



# Workshop & Team (GIT)

"Hur arbetar vi effektivt?"

# Table of Contents

01

Kursplan &  
Översikt

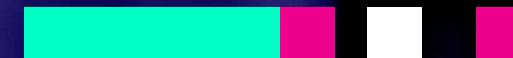
03

Bonus  
Material

02

Pull Requests &  
Github Projects,  
Pair programming  
& Workshop

# Overview



## **Moduler:**

- Code Review
- Pull Requests
- Github Projects
- User-/technical Stories
- Gannt Schema (*Roadmapping*)
- “Pair programming”
- Workshop
- Extreme Programming (*bonus*)
- Shared Codebase (*bonus*)
- Protecting Branches (*bonus*)



## Utbildningsmoment

- **Dataplatfformar, bakgrund och syfte**
- **Git och github i teamkontext** ✓
- **Komponenter och teknologier i en data platform**
- **ETL vs ELT**
- **Utveckling av mjukvara mot databaser**
- **Använda Python mot relationsdatabaser och andra datakällor såsom csv, http xml/json**
- **Använda Python mot realtidsdataströmmar såsom message queues och/eller event streaming platforms**
- **Använda Python och för att rensa, validera och transformera data**
- **Workflow processer**

# 02

Pull Requests & Github Projects,  
Pair programming & Workshop

# Talk LAB



2 min

Discuss the following topics:

- *Hur mycket bugar skulle vi undvika om någon såg min kod direkt?*
- *Brukar du tänka högt när du kodar?*  
Vad tror du hade hänt om du börjat prata högt för dig själv när du kodar?

# Pair Programming

## What is it?

*“Pair programming is an Agile software development technique originating from Extreme programming in which two developers team together on one computer”*

# Pair Programming

*"Pair Programming is a **collaborative** process where **one developer** takes the role of a **driver**, and **the other** takes the role of an **observer**. The **driver writes code** while the **observer reviews** the **code** and **suggests changes**. The two developers **switch roles frequently**, allowing **both** to have **hands-on experience** with the **codebase**."*

**Source:**

<https://medium.com/@teal33t/pair-programming-the-benefits-principles-and-how-to-do-it-right-c975e531ca2b>

# Pair Programming

## But why...?

# \*Secure\*

Studier som gjorts visar på ökad kod-kvalité och bättre säkrad kod

Ökad leverans för applikationens 'features'

# USA's Department of Defense

## (Key Challenges for DoD)

- **Slow Planning vs Fast Change:** Traditional planning doesn't match fast-moving software needs.
- **Cyber Threats:** Must fix vulnerabilities quickly to stay secure.
- **Tech Advancements:** Needs faster dev to keep up with AI and automation.
- **Contractor Dependence:** In-house dev is hard due to hiring and training gaps.
- **Team Rotations:** Frequent personnel changes hurt continuity.

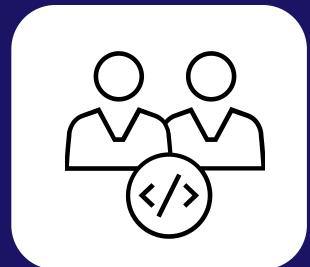
Source: <https://blogs.vmware.com/tanzu/using-extreme-programming-at-the-dod/>



# Proposed Solution

## (How XP can help)

- **Pair Programming** → Boosts code quality & handles staff turnover
- **TDD** → Builds secure, reliable systems
- **Continuous Integration** → Enables rapid updates
- **Small Releases** → Speeds up feedback & reduces risk



**Source:** <https://blogs.vmware.com/tanzu/using-extreme-programming-at-the-dod/>



# Meta-Analysis on Pair Programming (2009)

## Key Findings:

- **Quality:** Pair programming has a small but significant positive effect on code quality.
- **Duration:** It leads to a medium-sized significant reduction in development time.
- **Effort:** However, it requires a medium-sized significant increase in effort compared to solo programming.

**Source:** <https://www.sciencedirect.com/science/article/abs/pii/S0950584909000123>



## Problem

Pair programming...?



## Solution

Parprogrammering är en programvaruteknik där två utvecklare kodar inom en kodbas på en dator, en utses till 'driver' och 'navigator'. **Nyttan:** högre kod kvalité, ökad säkerhet och snabbare leverans.

*Frågor?*



# Pair Programming

## How is it done?

# Terminologies

## **Navigator**

Granskar kod och vägleder föraren. Planerar nästkommande steg.



## **Driver**

Implementerar kod och följer instruktionerna från navigatorn.



## **Ping Pong Pairing**

En programmeringsmetod där deltagarna ofta växlar roller



# Pair programming (Principles)

## Förare (Driver)

- Skriver själva koden
- Fokuserar på syntax, verktyg och implementeringsdetaljer
- Följer instruktioner från navigatören
- Tänker lokalt – "Hur löser jag det här?"

## Navigatör (Navigator)

- Granskar och vägleder föraren
- Tänker på helheten (design, edge cases, mål)
- Upptäcker buggar, föreslår alternativ och planerar nästa steg
- Tänker globalt – "Löser vi rätt problem på rätt sätt?"

# Pair programming (Things to avoid...)

## ✗ Förfare (Driver) bör undvika:

- Att **köra solo** utan att lyssna på navigatören
- Att **ta designbeslut** på egen hand
- Att **glömma att tänka högt**, vilket gör det svårt för navigatören att hänga med
- Att **koda för snabbt** utan reflektion eller feedback

## ✗ Navigatör (Navigator) bör undvika:

- Att vara **för passiv** – du är inte en åskådare
- Att **mikrostyra** varje tangenttryckning – lita på förarens kompetens
- Att **förlora fokus** eller multitaska
- Att **hoppa för långt fram** utan att lösa det nuvarande problemet

