

Git Merging – WPL



**DE HOGESCHOOL
MET HET NETWERK**

Elfde-Liniestraat 24, 3500 Hasselt, www.pxl.be

Git Merging

- Wanneer je samenwerkt met collega's aan een project, dan deel je een repository.
 - Aanpassingen worden pas gesynchroniseerd wanneer een lid zijn/haar project pusht.
 - Eens verschillende leden aanpassingen maken aan dezelfde code, dan ontstaan er **conflicten**.
 - Conflicten worden opgelost in een merge.



Overzicht

- Voorbereiding
- Merge
- Branch
- Checkout
- Remote Branches
- Merging Branches

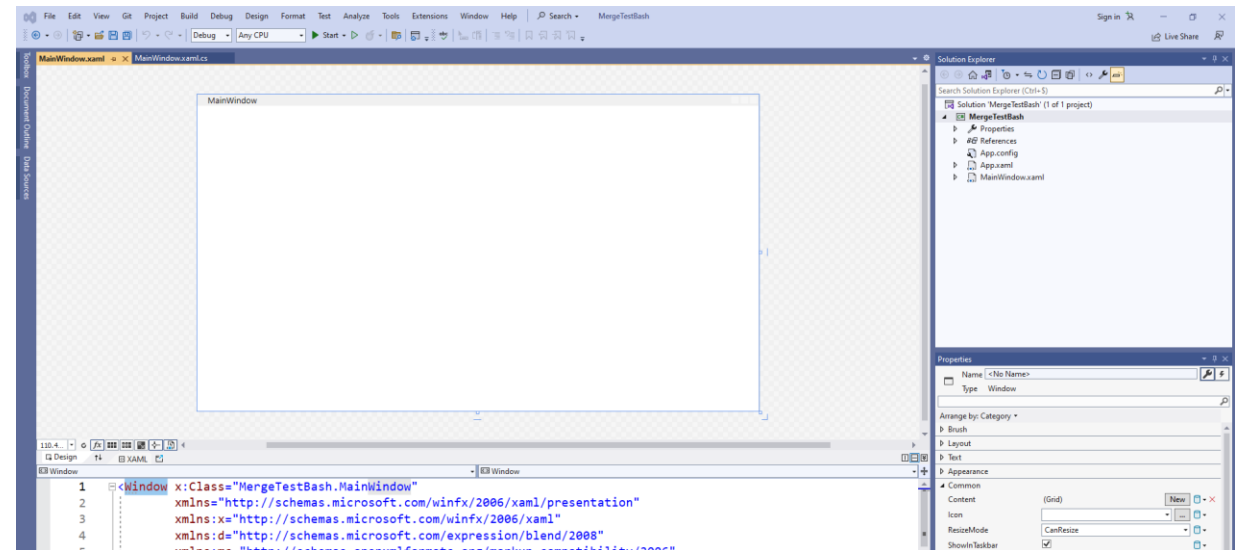
Vorbereiding

Samenwerken met een collega



Vorbereiding

- Om een **merge conflict** te veroorzaken, gaan we **samenwerken** met een collega.
 - Hiervoor hebben we een gedeelde **remote repo** nodig
- Volg het stappenplan van de slides om het project aan te maken:
 - Visual Studio
 - Git Bash



Vorbereiding

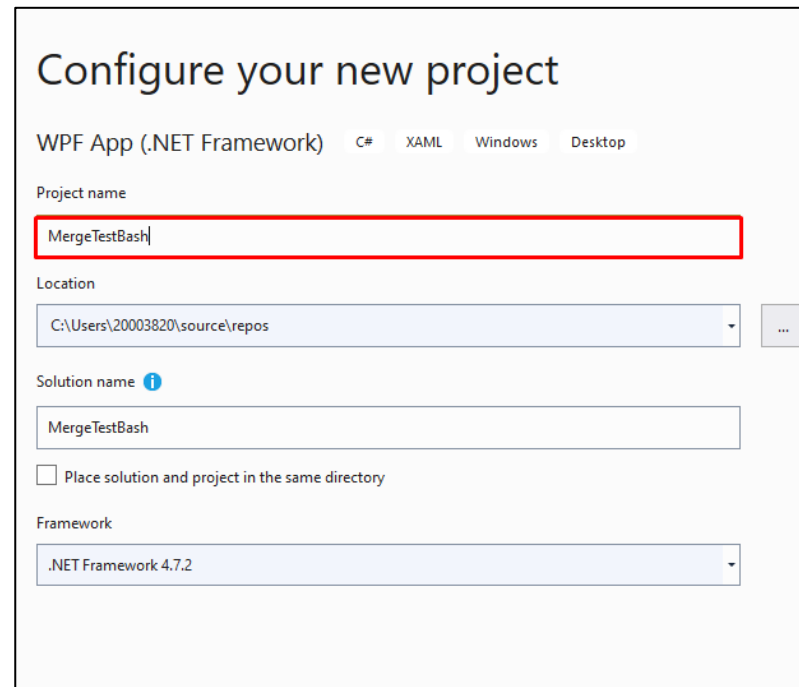
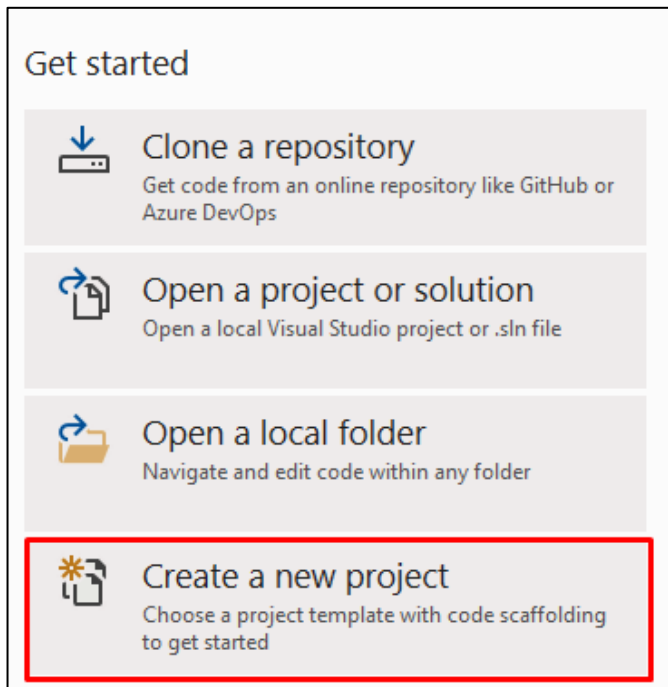
- **Remote Repo** kan je op twee manieren maken:
 - Of: Je maakt een lokale repo aan die je pusht naar Bitbucket/Github
 - Of: Je maakt een nieuwe remote repo en die clone je voordat je start met programmeren
- In het stappenplan maken we eerst een lokale repo aan
- Daarna pushen we de repo naar een remote repo



Voorbereiding



- Visual Studio:
 - Maak een WPF project aan



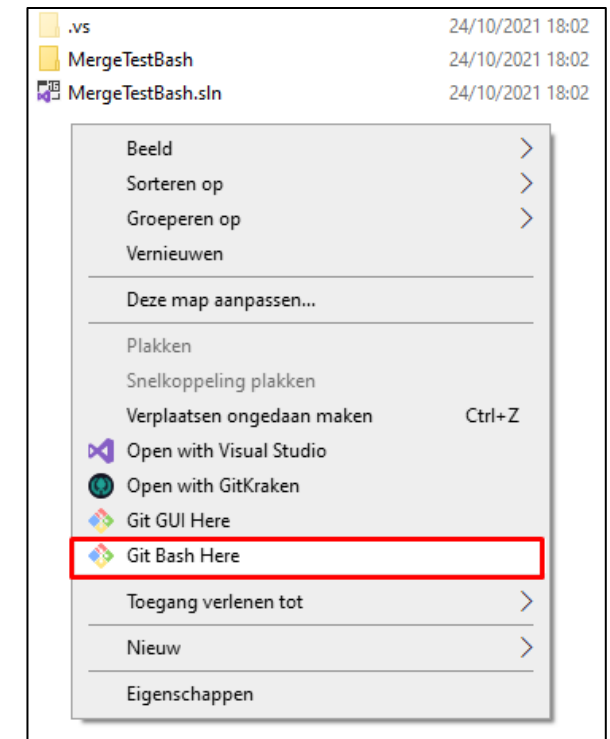
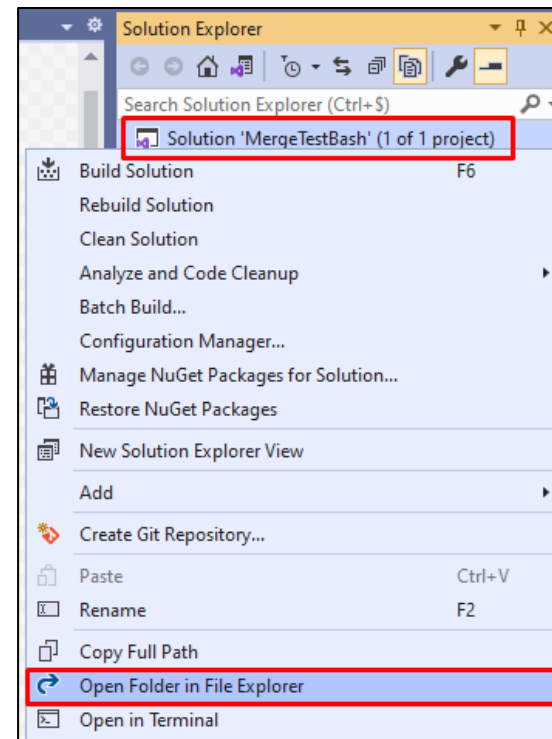


Vorbereiding

- Git Bash:
 - Of: Navigeer naar de folder met Git Bash waarin je het project wil starten
 - Of: RMK Solution in VS > “Open Folder in File Explorer” > RMK Folder in File Explorer > Git Bash Here

```
20003820/source/repos/MergeTestBash
```

```
MINGW64 ~/source/repos/MergeTestBash
```

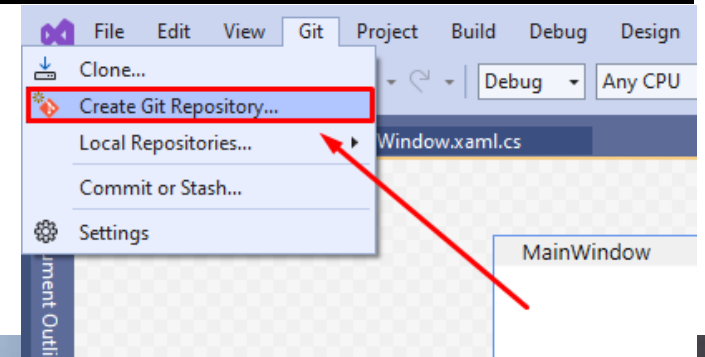


Vorbereiding

- Git Bash: 
 - Of: Navigeer naar de folder met Git Bash waarin je het project wil starten
 - Of: RMK Solution in VS > “Open Folder in File Explorer” > RMK Folder in File Explorer > Git Bash Here
 - > **git init**
 - Dit commando **start** een **git repository**

