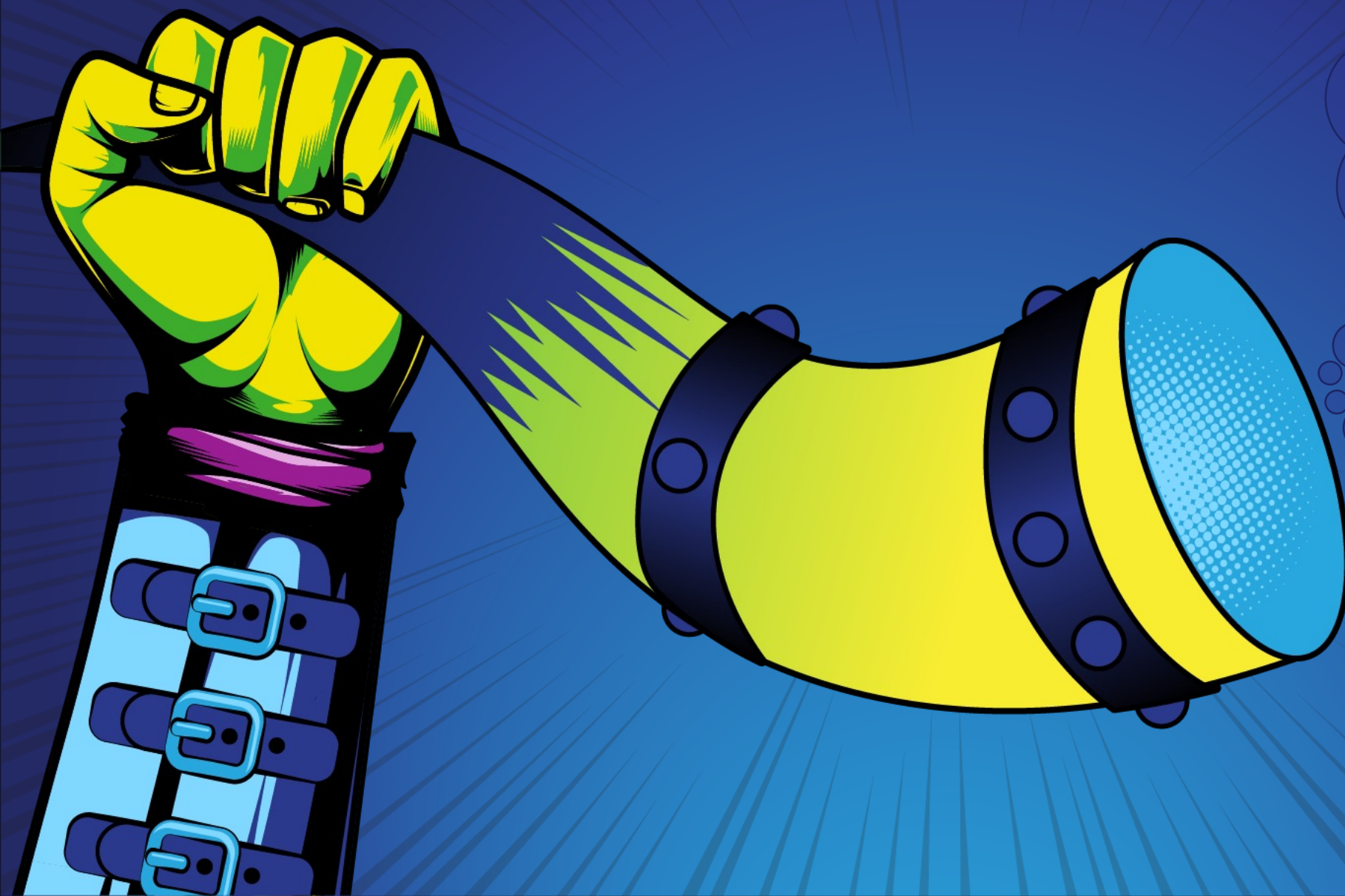
Sometimes I still get my hands dirty and write code ;)