# **Navigator** **(Feedback to Driver)**

Observerar ett  
eventuellt  
problem



Navigator

*"Fungerar det även om listan är  
tom?"*

*"Borde vi inte kalla på  
toString()?"*

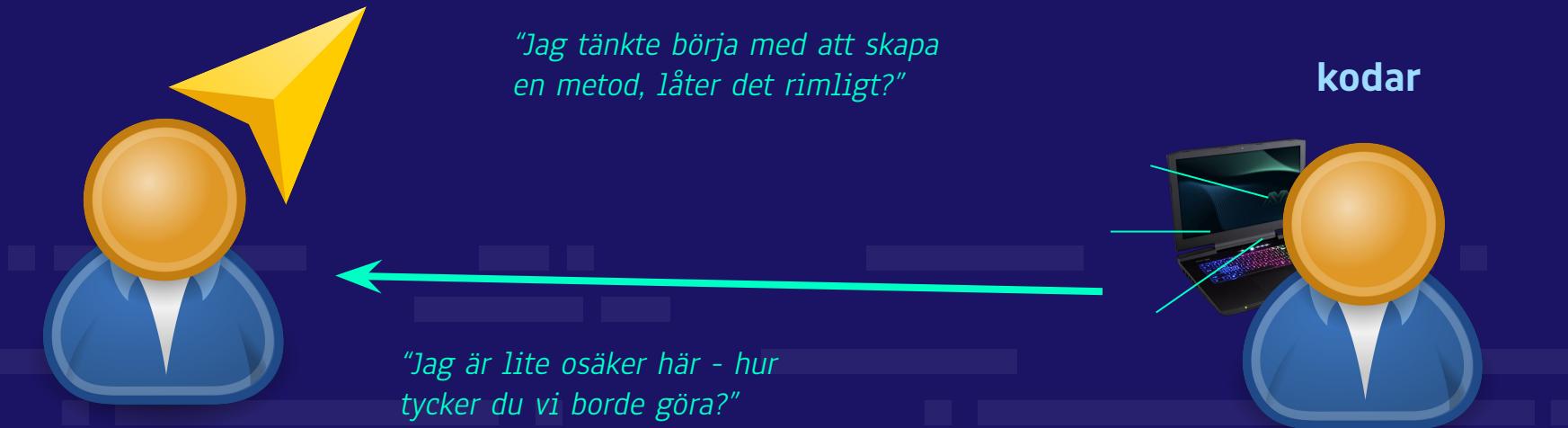
kodar



Driver

*"Jag tror att testet kan börja fallera,  
vad tror du?"*

# Driver (Communicating)



"Jag testar att loopa igenom listan först, så ser vi vad som händer"

# Talk LAB

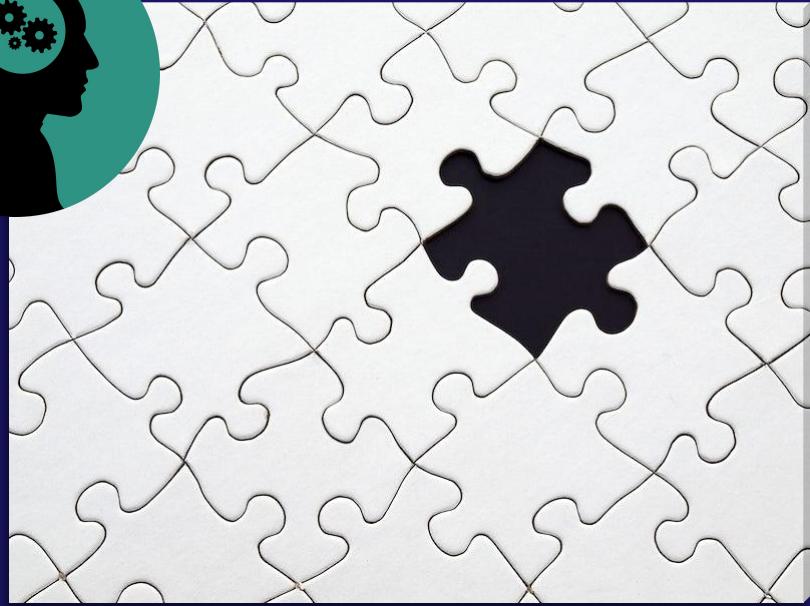


2 min

Discuss the following topics:

- **Kodar Förfaren bara?**  
*Hur samarbetar föraren med navigatorn?*
- **Vad bör Navigeraren kommunikera till Förfaren?**  
*Får navigatoren berätta hur kod skrivas?*

*Frågor?*



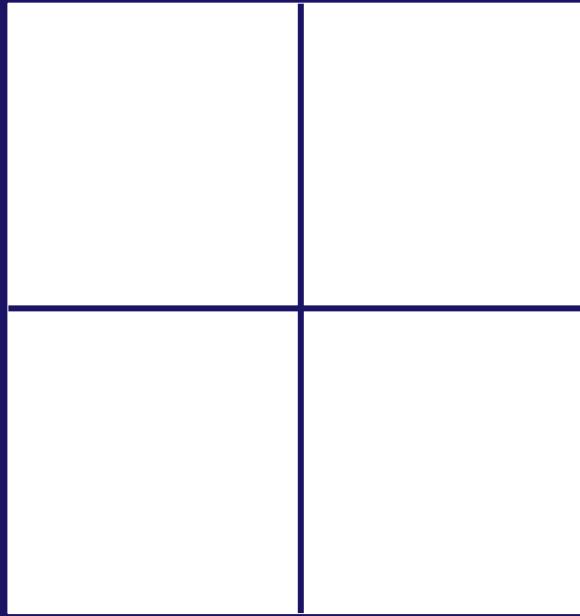
# Pair Programming Practical Example



Dra linjer så att fyra  
fyrkanter skapas.

Utse någon till ritare  
(RITARE MÅSTE VÄNTA SIG  
IVÄG FRÅN LÄRARER)

**2 minuter på sig**











# Pair Programming



## Work Lab

### Börja med att:

- Para ihop er 2-2
- Utse: förare & navigator

Diskutera målet tillsammans innan ni börjar. Målet ska vara förstått för båda parter innan ni börjar. **Har ni förkunskaperna?**

**Undvik:** "Börja med den här funktionen, sen gör det här."

**Navigator:** Tänker strategiskt,  
håller koll på buggar, problem,  
kod struktur & logik

## ÖVERFLÖDA INTE MED INFO

**Förare:** kodar, ställer frågor,  
fokuserar på syntax, följer  
instruktioner från navigatorn.

## TA INTE BESLUT SJÄLV

**Goal:** Find the number that doesn't match the rest.

### Input

```
numbers = [2, 4, 6, 8, 7, 10]
```

### Task

- All numbers except one follow the same rule
- Return the **odd one** (in this case: 7)

### Constraints

- No imports
- One loop max (optional)