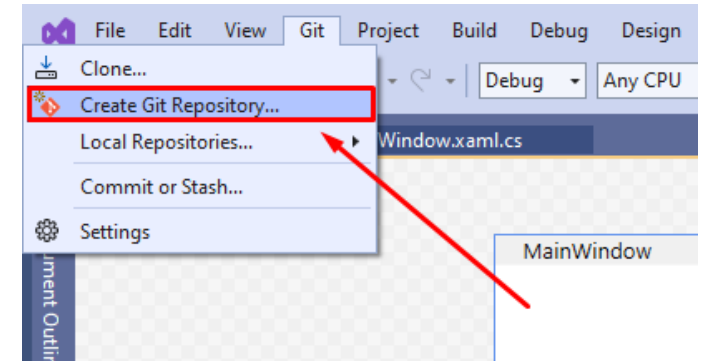
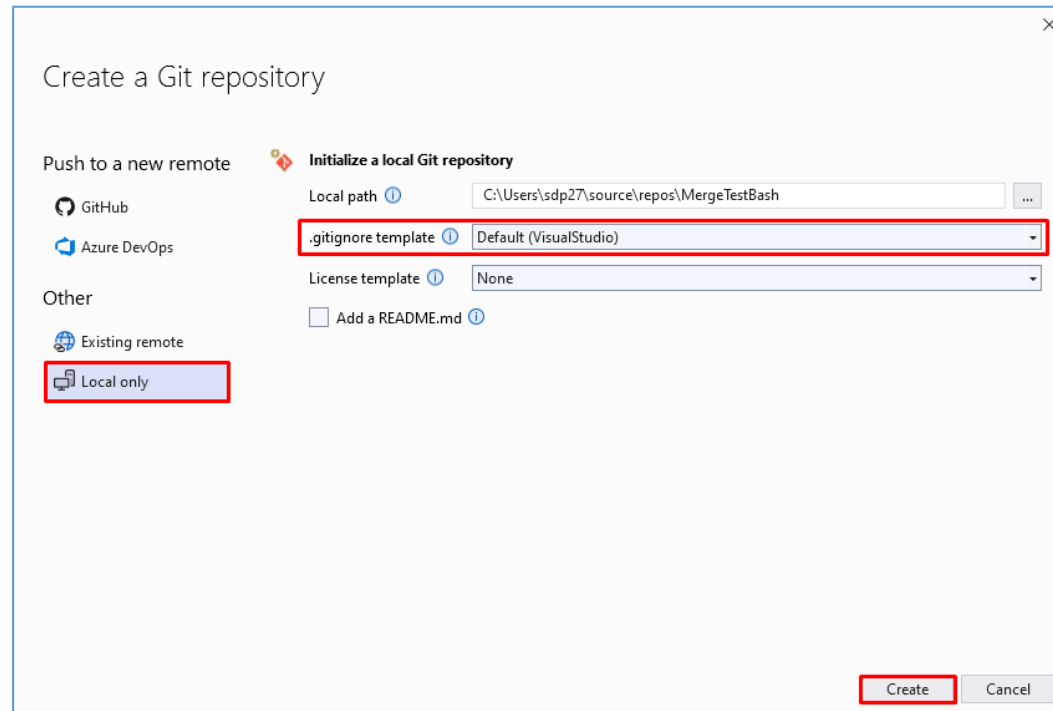
```
$ git init
Initialized empty git repository in c:/Users/20003820/source/repos/MergeTestBash/.git/
```

- Visual Studio: 
 - Of: Menu Git > Create Git Repository ... > Local Only



Voorbereiding

- Visual Studio:
 - Of: Menu Git > Create Git Repository ... > Local Only
 - .gitignore template
 - Create



Voorbereiding



LET OP: je kan ook onmiddellijk de lokale repo op een remote repo zetten, als je de remote repo al hebt aangemaakt

- Visual Studio:
 - Of: Menu Git > Create Git Repository ... > Existing remote
 - .gitignore template
 - **Plak de URL van de repo**
 - Create

Create a Git repository

Push to a new remote

- GitHub
- Azure DevOps

Other

- Existing remote**
- Local only

Initialize a local Git repository

Local path C:\Users\sdp27\source\repos\MergeTestBash

.gitignore template Default (VisualStudio)

License template None

☐ Add a README.md

Push your code to an existing remote

Remote URL **<Required>**



Voorbereiding

- Via git status zien we alle files van het project als unstaged
 - Indien je met Git Bash je project start, dan heb je nog geen .gitignore
 - Maar de “.vs” folder moet niet deel zijn van ons project
 - Maak een “.gitignore” file aan voor Visual Studio:
<https://github.com/github/gitignore/blob/master/VisualStudio.gitignore>

```
$ git status
On branch master

No commits yet

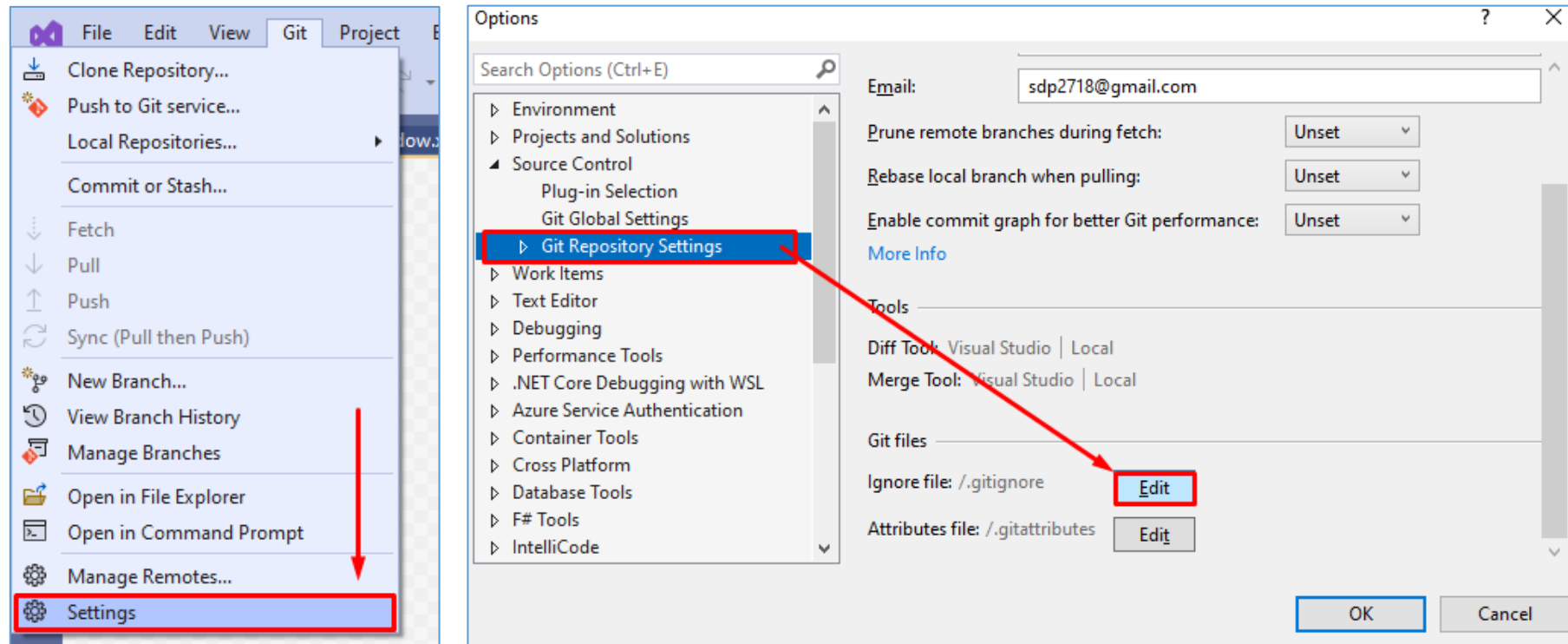
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .vs/
    MergeTestBash.sln
    MergeTestBash/

nothing added to commit but untracked files present (use "git add" to track)
```

Voorbereiding



- Visual Studio:
 - Je kan de .gitignore file later ook aanpassen/aanmaken via Visual Studio
 - Git > Settings > Git Repository Settings > Edit/Add





Voorbereiding

- Sla de .gitignore file op in je git project
 - > git add .gitignore
 - > git add MergeTestBash
 - > git add MergeTestBash.sln
 - > git commit -m "Project Created"

Naam	Gewijzigd op	Type
.git	24/10/2021 18:11	Bestandsmap
.vs	24/10/2021 18:02	Bestandsmap
MergeTestBash	24/10/2021 18:02	Bestandsmap
MergeTestBash.sln	24/10/2021 18:02	Visual Studio Solu...
.gitignore	24/10/2021 18:16	GITIGNORE-bestand

```
$ git commit -m "Project created"
[master (root-commit) 4bdffa8] Project created
13 files changed, 861 insertions(+)
create mode 100644 .gitignore
create mode 100644 MergeTestBash.sln
create mode 100644 MergeTestBash/App.config
create mode 100644 MergeTestBash/App.xaml
create mode 100644 MergeTestBash/App.xaml.cs
create mode 100644 MergeTestBash/Mainwindow.xaml
create mode 100644 MergeTestBash/Mainwindow.xaml.cs
create mode 100644 MergeTestBash/MergeTestBash.csproj
create mode 100644 MergeTestBash/Properties/AssemblyInfo.cs
create mode 100644 MergeTestBash/Properties/Resources.Designer.cs
create mode 100644 MergeTestBash/Properties/Resources.resx
create mode 100644 MergeTestBash/Properties/Settings.Designer.cs
create mode 100644 MergeTestBash/Properties/Settings.settings
```




Vorbereiding

- Nadat het project is aangemaakt, gaan we een repo klaar maken op bitbucket.
- Nadat de repo is aangemaakt, hebben we de clone URL nodig voor onze volgende stap.

Create a new repository [Import repository](#)

Workspace Guus Flater ▼

Project* Demo ▼

Repository name* MergeTestBash

Access level ☐ Private repository
Uncheck to make this repository public. Public repositories typically contain open-source code and can be viewed by anyone.

Include a README? No ▼

Default branch name master

Include .gitignore? No ▼

[Advanced settings](#)

[Create repository](#) [Cancel](#)

Let's put some bits in your bucket

HTTPS

Get started quickly
Creating a README or a .gitignore is a quick and easy way to get something into your repository.
[Create a README](#) [Create a .gitignore](#)

Get your local Git repository on Bitbucket

Step 1: Switch to your repository's directory

```
1 cd /path/to/your/repo
```

Step 2: Connect your existing repository to Bitbucket

```
1 git remote add origin https://GuusFlater2@bitbucket.org/guusflater2/mergetestbash
2 git push -u origin master
```

[Need more information? Learn more](#)

Step 1: Switch to your repository's directory

```
1 cd /path/to/your/repo
```

Step 2: Connect your existing repository to Bitbucket

```
1 git remote add origin https://GuusFlater2@bitbucket.org/guusflater2/mergetestbash.git
2 git push -u origin master
```

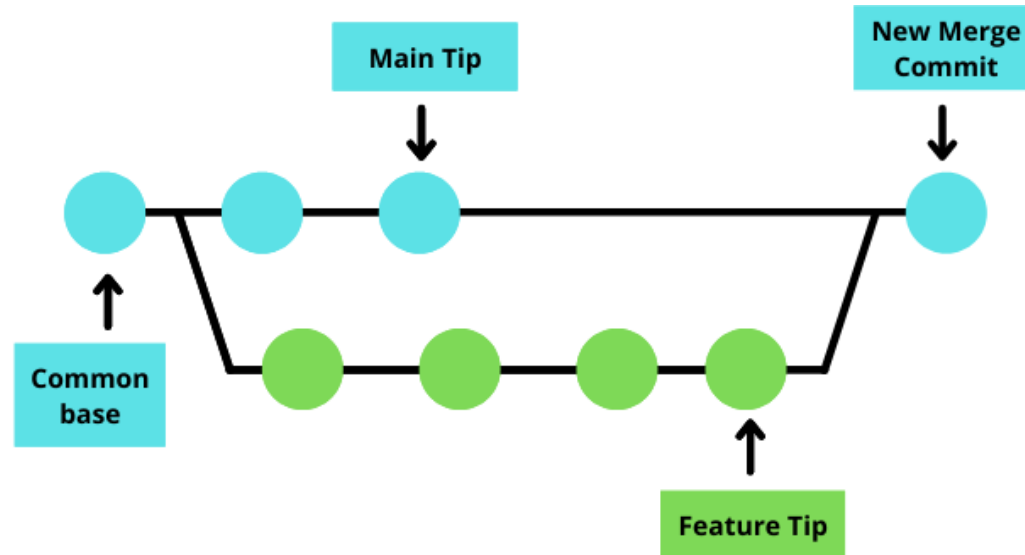


Vorbereiding

- Git Bash:
 - Nu kunnen we de files pushen van onze lokale repo naar de remote repo
 - > git remote add origin <https://username@uw.bitbucket.url/repo.git>
 - > git push -u origin master
- Klaar: Het WPF project staat op de BitBucket server.

Merge

*De verschillen van twee “tijdlijnen”
samenvoegen.*



Merge

- Het git project is klaar voor code:
 - Kies één remote repo per twee om aan te werken. -> **Stuur een verzoek.**

Add members

20003820@pxl.be x

Write
Can push code, create and merge pull requests

Cancel **Confirm**

Guus Flater / Demo

MergeTest

Invite **Set up build** Clone ...

Here's where you'll find this repository's source files. To give your users an idea of what they'll find here, [add a description to your repository](#).

master

Files Filter files

Guus Flater / Demo / MergeTest / Repository settings

User and group access

Add members

Grant access to this repository by adding users and [groups](#). You can find them by name if they already have workspace access. If not, type a full email address to add a new user.
[Learn more about granting access to repositories](#).

For a list of all users with access to any of your private repositories and to see which users count towards your bill, go to [Users on plan](#).