# TODAY'S AGENDA



- Intro
- Getting started with Git
- Branching strategy
- Pull Requests
- Demo



# What is Git?

Git repositories:

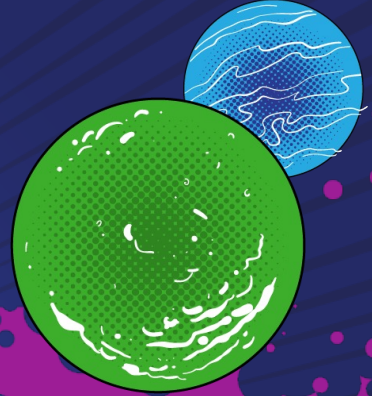
- **Azure DevOps**
- **GitHub**
- **GitLab**

# TFS vs Git Commits



Git commit is a hash:  
f52435ce2ffeb7d6b8f1573ca8a6bba9d0697520

# Git: Distributed version control



Repository

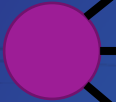
Time / Feature / Project →

Sara



c3867e5f

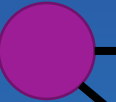
Daniel



d2620bd2



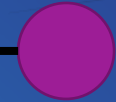
774fe382



99794ce0



f51ab6a1



b4e7cc4b

Jessica



5a011a31



670db80d

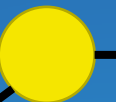


b17163e0

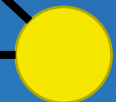
John



6af8580c



51c4f71c



a625121e

Sam



12a24a4f



# The comparison: TFS vs Git



## TFS

- Push based
- Branches last nearly forever and are hard to work with
- Branches are folders based
- Monolithic
- Supports multiple destination folders

## Git

- Modern & Agile
- Working offline (speed)
- Pull-based (Code reviews in one place)
- Branches are short-lived and are easy to work with
- More tools, possible build and deploy automations
- Requires reallocating projects in a single location