**Navigator:** Tänker strategiskt,  
håller koll på buggar, problem,  
kod struktur & logik

## ÖVERFLÖDA INTE MED INFO

**Förare:** kodar, ställer frågor,  
fokuserar på syntax, följer  
instruktioner från navigatorn.

## TA INTE BESLUT SJÄLV

**Goal:** Find the **first value that appears twice**.

### Input

```
numbers = [3, 1, 4, 2, 5, 2]
```

### Expected output

- 3

### Rules

- Return as soon as you find a duplicate
- No sorting allowed

# Kvalitetssäkring Quality Assurance



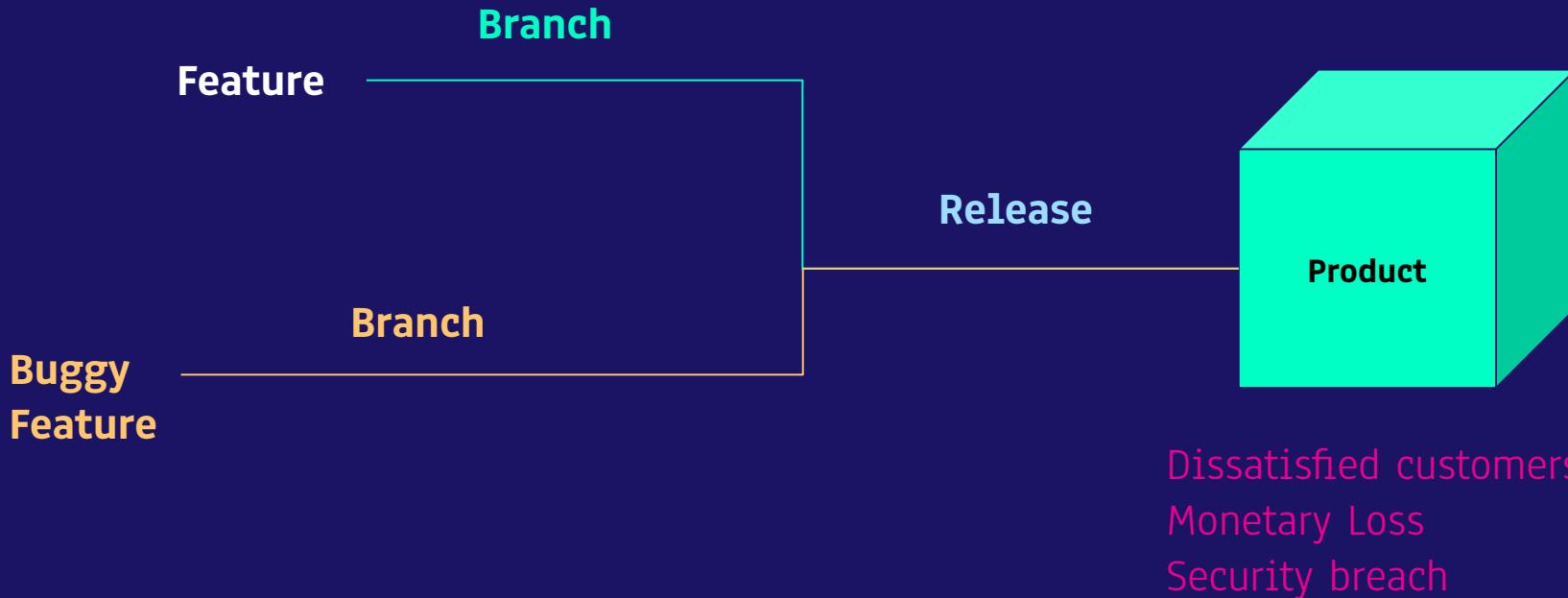
# Branching

## (What is it for?)



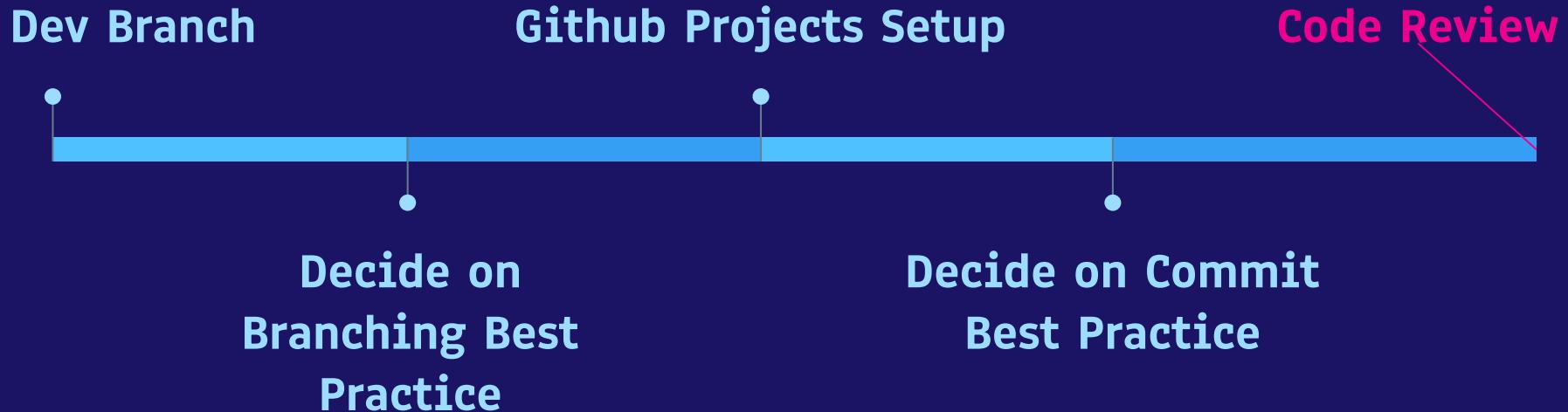
Dröm scenario är att vi alltid släpper buggfria 'features'.

# Quality Assurance (Reality)

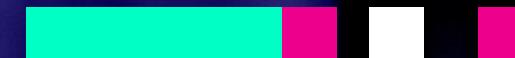


# Quality Assurance

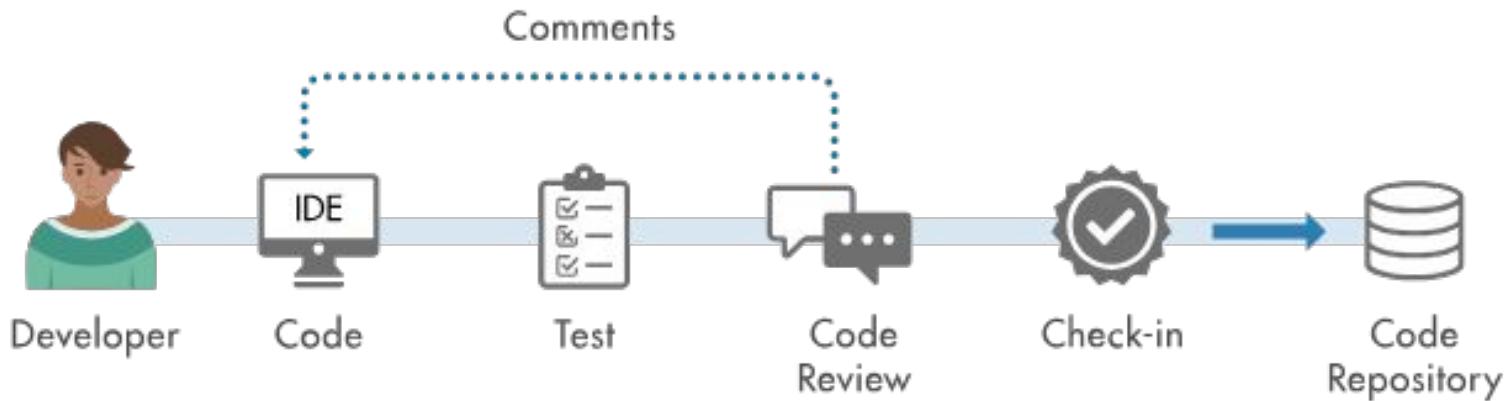
(But we missed something...)



# Code Review Pull Requests



# Developing (Flow)



Code reviews, also known as peer reviews, act as quality assurance of the code base.

Code reviews are methodical assessments of code designed to identify bugs, increase code quality, and help developers learn the source code.

After a software developer has completed coding, a code review is an important step in the software development process to get a second opinion on the solution and implementation before it's merged into an upstream branch like a feature branch or the main branch. The reviewer can also act as a second step in identifying bugs, logic problems, uncovered edge cases, or other issues.