Add members

20003820@pxl.be x

Write
Can push code, create and merge pull requests

Cancel **Confirm**

Merge

- Het git project is klaar voor code:
 - Kies één project per twee om aan te werken. -> Stuur een verzoek.
 - Clone het gekozen project van je partner

Project vereisten:

- Open het project en maak een button
 - Wanneer je op de button klikt, dan wordt er een boodschap getoond: “dit wordt een merge conflict.”
 - Vul deze code in **op je eigen manier!**

Merge

- Open het project en maak een button
 - Wanneer je op de button klikt, dan wordt er een boodschap getoond: “dit wordt een merge conflict.”
 - Voorbeeld mogelijke code:

```
<StackPanel>  
    <Button Margin="20" Click="ToonBerichtButton_Click">  
        Toon Bericht  
    </Button>  
    <TextBlock Name="BerichtTextBlock" Margin="20"  
        HorizontalAlignment="Center">  
    </TextBlock>  
</StackPanel>
```


Merge

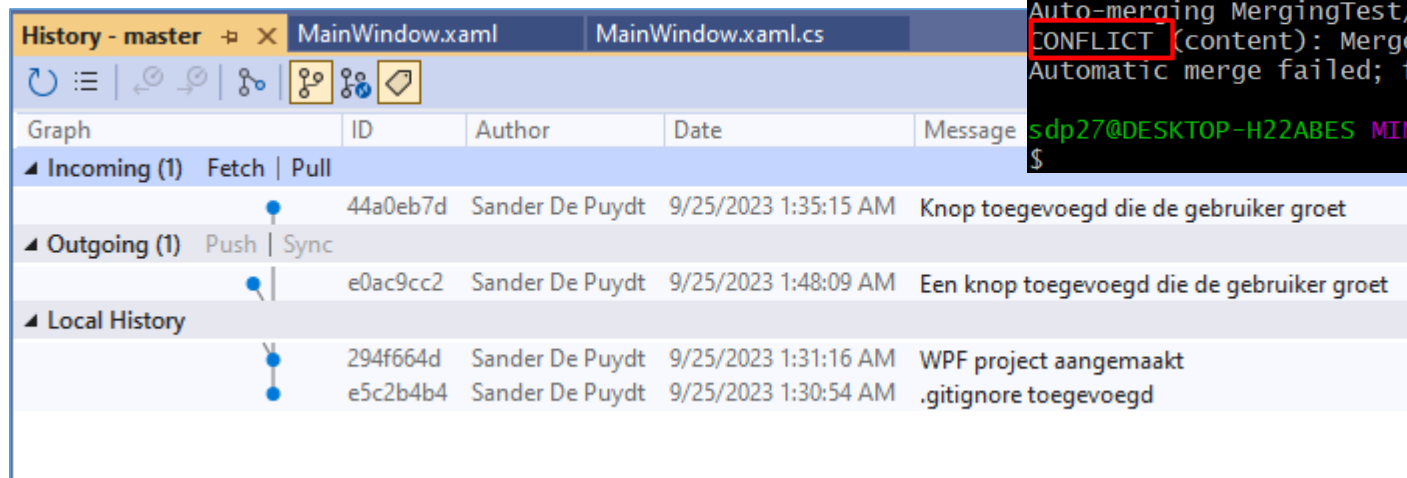
- Open het project en maak een button die tekst toont
 - Voorbeeld mogelijke code:

```
public partial class MainWindow : Window
{
    public MainWindow()
    {
        InitializeComponent();
    }

    private void ToonBerichtButton_Click(object sender, RoutedEventArgs e)
    {
        BerichtTextBlock.Text = "Dit wordt een merge conflict.";
    }
}
```

Merge

- Vervolgens maakt ieder een commit van zijn persoonlijke invulling van de code.
- Degene die als laatste pusht, zal een **merge conflict** krijgen.
 - Git probeert automatisch problemen op te lossen, maar je zal manueel moeten helpen.



The screenshot shows the Git GUI interface with a commit history table. The table has columns for Graph, ID, Author, Date, and Message. It displays three commit categories: Incoming (1), Outgoing (1), and Local History. The Incoming and Outgoing commits show a conflict in the file MainWindow.xaml.cs. The Local History shows the initial project creation and the addition of .gitignore.

Graph	ID	Author	Date	Message
▲ Incoming (1) Fetch Pull				
	44a0eb7d	Sander De Puydt	9/25/2023 1:35:15 AM	Knop toegevoegd die de gebruiker groet
▲ Outgoing (1) Push Sync				
	e0ac9cc2	Sander De Puydt	9/25/2023 1:48:09 AM	Een knop toegevoegd die de gebruiker groet
▲ Local History				
	294f664d	Sander De Puydt	9/25/2023 1:31:16 AM	WPF project aangemaakt
	e5c2b4b4	Sander De Puydt	9/25/2023 1:30:54 AM	.gitignore toegevoegd

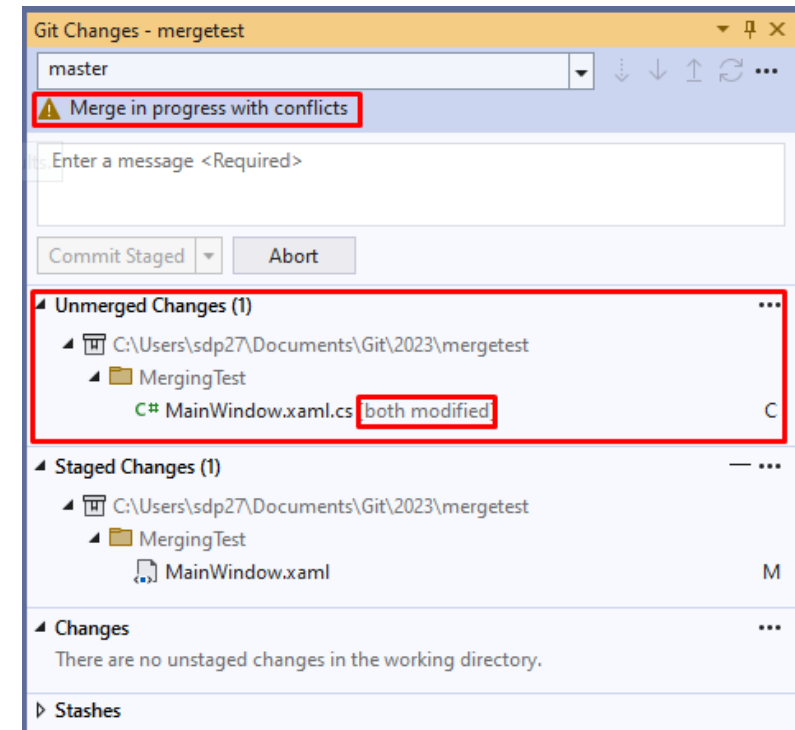
```
sdp27@DESKTOP-H22ABES MINGW64 ~/Documents/Git/2023/mergetest (master)
$ git pull
Auto-merging MergingTest/MainWindow.xaml.cs
CONFLICT (content): Merge conflict in MergingTest/MainWindow.xaml.cs
Automatic merge failed; fix conflicts and then commit the result.
sdp27@DESKTOP-H22ABES MINGW64 ~/Documents/Git/2023/mergetest (master|MERGING)
$
```

Merge

- Voer een “git pull” uit om de aanpassingen van je collega af te halen. Je zal deze aanpassing eerst lokaal moeten verwerken voordat je kan pushen.

```
$ git pull
Auto-merging MergeTest/Mainwindow.xaml.cs
CONFLICT (content): Merge conflict in MergeTest/Mainwindow.xaml.cs
Auto-merging MergeTest/Mainwindow.xaml
CONFLICT (content): Merge conflict in MergeTest/Mainwindow.xaml
Automatic merge failed; fix conflicts and then commit the result.
```

- Hoe zien deze conflicten er uit?
- Hoe kunnen we deze conflicten oplossen?



Merge

- Hoe zien deze conflicten er uit?
 - Ze worden gemarkeerd met “<<<<<< HEAD”, “=====” en “>>>>>>”

```
<<<<<< HEAD
    private void ToonBerichtButton_Click(object sender, RoutedEventArgs e)
    {
        BerichtTextBlock.Text = "Dit wordt een merge conflict.";
====
    private void BerichtButton_Click(object sender, RoutedEventArgs e)
    {
        MessageBox.Show("Dit wordt een merge conflict.");
>>>>>> b184375ff7fec2eaa45a9cb1b7999e951127efde
    }
```

Merge

```
<<<<<< HEAD
  <StackPanel>
    <Button Margin="20" Click="ToonBerichtButton_Click">
      Toon Bericht
    </Button>
    <TextBlock Name="BerichtTextBlock" Margin="20"
      HorizontalAlignment="Center">
    </TextBlock>
  </StackPanel>
=====
  <Grid>
    <Button Content="Button" HorizontalAlignment="Left" Margin="100,200,0,0"
      VerticalAlignment="Top" Width="75" Click="BerichtButton_Click"/>
  </Grid>
>>>>>> b184375ff7fec2eaa45a9cb1b7999e951127efde
```

Merge

- Hoe kunnen we deze conflicten oplossen?
 - Je moet het conflict oplossen door de code te herstellen.
 - Verzorg de applicatielogica.
 - Verwijder de conflict tags wanneer je klaar bent.

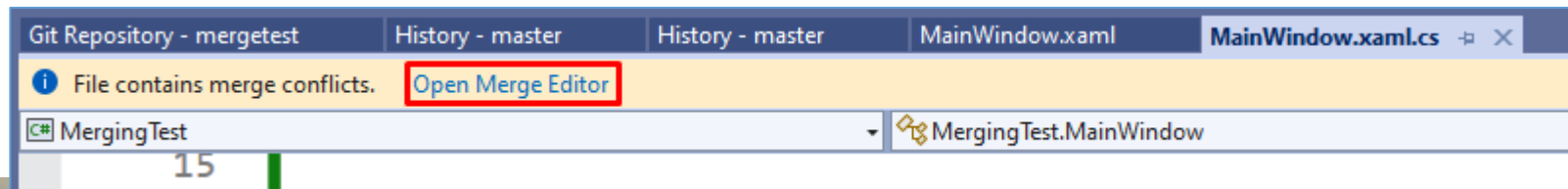
```
<<<<<< HEAD
    private void ToonBerichtButton_Click(object sender, RoutedEventArgs e)
    {
        BerichtTextBlock.Text = "Dit wordt een merge conflict.";
=====
    private void BerichtButton_Click(object sender, RoutedEventArgs e)
    {
        MessageBox.Show("Dit wordt een merge conflict.");
>>>>>> b184375ff7fec2eaa45a9cb1b7999e951127efde
    }
```


Merge

- Hoe kunnen we deze conflicten oplossen?
 - Je moet het conflict oplossen door de code te herstellen.
 - Verzorg de applicatielogica.
 - Verwijder de conflict tags wanneer je klaar bent.

```
private void ToonBerichtButton_Click(object sender, RoutedEventArgs e)
{
    BerichtTextBlock.Text = "Dit wordt een merge conflict.";
    MessageBox.Show("Dit wordt een merge conflict.");
}
```

- Of met de Merge Editor van Visual Studio



Merge

(master | MERGING)

- Wanneer het conflict is opgelost:
 - Voeg de aanpassingen toe aan staging
 - Maak een nieuwe commit
 - Push


```
$ git add .
```

```
$ git commit -m "merge fix"  
[master d2fa038] merge fix
```






```
$ git push  
Enumerating objects: 18, done.  
Counting objects: 100% (18/18), done.  
Delta compression using up to 4 threads  
Compressing objects: 100% (10/10), done.  
Writing objects: 100% (10/10), 1.18 KiB | 172.00 KiB/s, done.  
Total 10 (delta 8), reused 0 (delta 0), pack-reused 0  
To https://bitbucket.org/guusflater2/mergetest.git  
b184375..d2fa038 master -> master
```

Merge

- Op Bitbucket kan je de verschillende “tijdlijnen” van het project zien

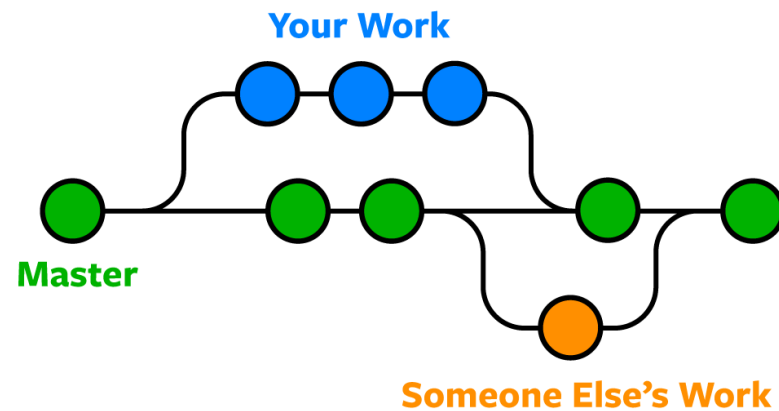


A vertical timeline visualization on the left side of the table shows the sequence of commits. It features a green line with five dots representing commits. A red line branches off from the second commit (Guus Flater) and merges back into the main line at the fifth commit (Guus Flater), illustrating a merge operation.