**AWESOME RIGHT ;)**

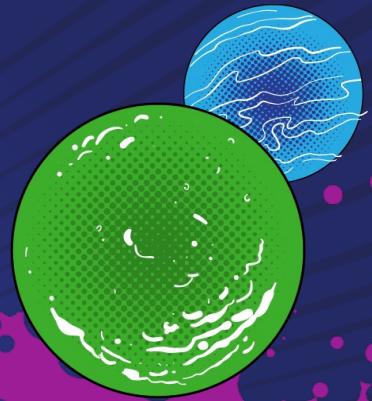
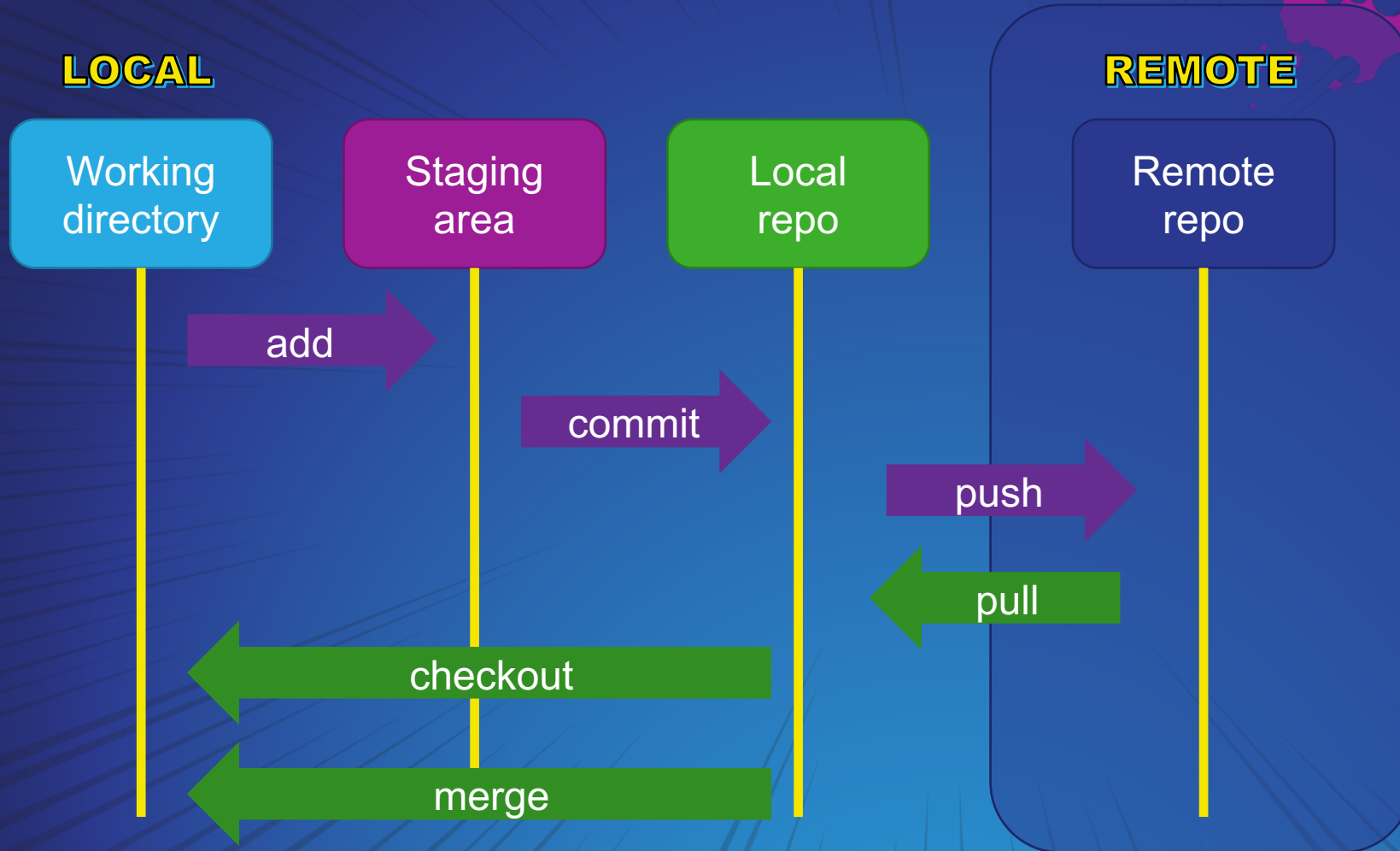


# Git Basics



Tell me  
more ;)

# Git Basics



# Git Basic Commands

## SETUP

Configuring user information used across all local repositories

```
git config --global user.name "Jonh Doe"  
git config --global user.email "email"
```

## INITIALIZE GIT REPOSITORY

Initializing local repository (inside a folder)

```
git init
```

Cloning remote repository to local repository (by default, the folder will be named the same as the remote repo)

```
git clone [url]
```

## STAGE & COMMIT

Check the current state of your branch.

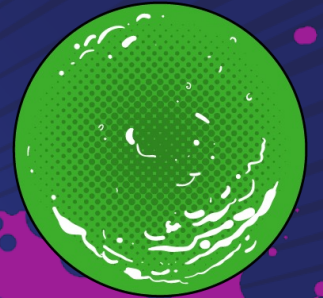
```
git status
```

Add all file changes for next commit.

```
git add -A
```

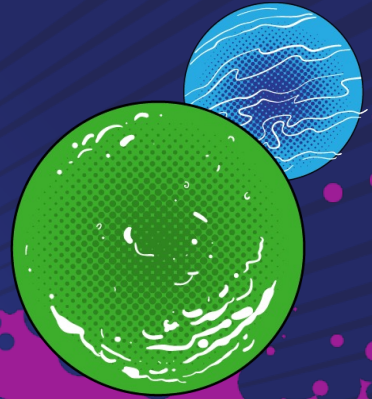
Commit your staged content as a new commit snapshot locally.

```
git commit -m "[descriptive message]"
```