Source: <https://about.gitlab.com/topics/version-control/what-is-code-review/>

# Pull Request

## (Prerequisites)

- #1.** Github Repository
- #2.** Project
- #3.** New Branch (navigate into it)



# Pull Request

## (Showcase: Step by Step)

```
kristofferjohansson@Kristoffers-MacBook-Pro TDD_Test_5 % git branch
* feature/pull-request-showcase
  main
```

*I detta demo använder jag ett äldre redan existerande projekt*

# Pull Request

## (Showcase: Step by Step)

```
kristofferjohansson@Kristoffers-MacBook-Pro TDD_Test_5 % git add .
kristofferjohansson@Kristoffers-MacBook-Pro TDD_Test_5 % git commit -m "Add: Empty Document"
[feature/pull-request-showcase 183ab6e] Add: Empty Document
 1 file changed, 4 insertions(+)
create mode 100644 src/main/java/com/krillinator/demo/PRShowcase.java
```

### Vi tillför:

```
$ git add .
$ git commit -m "Add: Empty Document"
$ git push
```

Notera.. Denna fil jag lagt till är en tom klass ( nästa sida)

# Pull Request

## (Java example)

```
1 package com.krillinator.demo;  
2  
3 public class PRShowcase {  
4 }
```

**Notera:** Realistiskt vill vi aldrig pusha en renсад klass till:  
'main/master' eller 'Release'

```
kristofferjohansson@Kristoffers-MacBook-Pro TDD_Test_5 % git push -u origin feature/pull-request-showcase
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (9/9), 659 bytes | 659.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote: 
remote: Create a pull request for 'feature/pull-request-showcase' on GitHub by visiting:
remote:     https://github.com/Krillinator/PR\_Showcase/pull/new/feature/pull-request-showcase
remote: 
To https://github.com/Krillinator/PR\_Showcase.git
 * [new branch]      feature/pull-request-showcase -> feature/pull-request-showcase
branch 'feature/pull-request-showcase' set up to track 'origin/feature/pull-request-showcase'.
kristofferjohansson@Kristoffers-MacBook-Pro TDD_Test_5 %
```

Notera att vi nu också får chansen att skapa ett pull request...

**Detta från 'main'!**

# Pull Request

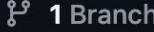
## (Github – Main)

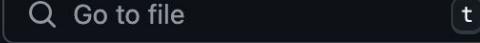
PR\_Showcase Private

 Unwatch 1

 feature/pull-request-showcase had recent pushes 2 seconds ago

 Compare & pull request

 main ▾  1 Branch  0 Tags

 Go to file 

 Add file ▾  Code ▾

Author	Commit Message	Date	Actions
 Krillinator	initial commit	0b0ca5d · 12 minutes ago	 1 Commit
	.idea	initial commit	12 minutes ago
	src	initial commit	12 minutes ago
	.gitignore	initial commit	12 minutes ago

# Pull Request (Simple Template)

**\*\* Pull Request Template\*\***

**\*\*Title\*\*:** \_Feature/pull-request-showcase - a short showcase of the concept\_

**\*\*Description:\*\***

- What this PR does: Adds an empty Class
- Why this change was necessary: It really wasn't

**\*\*Changes Made:\*\***

- Added ugly code to demonstrate why code reviews are important.

**\*\*Testing:\*\***

Tests included (if applicable).

Manual testing done.

**\*\*Notes:\*\***

Any additional comments, potential impacts, or follow-up tasks.

## Add a description

Write

Preview



H B I E <> ⌂ | ½ ⅓ ⅔ | ⌐ @ ⌘ ↵ ↺ ↻

\*\*Pull Request Template\*\*

\*\*Title\*\*: \_Feature/pull-request-showcase - a short showcase of the concept\_

\*\*Description:\*\*

- What this PR does: Adds an empty Class
- Why this change was necessary: It really wasn't

\*\*Changes Made:\*\*

- Added ugly code to demonstrate why code reviews are important.

\*\*Testing:\*\*

Tests included (if applicable).

Manual testing done.

\*\*Notes:\*\*

Any additional comments, potential impacts, or follow-up tasks.

Markdown is supported

Paste, drop, or click to add files

Create pull request

Innan du trycker...  
Kolla gärna på  
'preview' också

## Reviewers

No reviews

## Assignees



Krillinator

## Labels

None yet

## Projects

None yet

## Milestone

No milestone

## Development

Use [Closing keywords](#) in the description to automatically close issues

## Helpful resources

[GitHub Community Guidelines](#)



# Pull Request (Sidepanel)

Notera att vi kan koppla ihop detta till 'github Projects'

 Krillinator self-assigned this now



 **No conflicts with base branch**

Merging can be performed automatically.

**Merge pull request**

You can also merge this with the command line. [View command line instructions.](#)



**Add a comment**

Write

Preview



Add your comment here...

 Markdown is supported

 Paste, drop, or click to add files

 Close pull request

Comment

 Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Notera...  
Vi kan göra en  
merge här...

Men vi kan också  
NEKA förändringen  
och samtidigt ge  
feedback!

LÅT OSS

# Pull Request

## (Navigate)

### Add: Empty Document #1

 Open

Krillinator wants to merge 1 commit into `main` from `feature/pull-request-showcase` 