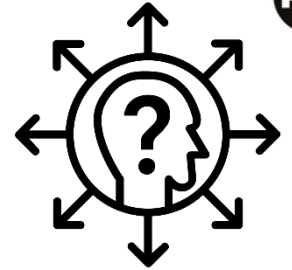
Author	Commit	Message
 Guus Flater	d2fa038	MERGED merge fix
 Guus Flater	8c621ab	Tweede push zorgt voor een merge conflict gelijkaardige aanpassing
 Sander	b184375	First
 Guus Flater	1c03706	Project created Project Created with wpf template
 Guus Flater	ebc9d36	Initial commit

Branch

Een tijdlijn van versies: een lijn van opeenvolgende commits



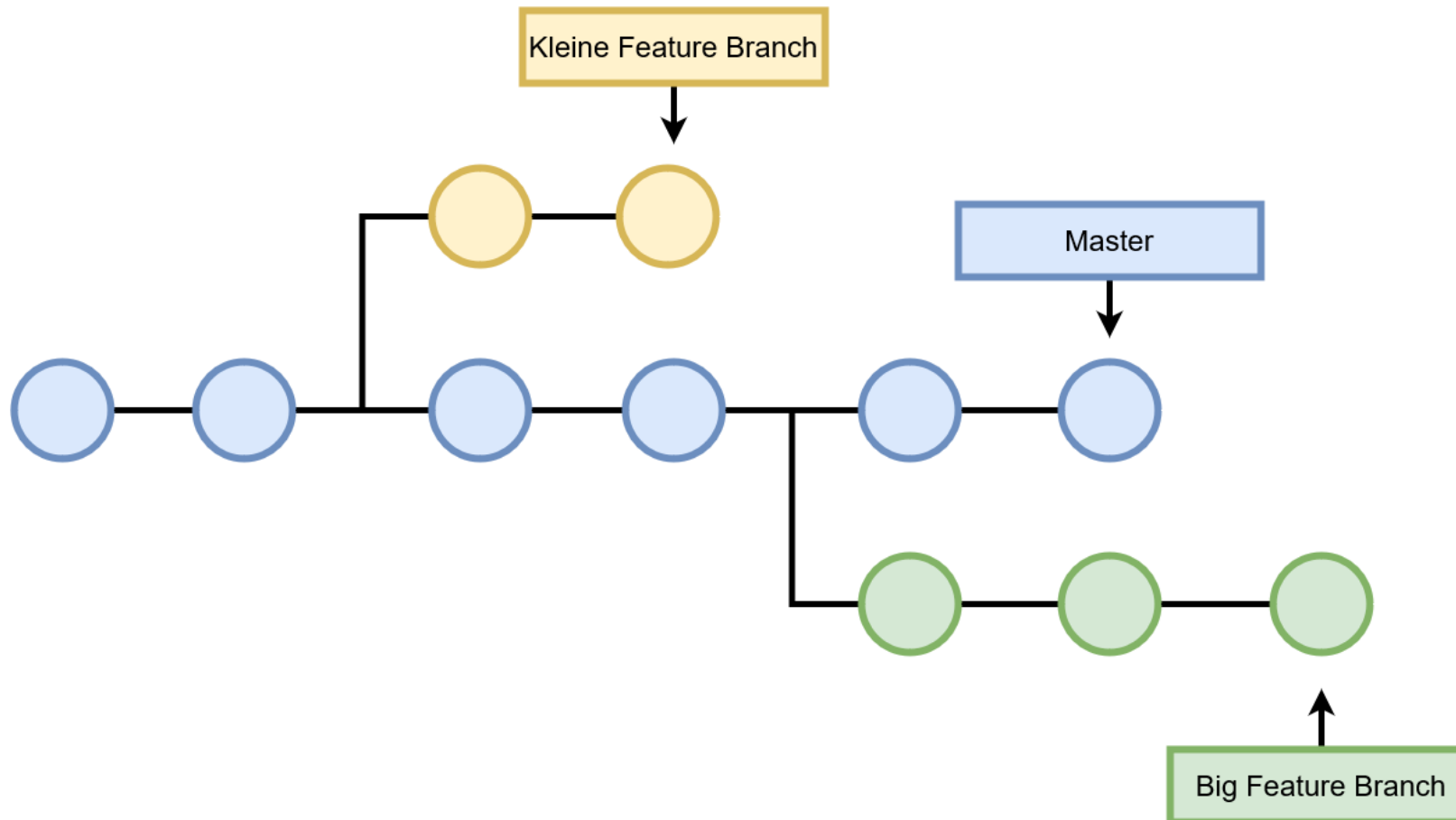
Branch



- Wat als ik een **nieuwe feature** wil maken voor het project, maar de feature zou best in **verschillende commits** gedaan worden?
 - Maken we commits voor een “**unfinished feature**” op de master branch?
 - Wat als deze commits het project **unstable** maken?
- Wat als we **meerdere features tegelijkertijd** willen ontwikkelen?
 - Willen dezelfde features aanpassingen maken in dezelfde code?
- Wat als we in een groter team werken en iedereen regelmatig commits maakt?
 - Moeten we **continu merge conflicts** oplossen?
- Wat als we een **test versie** van het project willen ontwikkelen?

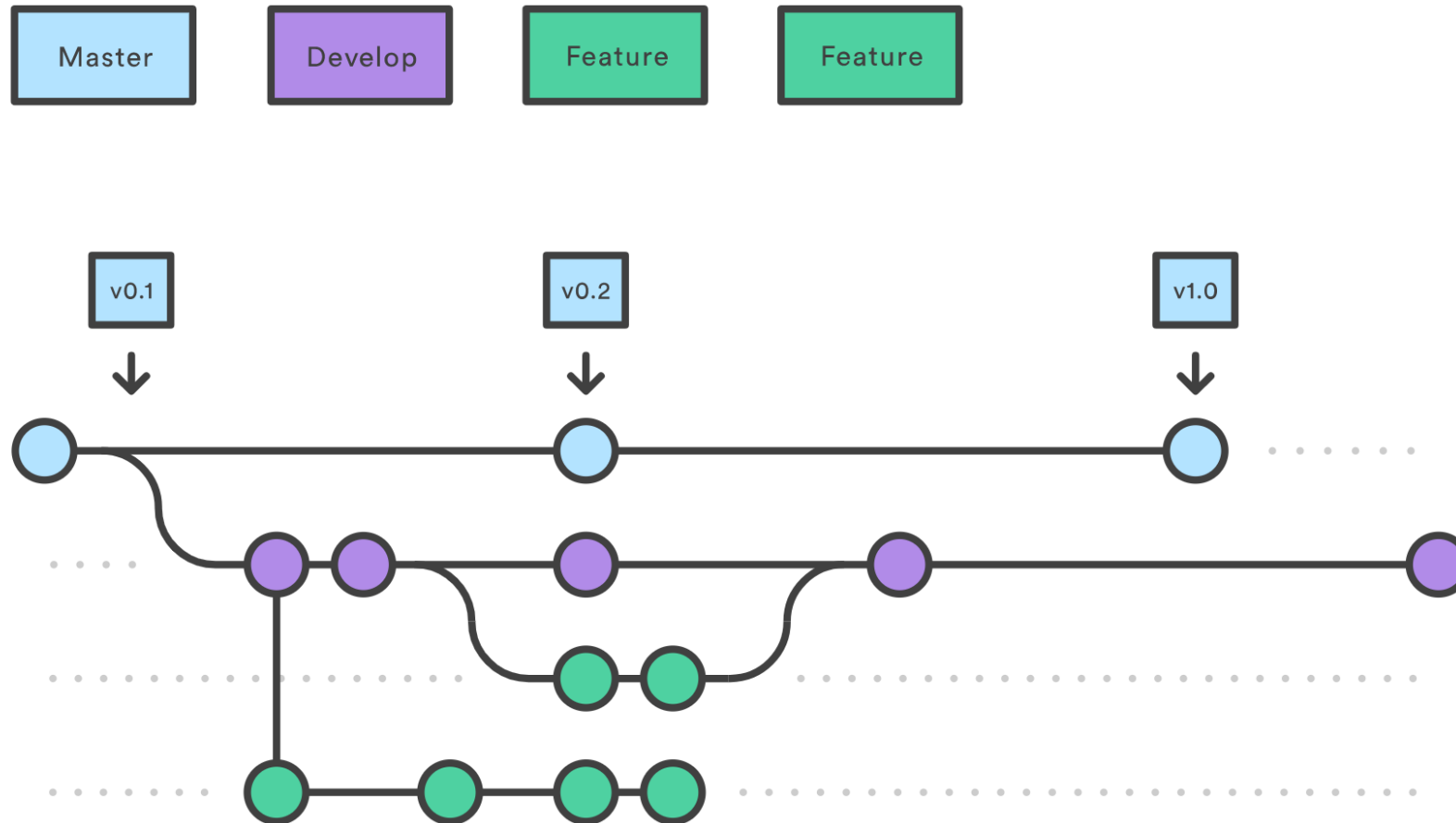
Branch

- Al deze problemen worden opgelost met branches.



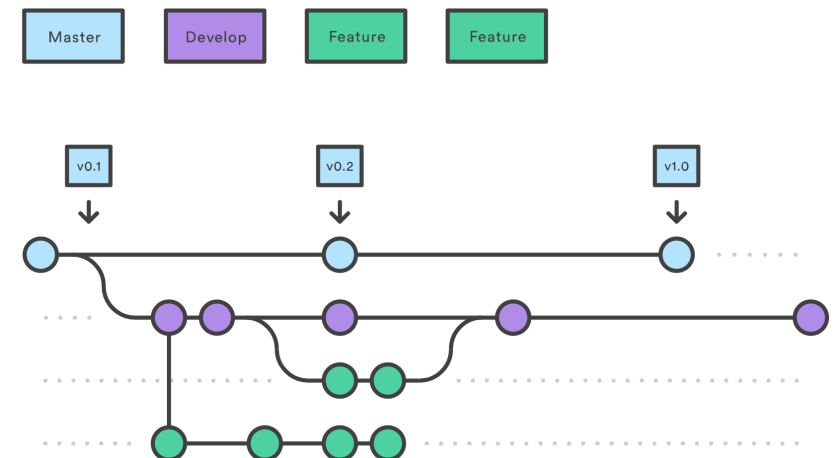
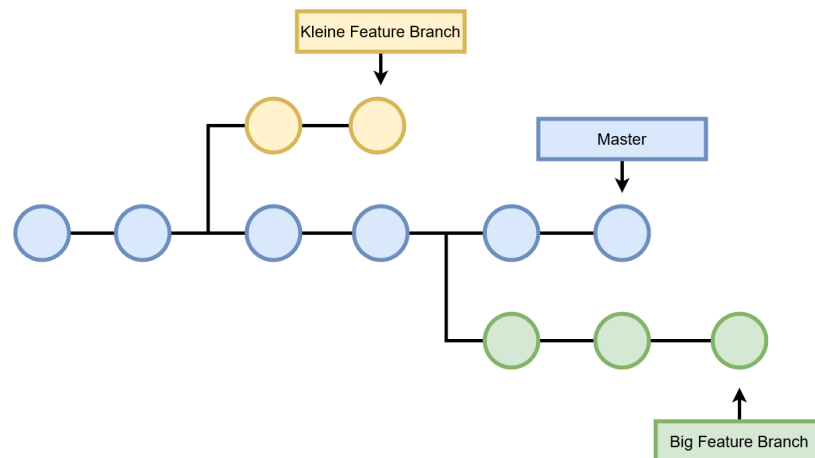
Branch

- Al deze problemen worden opgelost met branches.



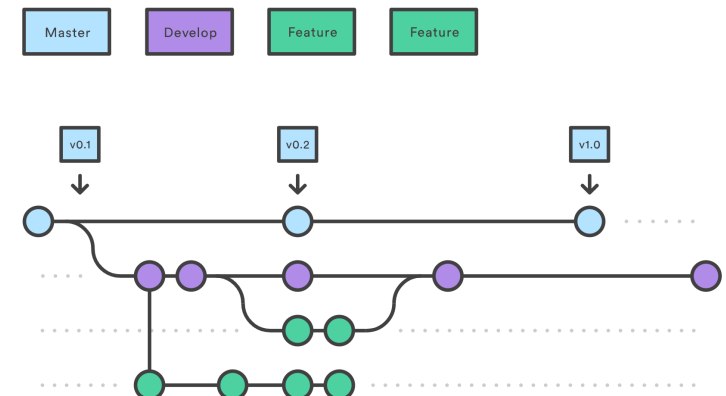
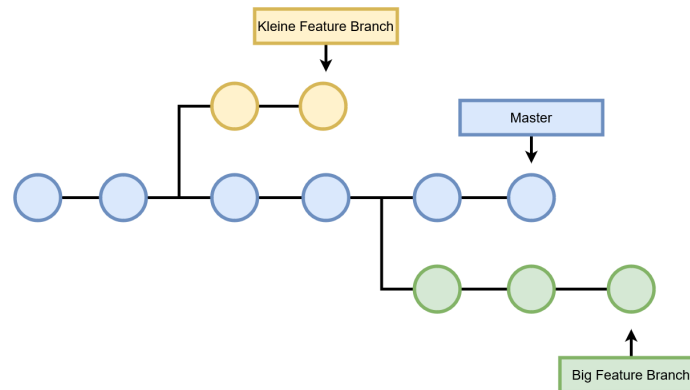
Branch

- Al deze problemen worden opgelost met branches.
- Een branch kan gebruik worden om een feature in te ontwikkelen:
 - Elke ontwikkelaar kan verschillende commits maken op een feature branch zonder conflicten te maken op andere branches.
 - Er is enkel kans op een merge conflict wanneer de branch terug gemerged wordt met de master/main branch.