Git Cheat Sheet: <https://education.github.com/git-cheat-sheet-education.pdf>

# Git Basic Commands



## BRANCH & MERGE

List all branches

```
git branch
```

Create a new branch at the current commit

```
git branch [branch name]
```

Switch to another branch and check it out in your working directory

```
git checkout [branch name]
```

Merge the specified branch's history into the current one

```
git merge [branch name]
```

Show all commits in the current branch's history

```
git log
```

## SHARE & UPDATE

Add a git URL as an alias.

```
git remote add [alias] [url]
```

Fetch down all the branches from that Git remote (get them listed in the local repository).

```
git fetch [alias]
```

Transmit local branch commits to the remote repository branch (send it to server).

```
git push [alias] [branch]
```

Fetch and merge any commits from the tracking remote branch (get all the changes).

```
git pull
```

Git Cheat Sheet: <https://education.github.com/git-cheat-sheet-education.pdf>

# Git Tools



**TortoiseGit**

There are other alternatives.

# Getting started with Git

---



1. Create Git Repository
2. New Branch
3. Add .gitignore
4. Add .gitattributes
5. Add the code (models and projects)
6. Commit and push to remote

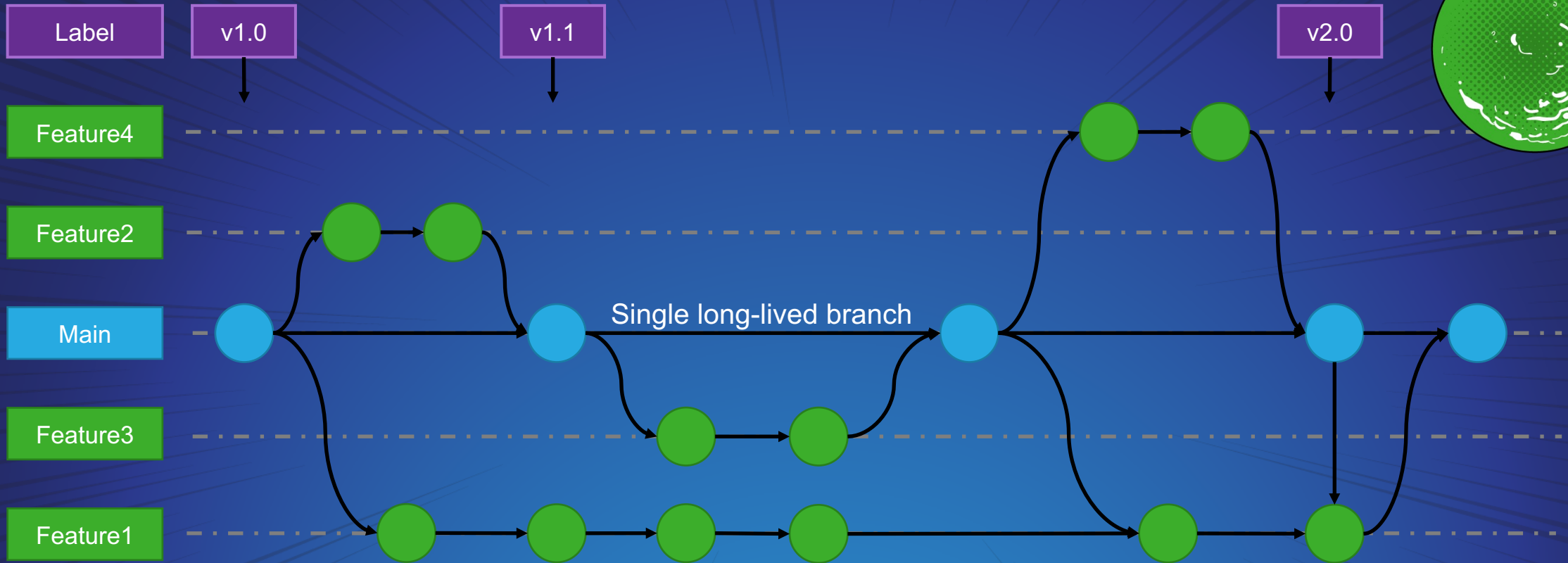
# Git Branches



**What are  
Branches?**

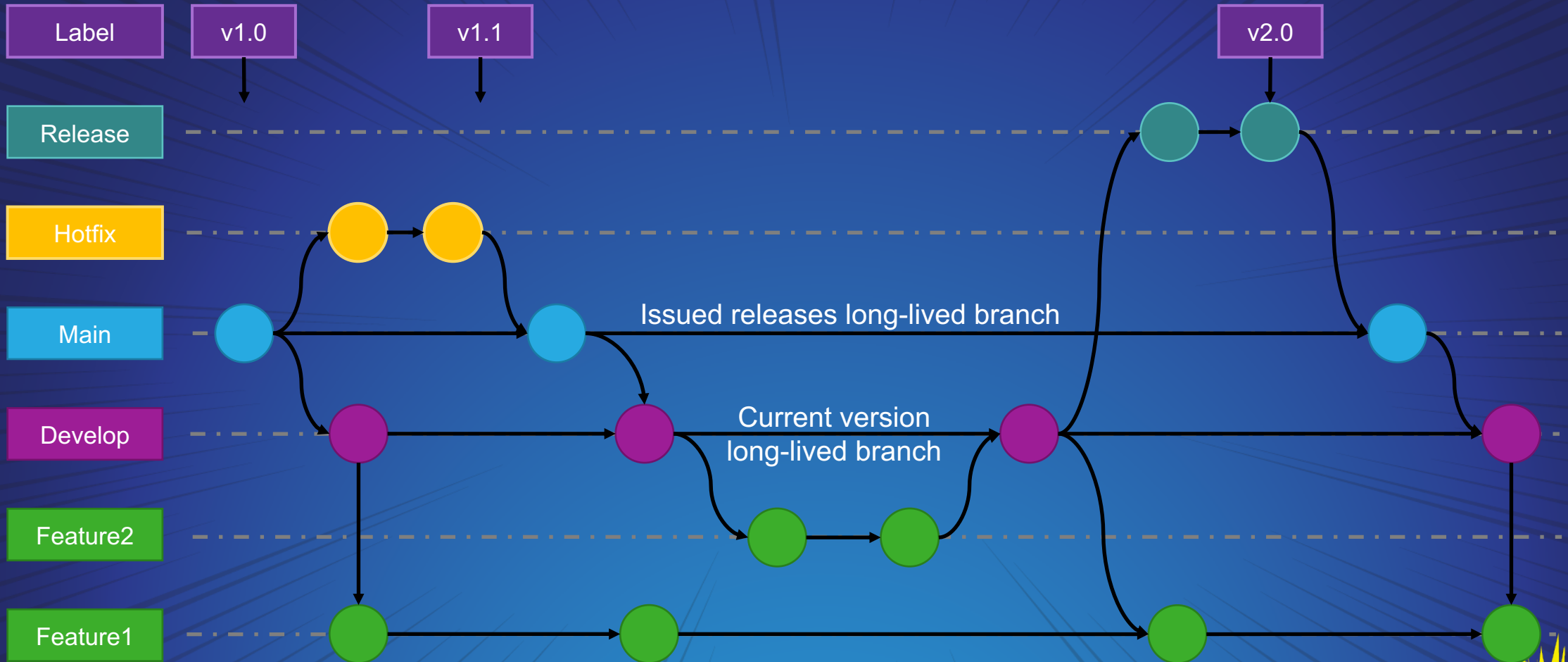
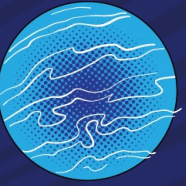
How to work  
with  
Branches?