 Conversation 0

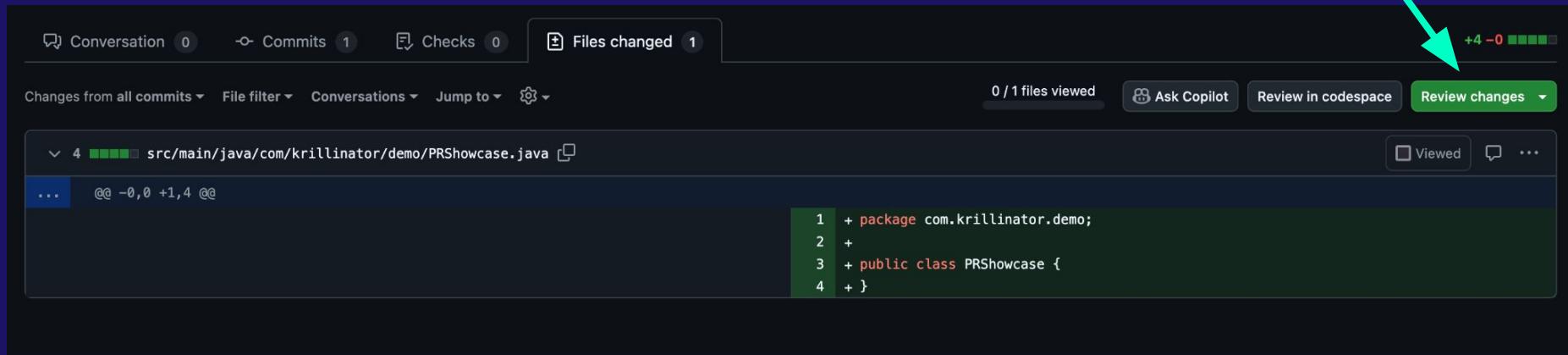
 Commits 1

 Checks 0

 Files changed 1



# Pull Request (Review Changes)

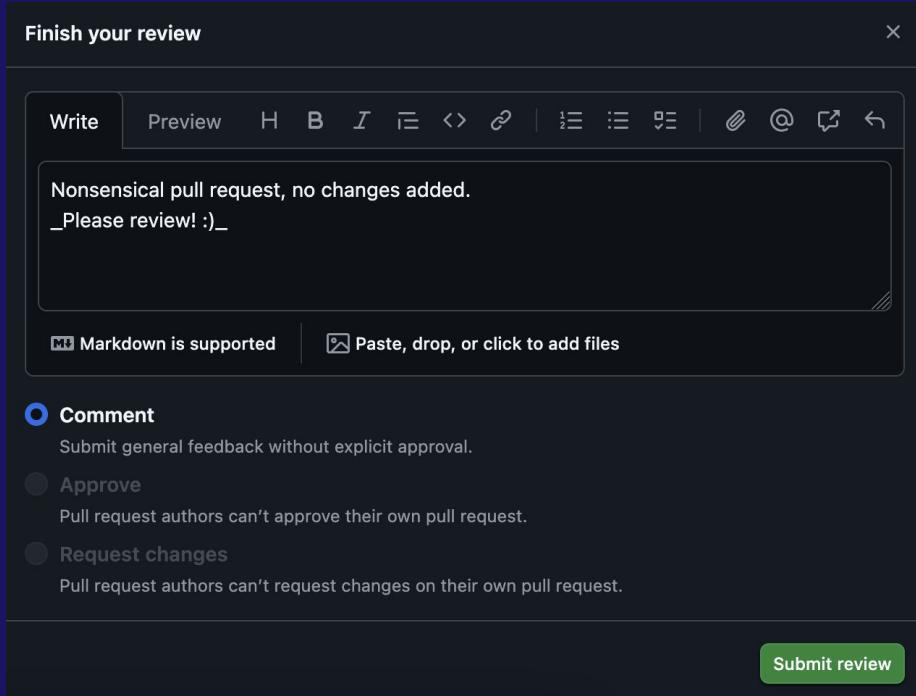


A screenshot of a GitHub Pull Request interface. At the top, there are four tabs: 'Conversation' (0), 'Commits' (1), 'Checks' (0), and 'Files changed' (1). A green arrow points from the top right towards the 'Review changes' button. Below the tabs, there are filters: 'Changes from all commits', 'File filter', 'Conversations', 'Jump to', and a dropdown. To the right are buttons for 'Ask Copilot', 'Review in codespace', and 'Review changes'. The main area shows a diff for 'src/main/java/com/krillinator/demo/PRShowcase.java'. The diff shows four additions (green) and zero deletions (red). The code added is:

```
+ package com.krillinator.demo;  
+  
+ public class PRShowcase {  
+ }
```

# Pull Request

(Can't approve my own pull request)



## Skapa en kommentar!

# Pull Request (MERGE)

 **No conflicts with base branch**  
Merging can be performed automatically.

**Merge pull request** ▾ You can also merge this with the command line. [View command line instructions.](#)

 Commit message

Extended description

This commit will be authored by 56152114+Krillinator@users.noreply.github.com.

**Confirm merge** **Cancel**

# Pull Request

(Let's accept it for now...)



**Pull request successfully merged and closed**

You're all set — the `feature/pull-request-showcase` branch can be safely deleted.

**Delete branch**

Vissa utvecklare väljer att ändå ha kvar sin 'branch' om de väljer att jobba vidare eller bugfixa tidigt..

Men det är vanligt att ta bort en gren efter att den är klar.  
När vi tar bort en gren, så konverteras commits till 'main'  
vilket gör det 'kontextuellt' svårare att hitta



# Pull Request (DONE)

<> Code    ⚡ Issues    🛡 Pull requests    ⏪ Actions    📋 Projects    🛡 Security    🔍 Insights    ⚙ Settings

Filters ▾

is:pr is:open

Labels 9

Milestones 0

New pull request

0 Open ✓ 1 Closed

Author ▾

Label ▾

Projects ▾

Milestones ▾

Reviews ▾

Assignee ▾

Sort ▾



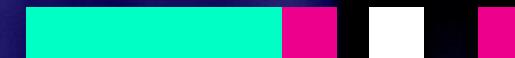
**There aren't any open pull requests.**

You could search [all of GitHub](#) or try an [advanced search](#).

*Frågor?*



# Github Projects Explained



# Github Projects?

## About Projects

---

A project is an adaptable spreadsheet, task-board, and road map that integrates with your issues and pull requests on GitHub to help you plan and track your work effectively. You can create and customize multiple views by filtering, sorting, grouping your issues and pull requests, visualize work with configurable charts, and add custom fields to track metadata specific to your team. Rather than enforcing a specific methodology, a project provides flexible features you can customize to your team's needs and processes.

To get started and create a project, see "[Creating a project](#)." To learn more about the different layouts, see "[Changing the layout of a view](#)."

# Example

The screenshot shows a digital project management tool interface with four main columns: Backlog, Doing, Review, and Done.

- Backlog:** Contains items labeled "DigitalStock\_Frontend #1" (Routing & Navbar) and "DigitalStock\_Frontend #5" (Dark Theme Implementation). A note says: "As a < type of user >, I want < some goal > so that < some reason >."
- Doing:** Contains items labeled "DigitalStock\_Frontend #9" (Signup Form) and "DigitalStock\_Frontend #8" (Login Form).
- Review:** Contains items labeled "DigitalStock\_Frontend #2" (Fetch API, Load Products).
- Done:** Contains items labeled "DigitalStock\_Frontend #3" (Tailwind Custom Color Theme) and "DigitalStock\_Frontend #4" (Landing Page).

At the bottom of each column, there is a "+ Add item" button.

Header elements include:

- @Krillinator's DigitalStock project
- Increased items preview
- Feedback
- Add status update
- Backlog
- Team capacity
- Current iteration
- Roadmap
- My items
- New view
- Filter by keyword or by field
- Discard
- Save

# Github Projects Tutorial



# Börja först med att skapa ett Repository



New repository

Overview   Repositories 142   Projects   Packages   Stars

## Welcome to projects

Built like a spreadsheet, project tables give you a live canvas to filter, sort, and group issues and pull requests. Tailor them to your needs with custom fields and saved views.

Learn more

is:open

x

New project



# Projects are linked Automatically When created from repositories

Overview    Repositories 142    Projects    Packages    Stars

## Welcome to projects

Built like a spreadsheet, project tables give you a live canvas to filter, sort, and group issues and pull requests. Tailor them to your needs with custom fields and saved views.

[Learn more](#)



is:open

New project

1 Open 0 Closed

Sort ▾

@Krillinator's DigitalStock project Private ...

#1 updated 5 minutes ago

# If you want to manually set it up...

The screenshot shows a GitHub project dashboard for the repository "Krillinator / DigitalStock\_Frontend". The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects (with 1 item), Wiki, Security, Insights, and Settings. A search bar and various icons are also present.

The main area features a "Welcome to projects" section with a description: "Built like a spreadsheet, project tables give you a live canvas to filter, sort, and group issues and pull requests. Tailor them to your needs with custom fields and saved views." Below this is a "Learn more" button and a search bar with the query "is:open".

A large, stylized image of a project board with cards and user avatars serves as a background for the dashboard.

Below the search bar are buttons for "+ New project" and "Link a project".

The project list shows 1 Open issue and 0 Closed issues. A card for "@Krillinator's DigitalStock project" (Private) is listed, with the note "#1 updated 6 minutes ago".