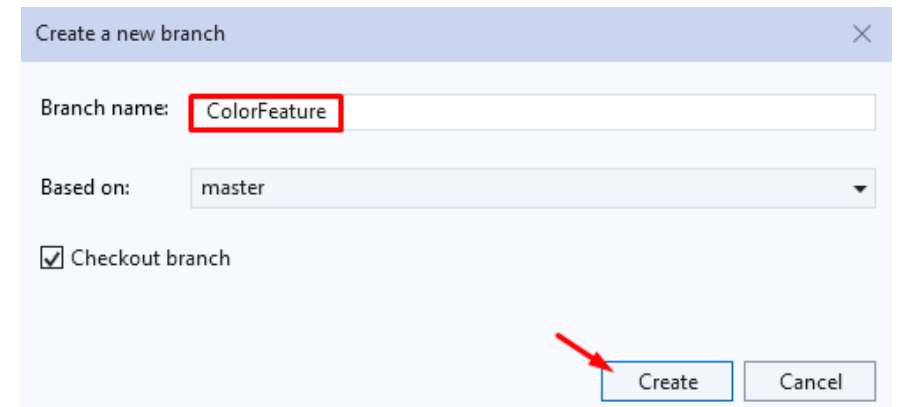
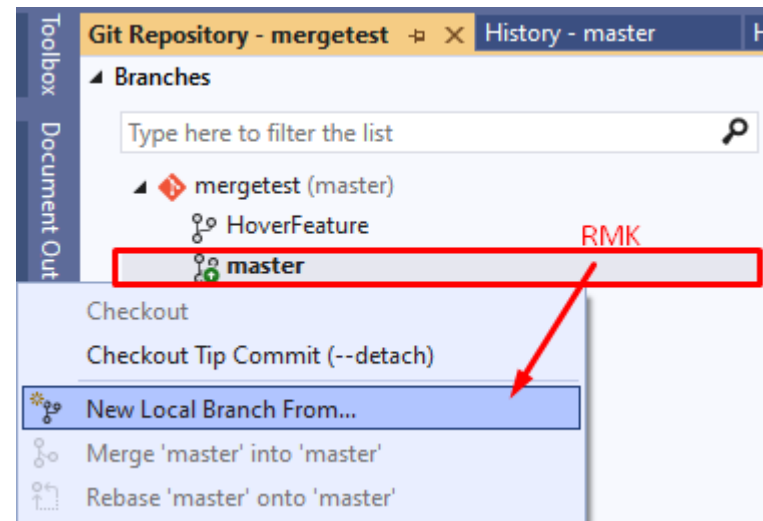
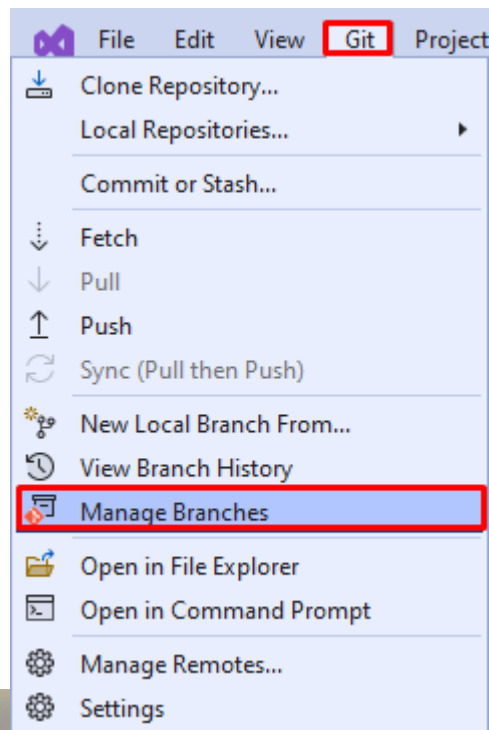
Branch

- Al deze problemen worden opgelost met branches.
- Een branch kan gebruik worden om verschillende versies van het project te maken:
 - Een productie versie die naar de klant gaat. (meestal de master branch)
 - Een develop branch waarin de volgende versie van het project wordt ontwikkelt.
 - Acceptance branch
 - Test branch



Branch


- Visual Studio:
 - Een branch zelf wijst enkel naar een commit.
 - Wanneer we een nieuwe branch aanmaken, dan hebben we nog geen nieuw spoor. Er zijn nu twee branches die naar dezelfde commit wijzen.





Branch

- Visual Studio:
 - Een branch zelf wijst enkel naar een commit.
 - Wanneer we een nieuwe branch aanmaken, dan hebben we nog geen nieuw spoor. Er zijn nu twee branches die naar dezelfde commit wijzen.

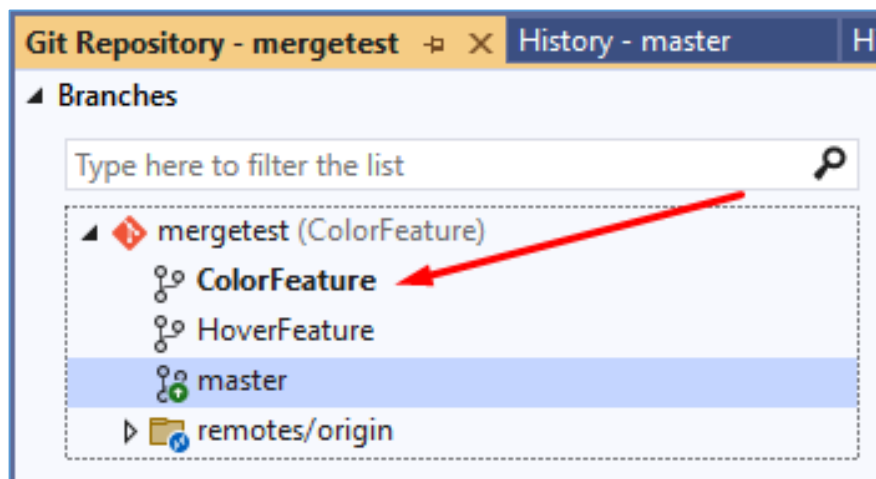
Graph	Message		Author	Date	ID
▷ Incoming (0)	Fetch Pull				
▷ Outgoing (0)	Push Sync				
▲ Local History					
	Merge fix	ColorFeature	Sander De...	9/25/2023...	0c44f364
	Knop toegevoegd die de gebruiker groet	HoverFeature	Sander De...	9/25/2023...	44a0eb7d
	Een knop toegevoegd die de gebruiker groet	master	Sander De...	9/25/2023...	e0ac9cc2
	WPF project aangemaakt		Sander De...	9/25/2023...	294f664d
	.gitignore toegevoegd		Sander De...	9/25/2023...	e5c2b4b4

Zowel de ColorFeature branch als de master branch verwijzen naar dezelfde commit.

Branch



- Visual Studio:
 - Als de naam van de branch vetgedrukt staat, dan is deze branch actief
 - In de onderstaande afbeeldingen is de ColorFeature branch uitgecheckt. Dit betekent dat nieuwe commits op deze branch gemaakt worden.
 - Je kan veranderen van branch door te dubbelklikken op een andere branchnaam.





Branch

- Git Bash Studio:
 - De naam van de actieve branch staat in het blauw achteraan de prompt.
 - In de onderstaande afbeeldingen is de ColorFeature branch uitgecheckt. Dit betekent dat nieuwe commits op deze branch gemaakt worden.

```
sdp27@DESKTOP-H22ABES MINGW64 ~/Documents/Git/2023/mergetest (ColorFeature)  
$
```



Branch

- Git Bash:
 - Een branch aanmaken in Git Bash kan via het “git branch” commando.
 - git branch toont alle branches die lokaal staat in je repository.

```
20003820@5CG70133M3 MINGW64 ~/source/repos/MergeTest (master)
$ git branch
ColorFeature
* master
```

- git branch <BranchName> maakt een nieuwe branch aan die lokaal staat in je repository.

```
20003820@5CG70133M3 MINGW64 ~/source/repos/MergeTest (master)
$ git branch HoverFeature
```



Branch

- Git Bash:

- `git branch -d <BranchName>` verwijdert de gegeven branch uit je lokale repository.
- `git branch -m <BranchName>` hernoemt de huidige branch naar `<BranchName>`.
- `git branch -a` toont alle remote branches.

```
20003820@5CG70133M3 MINGW64 ~/source/repos/MergeTest (master)
$ git branch -a
  ColorFeature
  HoverFeature
* master
  remotes/origin/master
```



Branch

- Git Bash:
 - Wanneer we met Git Bash een branch aanmaken, dan zijn we nog niet aan het werken op de nieuwe branch. Dit is wel het geval in GitKraken.
 - Hoe kunnen we er voor zorgen dat we van master naar een feature branch gaan?

```
20003820@5CG70133M3 MINGW64 ~/source/repos/MergeTest (master)
```

```
$ git branch -a
```

```
ColorFeature
```

```
HoverFeature
```

```
* master
```

```
remotes/origin/master
```

```
20003820@5CG70133M3 MINGW64 ~/source/repos/MergeTest (ColorFeature)
```

```
$ git branch -a
```

```
* ColorFeature
```

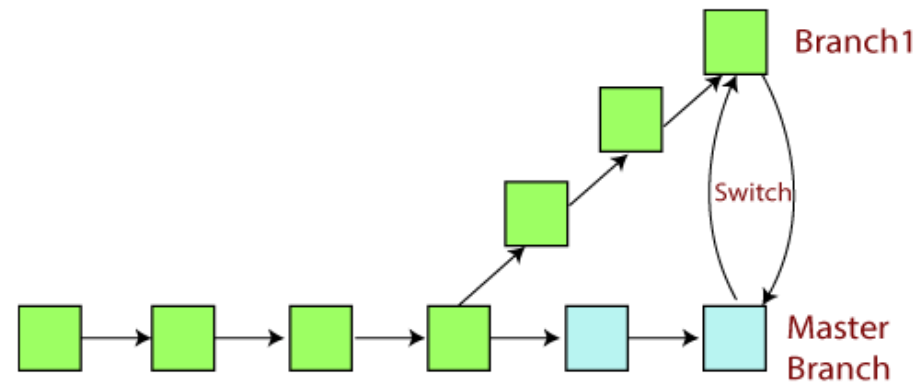
```
HoverFeature
```

```
master
```

```
remotes/origin/master
```

Checkout

Switchen naar een branch



Git Checkout



Checkout

- Met het “`git checkout`” commando kunnen we:
 - Wisselen van branch waarop we werken.
 - Alle nieuwe commits die we maken komen op deze branch
- In Git Bash:



```
20003820@5CG70133M3 MINGW64 ~/source/repos/MergeTest (ColorFeature)
$ git checkout HoverFeature
Switched to branch 'HoverFeature'

20003820@5CG70133M3 MINGW64 ~/source/repos/MergeTest (HoverFeature)
$ git branch
  ColorFeature
* HoverFeature
  master
```