# Branching with GitHub flow





# Branching with GitFlow



# Branch Workflow Comparison



## Github flow

- Simple to use
- A single long-lasting branch (main)
- Intended for each day deployment

## GitFlow

- More complex to use
- Two long-lasting branches (main and develop)
- Easier to create hotfixes, new releases
- Intended for product development

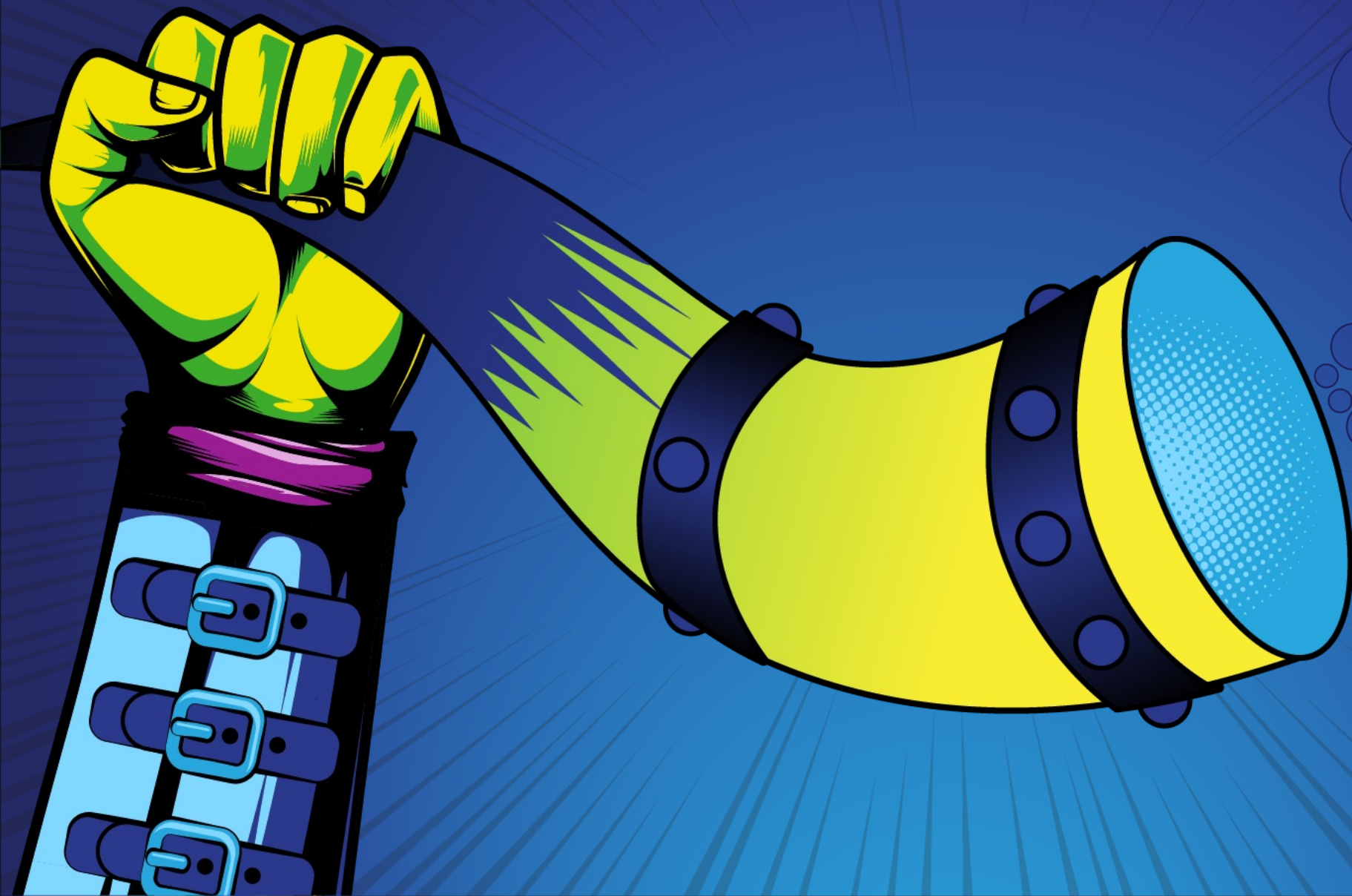
**Both options are excellent!**

The development process is constantly improving and we can change the branch workflow later without hassle.