Two cyan arrows highlight specific elements: one points to the "Projects" tab in the top navigation bar, and another points to the "Sort" dropdown menu in the project list area.

*Frågor?*



# User-/Technical Stories



# What is a User Story?

A User Story describes **what a user wants** and **why**.  
It focuses on **value**, not implementation.

Think of it as:

*"What problem are we solving for the user?"*



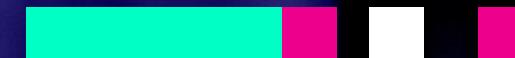
## The User Story template

As a <type of user>  
I want <something>  
So that <reason / value>

### Source:

<https://krillinator.github.io/developer-fundamentals/terminal,-git-&-github/user-&-technical-stories/user-stories-vs-technical-user-stories.html>

# Creating a User Story



Backlog

Filter by keyword or by field

○ Todo 0 / 5 Estimate: 0

This item hasn't been started

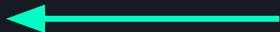
...

+ Add item

# Add Item to Backlog

# Write the User Story

⊕ Create new issue



○ Create a draft



↗ Add item from repository



As a \_\_ i want \_\_ so that \_\_|



# Draft - part #1

⊕ Create new issue



○ Create a draft



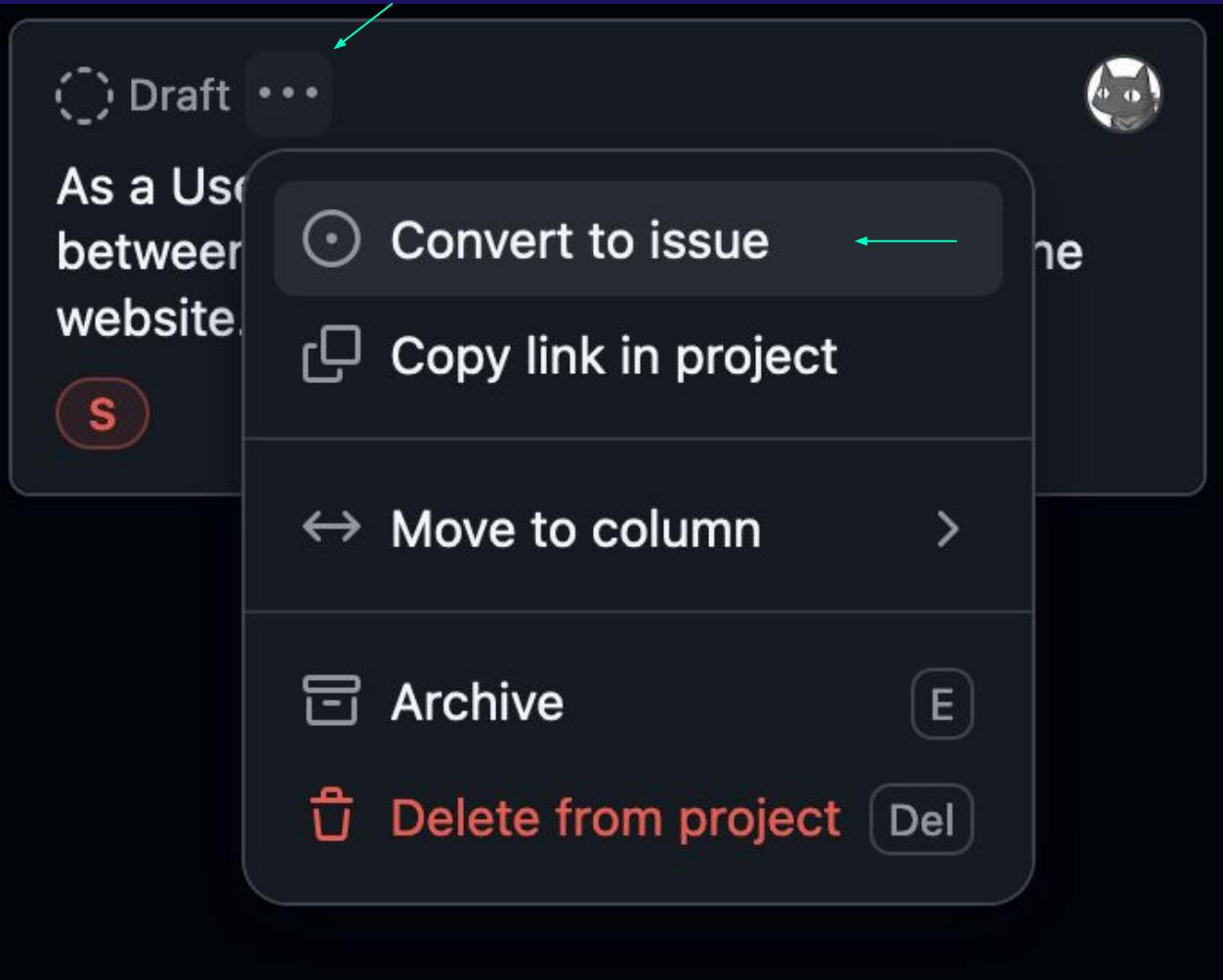
Om ni valt detta..



↗ Add item from repository



As a \_\_ i want \_\_ so that \_\_|



Draft - part  
#2

After the creation of a User Story, click its TEXT

 Draft ...

As a User, I want to be able to Navigate  
between pages, so that i can get around the  
website.

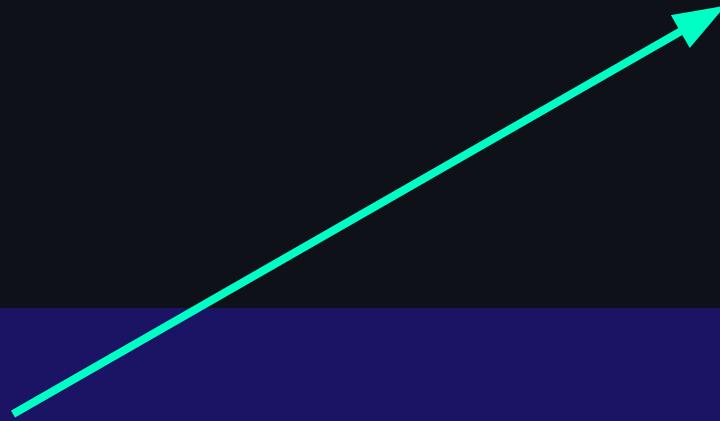


Krillinator 2 minutes ago (edited)

Edit

## Todo

- Routes
- Dynamic Routes & Params
- Documentation



**Edit to add context to the User Story  
In my case, i add a 'TODO'**

**Assignees**

Add assignees...



Add a User - ME in this case!

**Status**

Backlog

**Priority**

Choose an option...



**Size**

Choose an option...



**Estimate**

Enter a number...

**Iteration**

Choose an iteration...



**Start date**

Enter a date...

**End date**

Enter a date...

Assignees

Add assignees...

Status

Backlog

Priority

Choose an option... ▾

Size

Choose an option... ▾

Estimate

Enter a number...

Iteration

Choose an iteration... ▾

Start date

Enter a date...

End date

Enter a date...