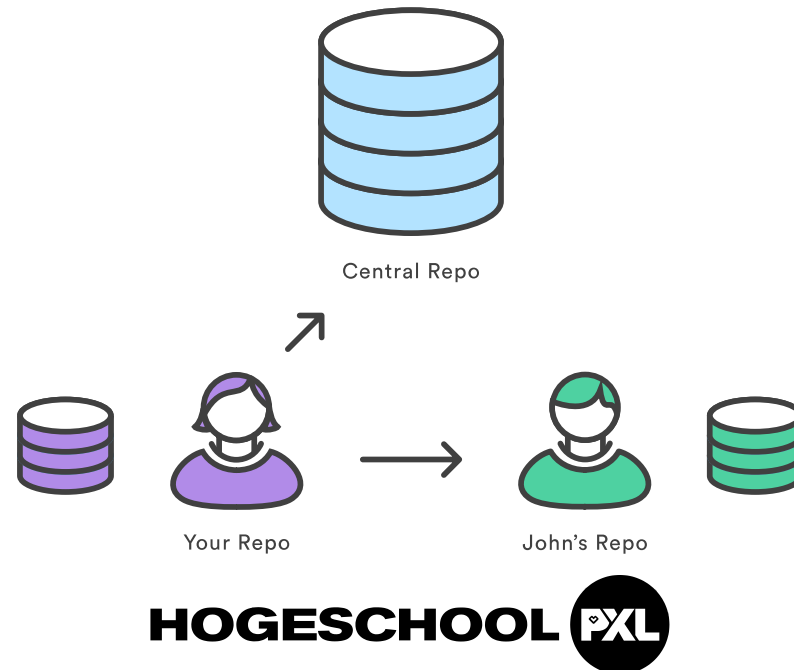



Checkout

- In Git Bash:
 - Als je een remote branch wil uitchecken van je collega, dan moet je deze eerst fetchen (gelijkaardig aan het pull commando voor commits).
 - `git fetch --all` haalt alle remote branches op van de remote repository.

```
20003820@5CG70133M3 MINGW64 ~/source/repos/MergeTest (HoverFeature)
$ git fetch --all
Fetching origin
```

Remote Branches





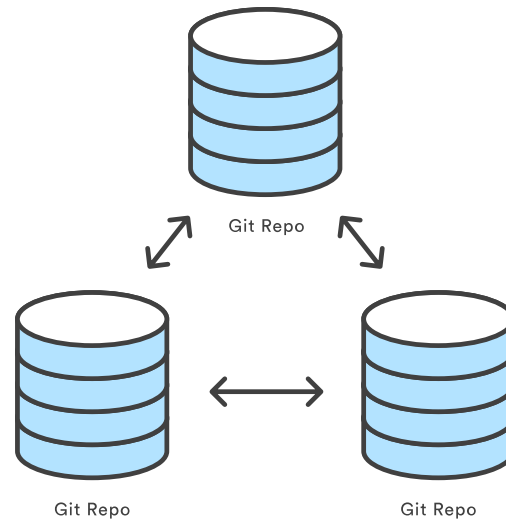
Remote Branches

- In Git Bash:
 - `git checkout <BranchName>`
 - `git push -u origin <BranchName>` pusht de branch naar de remote repository.

```
20003820@5CG70133M3 MINGW64 ~/source/repos/MergeTest (HoverFeature)
$ git push -u origin HoverFeature
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create pull request for HoverFeature:
remote:   https://bitbucket.org/guusflater2/mergetest/pull-requests/new?source=HoverFeature&t=1
remote:
To https://bitbucket.org/guusflater2/mergetest.git
 * [new branch]      HoverFeature -> HoverFeature
Branch 'HoverFeature' set up to track remote branch 'HoverFeature' from 'origin'
.
```

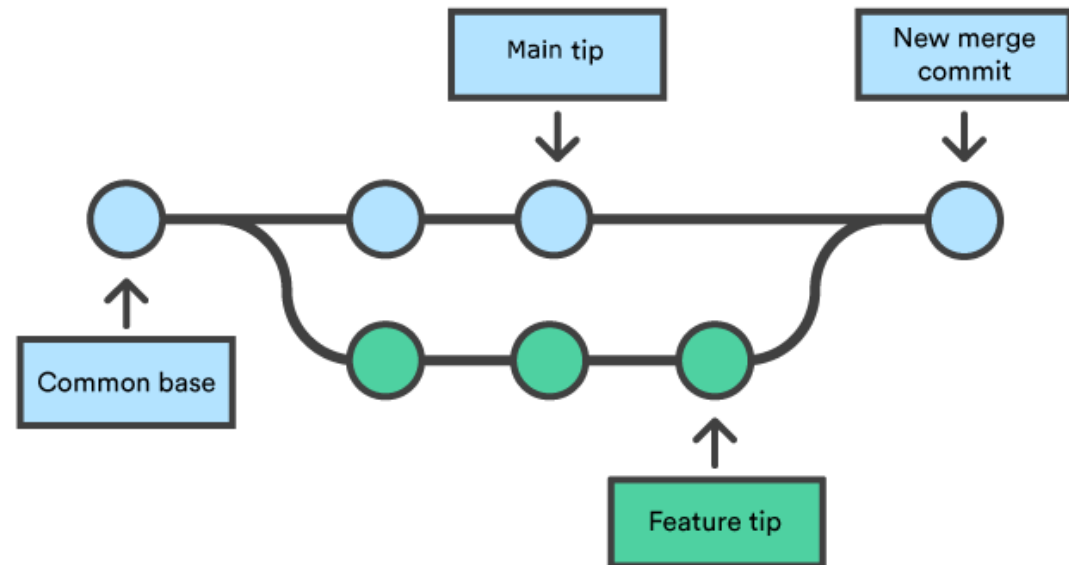
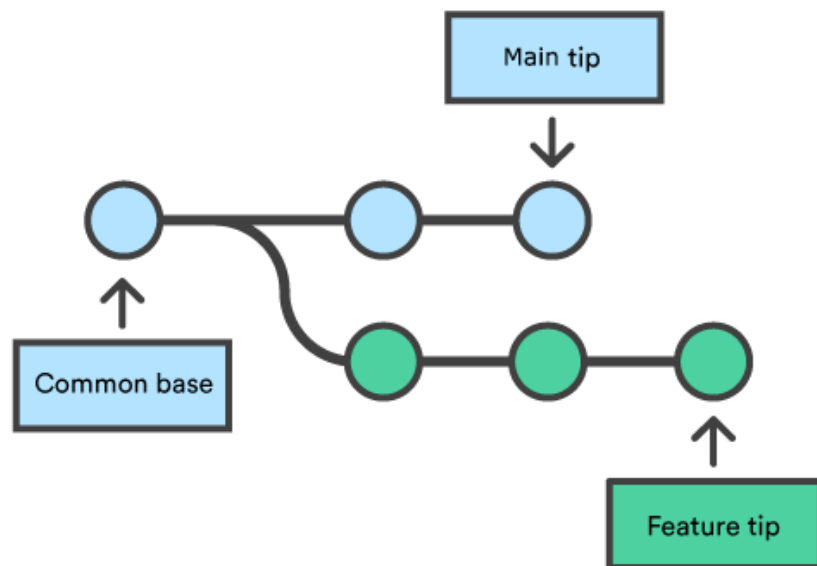
Merging Branches

Repo-To-Repo Collaboration



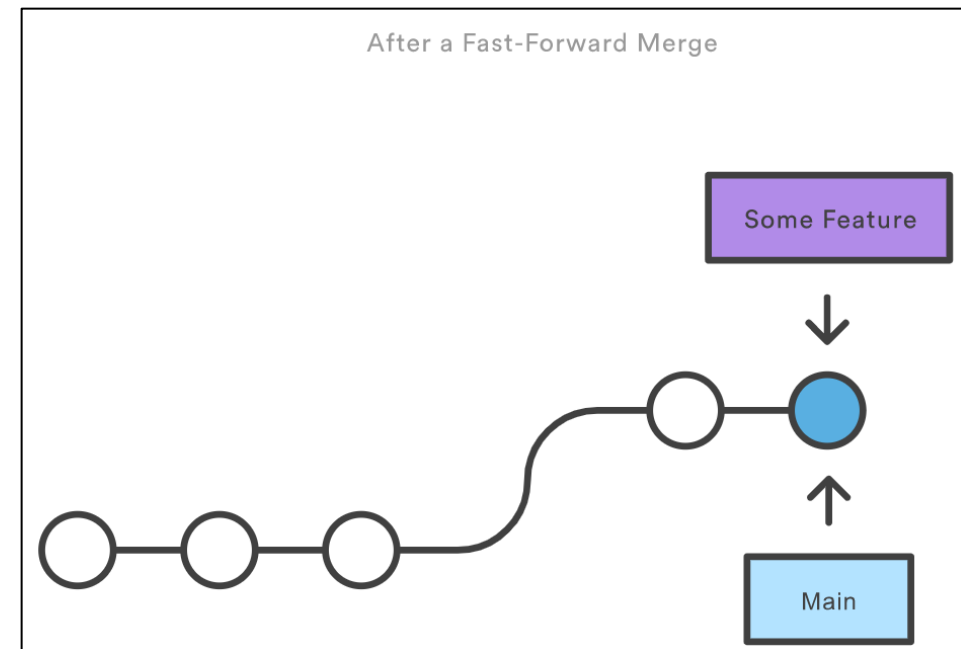
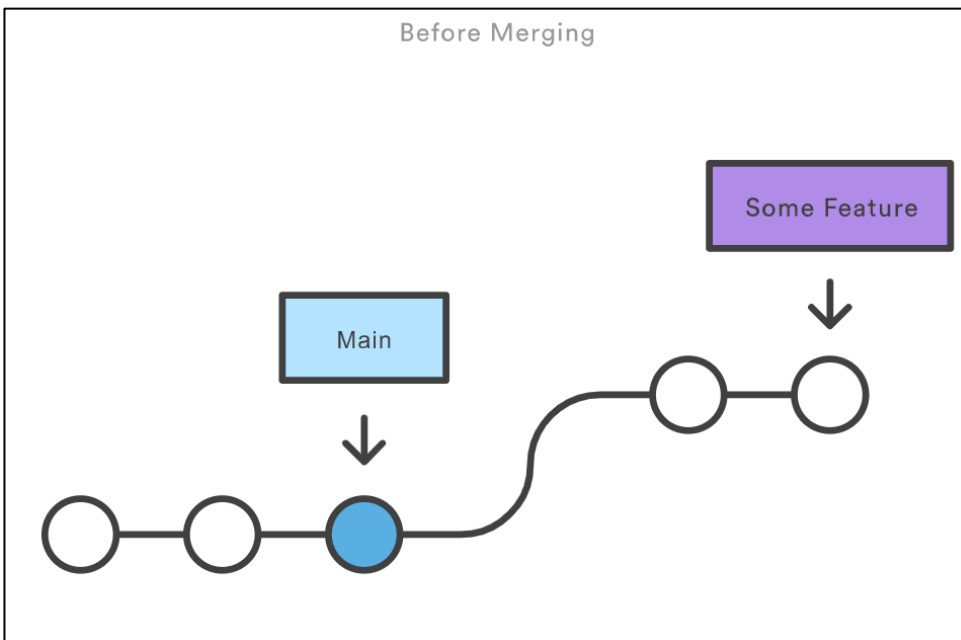
Merging Branches

- De feature is klaar ... wat nu?
 - Hoe krijgen we de aanpassingen van de feature branch naar de master/main branch?
- => We mergen de branch terug naar master.



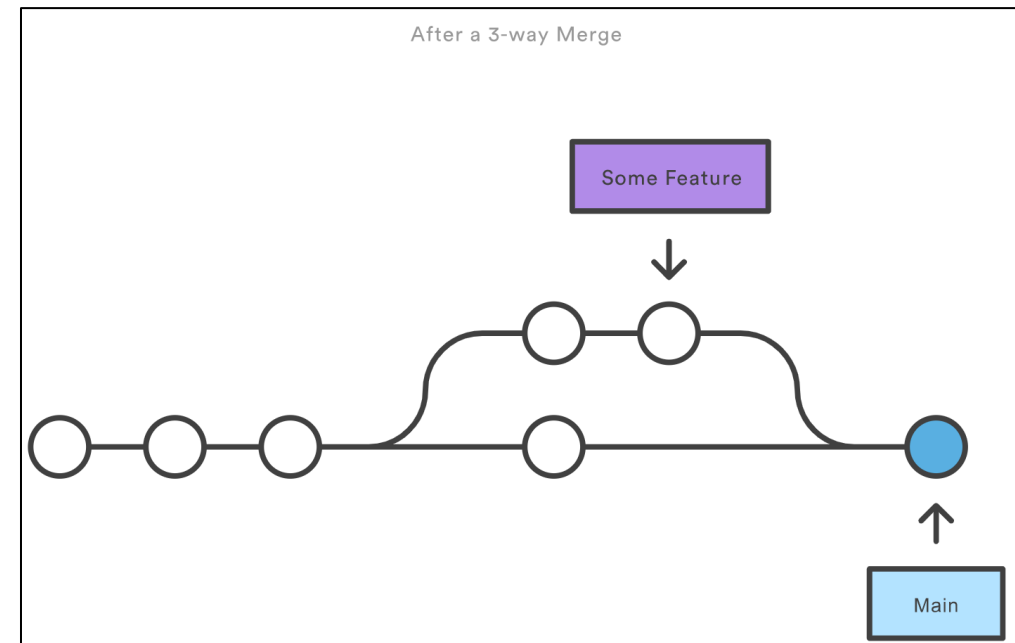
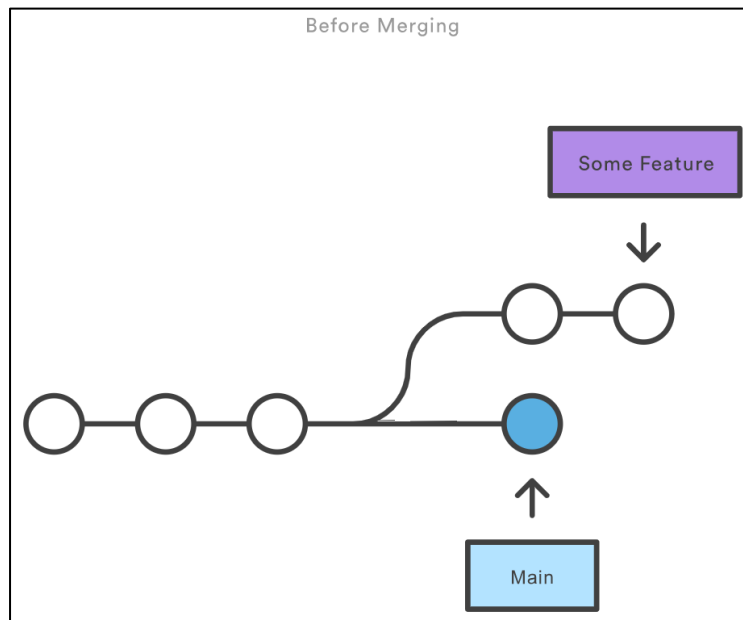
Merging Branches