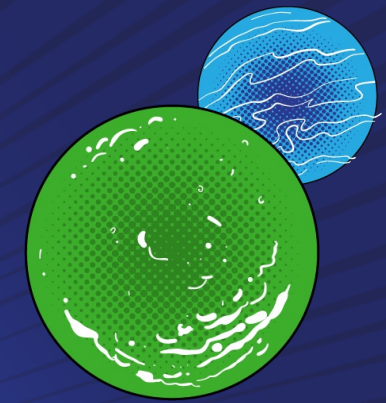
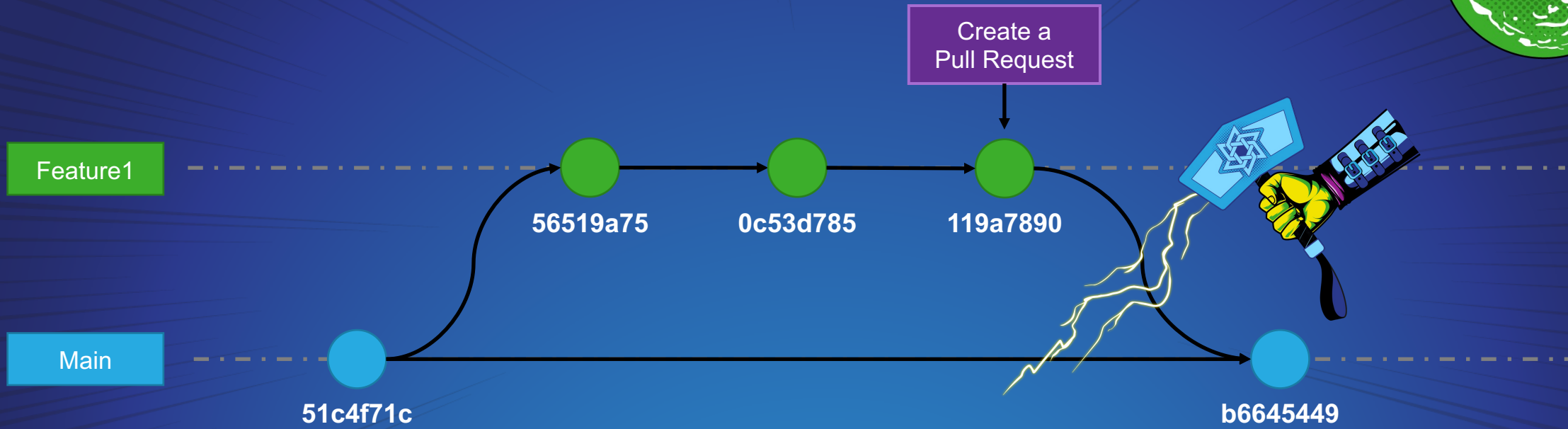
# Pull Requests

What are  
Pull  
Requests?

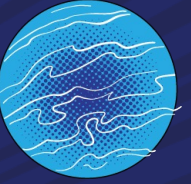
How are they  
related with  
branches?



# Pull Requests



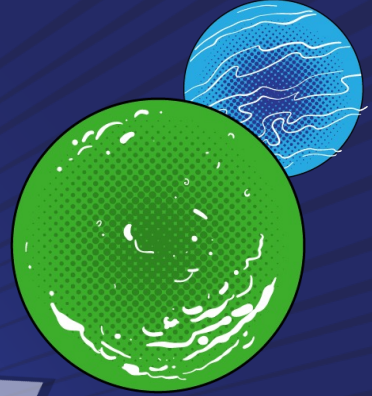
# Multiple Projects on D365FO OneBox



**Finally!  
I want a  
demo ;)**



# FINAL THOUGHTS



**That's  
all folks!**

**Thank you!**

## **Used resources**

- [Git template repository](#)
- [Gitflow for Visual Studio 2019](#)
- [Pull Request for Visual Studio 2019](#)