## Priority for User Story

Select an item

Filter options

P0

P1

P2

	Urgency	Impact	Example	Response
P0	Critical	Extensive	System outage	Immediate
P1	High	Large	Major feature malfunctioning	Urgent but not out of BAU schedule
P2	Moderate	Moderate	Minor feature malfunctioning	Important but needs to be prioritized against other issues
P3	Low	Minor	Functionality or feature prevents a few users from using the product	Part of routine work
P4	Negligible	Negligible	Minor issue that doesn't affect user base	Should be placed on backlog

Source: <https://fibery.io/blog/product-management/p0-p1-p2-p3-p4/>

Assignees

Add assignees...

Status

Backlog

Priority

Choose an option...

Size

Choose an option...

Estimate

Enter a number...

Iteration

Choose an iteration...

Start date

Enter a date...

End date

Enter a date...

Estimate

Size

Estimate

Iteration

Start date

End date

Convert to issue

Copy link in pro

S

Select an item

Q Filter options

XS

S

M

L

XL

Assignees



Krillinator

Status

Backlog

Priority

Choose an option... ▾

Size

S

Estimate

Enter a number...

Iteration

Choose an iteration... ▾

Start date

Nov 5, 2024

End date

Nov 6, 2024

ALL IN ALL

**Backlog** 6 / 5 Estimate: 0

This item hasn't been started

Draft My Test User Story

Draft Item 2

Draft Item 3

## Column Limit?

### Intended Outcome

By setting and visually highlighting limits on the number of items in a column on a board, users and teams will be able to prioritize their work through the different stages of development. This will increase focus and clarity on the most important work items.

**Don't want to manually convert to issue?**

**Automate!** Avoids having to pinpoint repository for ISSUE CREATION manually

• DigitalStock\_Frontend #1



As a User, I want to be able to Navigate between pages, so that i can get around the website.

S

# About issues

Learn how you can use GitHub Issues to track ideas, feedback, tasks, or bugs.

You can create issues in your repository to plan, discuss, and track work. Issues are quick to create, flexible, and can be used in many ways. Issues can track bug reports, new features and ideas, and anything else you need to write down or discuss with your team. You can also break your work down further by adding sub-issues and easily browse the full hierarchy of work to be done.

Issues can be created in a variety of ways, so you can choose the most convenient method for your workflow. For example, you can create an issue from a repository, while adding sub-issues, convert a comment in an issue or pull request, create an issue from a specific line of code, or via a URL query. You can also create an issue from your platform of choice: through the web UI, GitHub Desktop, GitHub CLI, GraphQL and REST APIs, or GitHub Mobile. See "[Creating an issue.](#)"

Source: <https://docs.github.com/en/issues/tracking-your-work-with-issues/about-issues>

# As a User, I want to be able to Navigate between pages, so that i can get around the website. #1

*Observera: historik finns nu inom vår User Story*



Krillinator opened 2 minutes ago ...

**Todo**

- Routes
- Dynamic Routes & Params
- Documentation

Smiley icon

Krillinator added this to [@Krillinator's DigitalStock project](#) 19 minutes ago

User icon

Krillinator self-assigned this 2 minutes ago

User icon

Krillinator converted this from a draft issue 2 minutes ago

User icon

Assign

K

Labels

No lab

Project

Stat

Prior

Size

Estim

Itera

Star



## Add a comment

Write

Preview

H B I | E <> C | E 1E 2E | @ ↗ ← ↘

\*\*This idea would work extremely well\*\* with Dynamic parameters for a [Slug] pattern especially [...Slug]

📎 Paste, drop, or click to add files

Close with comment

Comment

*Inom din user story, prova skapa en kommentar inom ditt '**ISSUE**'*



Krillinator opened 2 minutes ago

...

## Todo

- Routes
- Dynamic Routes & Params
- Documentation



Krillinator added this to [@Krillinator's DigitalStock project](#) 19 minutes ago



Krillinator self-assigned this 2 minutes ago



Krillinator converted this from a draft issue 2 minutes ago



Krillinator now

Owner

Author

...

This idea would work extremely well with Dynamic parameters for a [Slug] pattern especially [...Slug]



1





Write

Preview

H B I | ↵ ↶ ↷ | ⚡ ⚡ ⚡ | @ ↗ ↙ ↖ ↘ | ☰

### ### Todo

- Routes
- Dynamic Routes & Params
- Documentation

Documentation: [Routing](<https://nextjs.org/docs/app/building-your-application/routing>)

📎 Paste, drop, or click to add files

Cancel

Save

**HINT:** Add Documentation Links for cleaner TODO's.  
*Helps developers track source information*

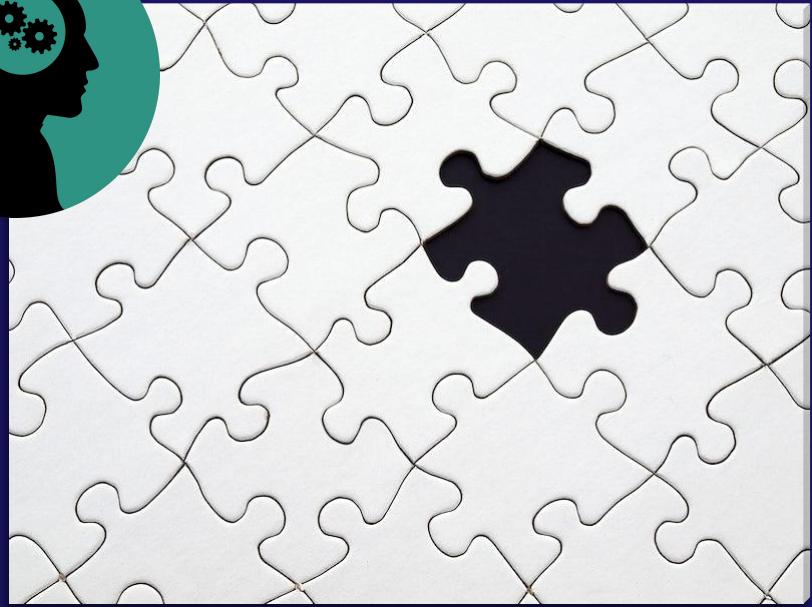


Krillinator / Projects / @Krillinator's DigitalStock project ⚙

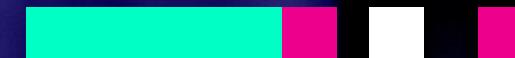
## 🌐 @Krillinator's DigitalStock project