- Er zijn twee soorten merges:
 - Fast-forward merge: Dit is een merge waarin er een lineair pad kan gemaakt worden tussen de master en feature branch.
 - 3-way merge



Merging Branches

- Er zijn twee soorten merges:
 - Fast-forward merge
 - 3-way merge: Er bestaat geen lineair pad tussen de feature branch en de master branch. Er is een commit nodig om de twee branches samen te binden.





Merging Branches

- In Git Bash:
 - Maak aanpassingen > git add . > git commit -m "boodschap"

```
20003820@5CG70133M3 MINGW64 ~/source/repos/MergeTest (ColorFeature)
```

```
$ git status
```

```
gon branch ColorFeature
```

```
Your branch is up to date with 'origin/ColorFeature'.
```

```
Changes not staged for commit:
```

```
(use "git add <file>..." to update what will be committed)
```

```
(use "git restore <file>..." to discard changes in working directory)
```

```
modified: MergeTest/Mainwindow.xaml
```

```
no changes added to commit (use "git add" and/or "git commit -a")
```

```
20003820@5CG70133M3 MINGW64 ~/source/repos/MergeTest (ColorFeature)
```

```
$ git add .
```

```
20003820@5CG70133M3 MINGW64 ~/source/repos/MergeTest (ColorFeature)
```

```
$ git commit -m "updated the colors to improve UI design"
```

```
[ColorFeature 14eb480] updated the colors to improve UI design
```

```
1 file changed, 2 insertions(+), 2 deletions(-)
```



Merging Branches

- In Git Bash:
 - Er is geen lineair pad voor de ColorFeature branch te verbinden met master. (**3-way merge**)
 - `git checkout master` we willen de master branch uit checken om de feature in te mergen.

```
20003820@5CG70133M3 MINGW64 ~/source/repos/MergeTest (ColorFeature)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
```

- `git merge <BranchName>` we mergen de ColorFeature in master.

```
20003820@5CG70133M3 MINGW64 ~/source/repos/MergeTest (master)
$ git merge ColorFeature
Auto-merging MergeTest/Mainwindow.xaml
CONFLICT (content): Merge conflict in MergeTest/Mainwindow.xaml
Automatic merge failed; fix conflicts and then commit the result.
```



Merging Branches

- In Git Bash:
 - Een merge van branches kan conflicten veroorzaken net zoals een merge conflict van commits die gepusht zijn door collega's.

```
20003820@5CG70133M3 MINGW64 ~/source/repos/MergeTest (master|MERGING)
$ git status
on branch master
Your branch is up to date with 'origin/master'.

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   MergeTest/Mainwindow.xaml

no changes added to commit (use "git add" and/or "git commit -a")
```



Merging Branches

- In Git Bash:
 - Los merge conflict op
 - `git add .`
 - `git commit -m "merge fix: ColorFeature into master"`
- Wanneer je feature af is en je hebt de branch niet meer nodig, dan kan je deze lokaal verwijderen met:
`"git branch -d <BranchName>"`



Merging Branches

- In Git Bash:
 - Branch verwijderen “`git branch -d <BranchName>`”
 - Als je de volgende foutmelding krijgt, dan is je lokale branch die je wilt verwijderen nog niet gepushed (en daarbij remote opgeslagen).

```
20003820@5CG70133M3 MINGW64 ~/source/repos/MergeTest (master)
$ git checkout colorFeature
Switched to branch 'ColorFeature'
Your branch is ahead of 'origin/ColorFeature' by 1 commit.
  (use "git push" to publish your local commits)
```

- `git checkout <BranchName>`
- `git push`
- `git checkout master`
- `git branch -d <BranchName>`



Merging Branches

- In Git Bash:

- `git checkout <BranchName>`
- `git push`
- `git checkout master`
- `git branch -d <BranchName>`

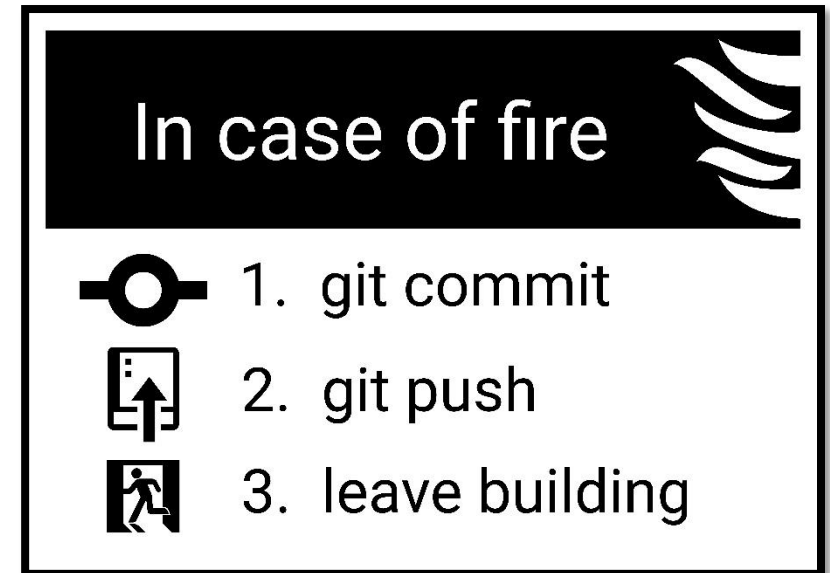
```
20003820@5CG70133M3 MINGW64 ~/source/repos/MergeTest (master)
$ git checkout ColorFeature
Switched to branch 'ColorFeature'
Your branch is ahead of 'origin/ColorFeature' by 1 commit.
  (use "git push" to publish your local commits)
g
20003820@5CG70133M3 MINGW64 ~/source/repos/MergeTest (ColorFeature)
$ git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create pull request for ColorFeature:
remote:   https://bitbucket.org/guusflater2/mergetest/pull-requests/
remote:
To https://bitbucket.org/guusflater2/mergetest.git
   00631da..14eb480  ColorFeature -> ColorFeature

20003820@5CG70133M3 MINGW64 ~/source/repos/MergeTest (ColorFeature)
$ git checkout master
Updating files: 100% (2/2), done.
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

20003820@5CG70133M3 MINGW64 ~/source/repos/MergeTest (master)
$ git branch -d ColorFeature
Deleted branch ColorFeature (was 14eb480).
```

Er is nog veel meer dat je kan leren

- Fork
- Stash
- **Pull request**
- Rebase
- **Reset**
- ...



Bekijk <http://rogerdudler.github.io/git-guide/index.html>
voor een korte samenvatting

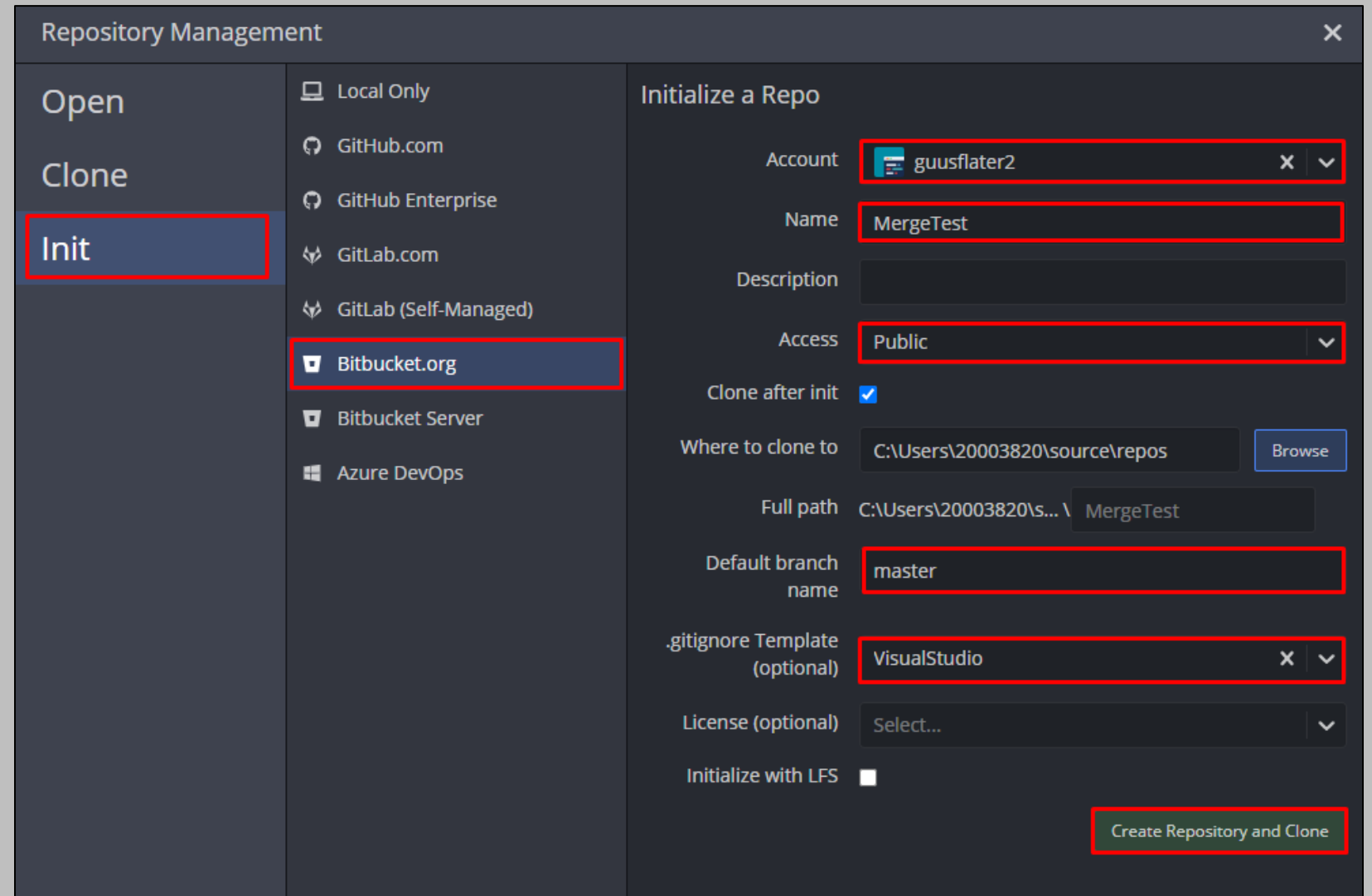
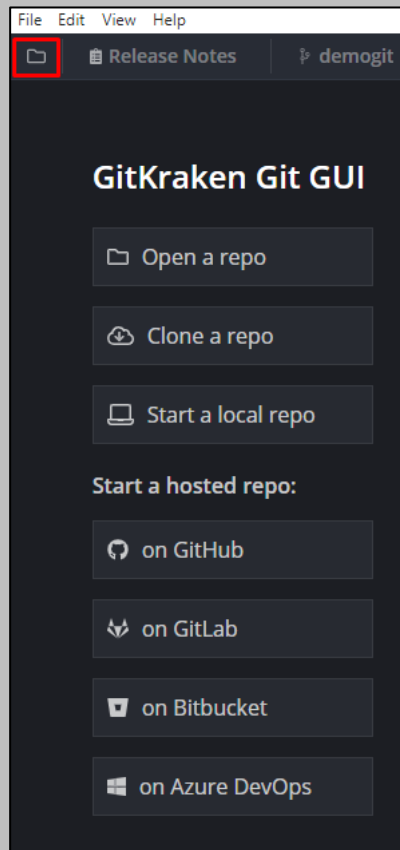
Extra: GitKraken



Vorbereitung



- GitKraken:



Voorbereiding

- GitKraken:

Get started



Clone a repository

Get code from an online repository like GitHub or Azure DevOps



Open a project or solution

Open a local Visual Studio project or .sln file



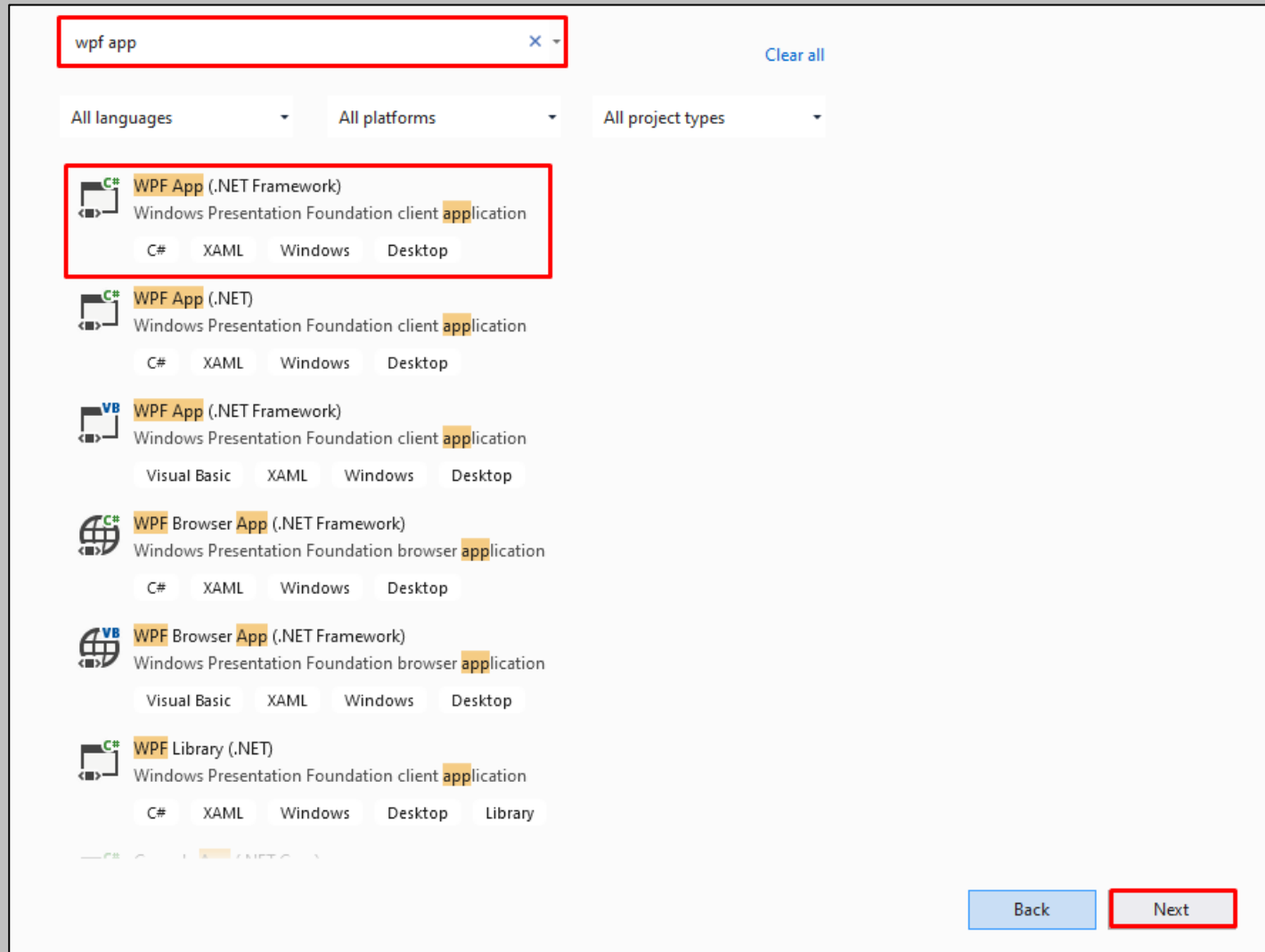
Open a local folder

Navigate and edit code within any folder



Create a new project

Choose a project template with code scaffolding to get started



Vorbereiding

- GitKraken:
 - Het pad en project naam is hetzelfde als het aangemaakte project via GitKraken!

Configure your new project

WPF App (.NET Framework) C# XAML Windows Desktop

Project name

MergeTest

Location

C:\Users\20003820\source\repos

Solution name ⓘ

MergeTest

☐ Place solution and project in the same directory

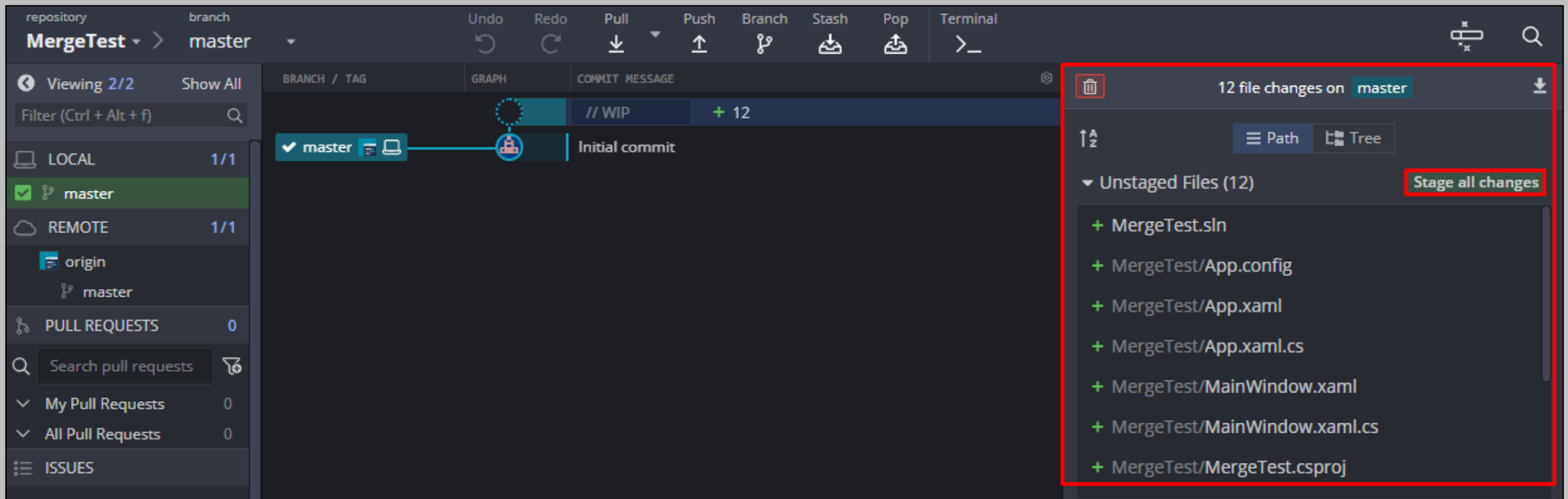
Framework

.NET Framework 4.7.2

Back Create

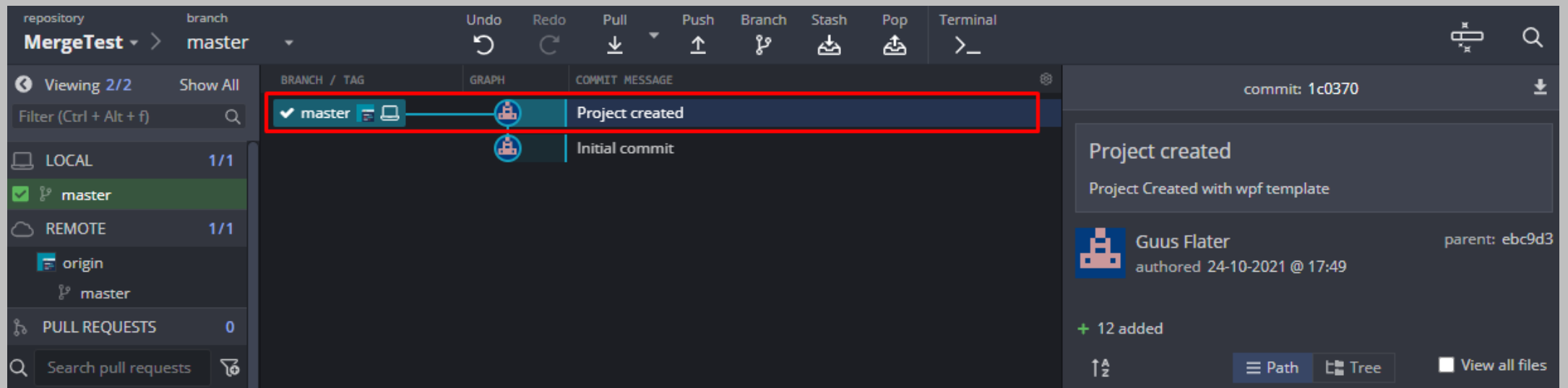
Vorbereitung

- GitKraken:
 - Eens het project is aangemaakt, zie je de aanpassingen in GitKraken.



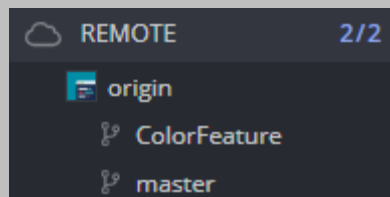
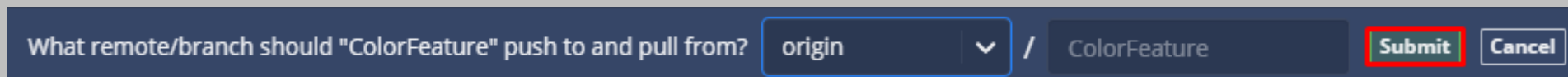
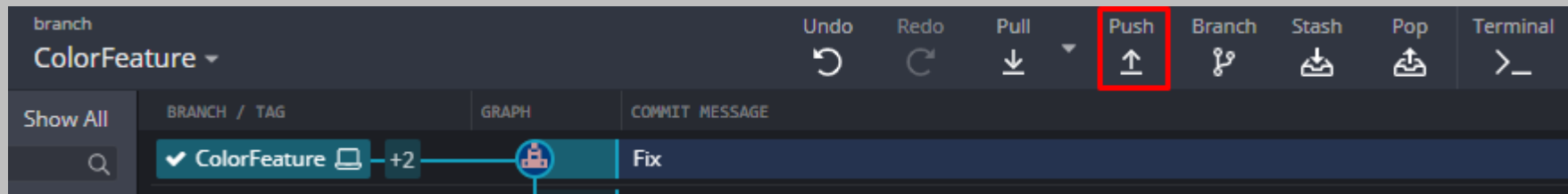
Vorbereiding

- GitKraken:
 - Voeg alle aanpassingen toe aan staging
 - Commit
 - Push
- Klaar: Het WPF project staat op de BitBucket server.



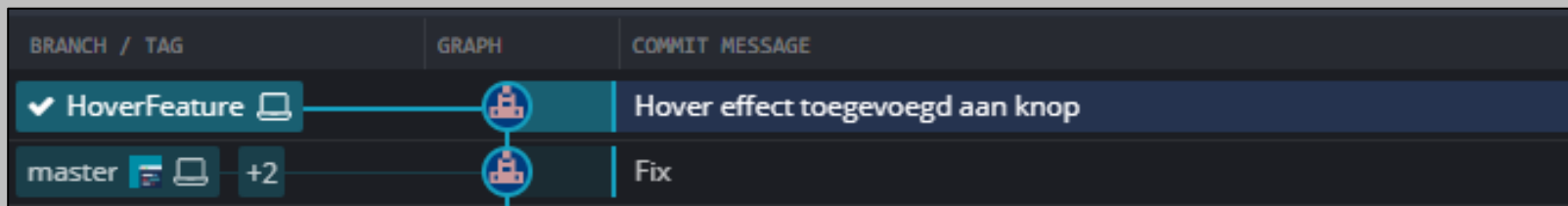
Remote Branches

- **Synchroniseren** van branches met de **remote repo**:
- In GitKraken:
 - Klik Push wanneer de branch checked out is.
 - Submit de branch naar origin met een gepaste naam.



Merging Branches

- In GitKraken:
 - De HoverFeature branch staat één commit voor op master. (**fast-forward**)



- Om te mergen kan je één branch drag en droppen op een andere branch.
- Ofwel RMK op master branch > Merge HoverFeature into master