Ändra från default PRIVATE till Public genom att trycka här

*Frågor?*

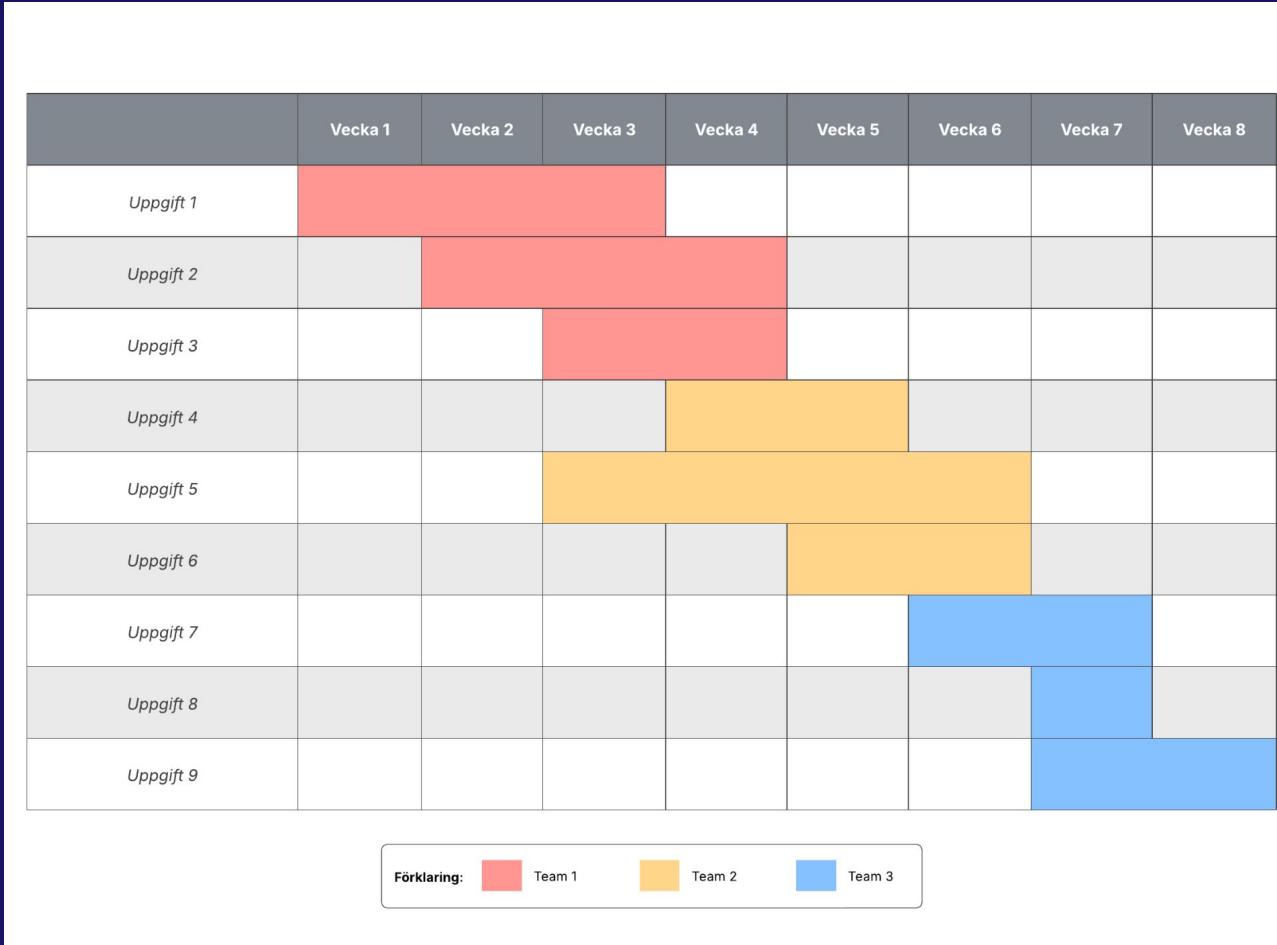


# Gannt Schema Roadmap



Ett **Gantt-schema**, engelska Gantt chart, är en typ av **flödesschema** som används inom **projektledning** för att beskriva **olika faser**, **tidsaspekter** och **beroenden** mellan dessa inom ett projekt.

*Schemat är döpt efter dess upphovsman, Henry Gantt, som konstruerade schemat någon gång mellan 1910 och 1915.*



[Backlog](#)[Team capacity](#)[Current iteration](#)[Roadmap](#)[My items](#)[New view](#) Filter by keyword or by field[Discard](#)[Save](#)

April 2025

13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12

1 ● Spring Security override #1

May 2025

[Markers](#)[Sort](#)[Date fields](#)[Month](#)[Today](#)[+](#) Add item

Notera att alla '**Issues**' & '**User Stories**' kopplas automatiskt till '**Roadmap**'

December 2024

Markers

Sort

Date fields

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Spring Security override #1



Filtrera utifrån behov, vare sig det är en '**Sprint**' AKA Iteration start/end  
eller **specifikt datum**.

Start date

✓ Iteration start

Start date

End date

No start date

Target date

✓ Iteration end

Start date

End date

No target date

October 2024					November 2024											Actions													
24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	● Signup Form #9																								● Signup Form #9				
2	● Login Form #8							+																					
3	● Fetch API, Load Products #2																												
4	● Tailwind Custom Color Theme #3																												
5	● Landing Page #4							+																					
6	● As a < type of user >, I want < some goal > s...							+																					
7	● Routing & Navbar #1							+																					
8	● Dark Theme Implementation #5							+																					
9	● Custom Font Implementation #6							+																					
10	● Button Component #7							+																					

Detta blir relevant inom större projekt, där planering och fördelning av arbete är av extra stor vikt

*Frågor?*



# 03

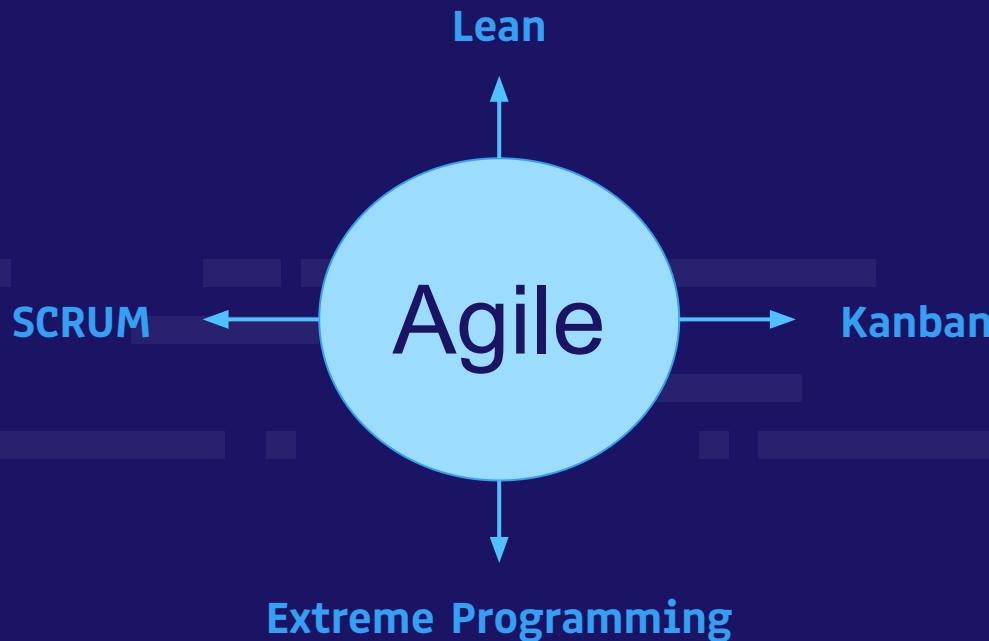
## Bonus

(bonus)

# Extreme Programming

## What is it?

# Extreme Programming (Methodology/Framework)

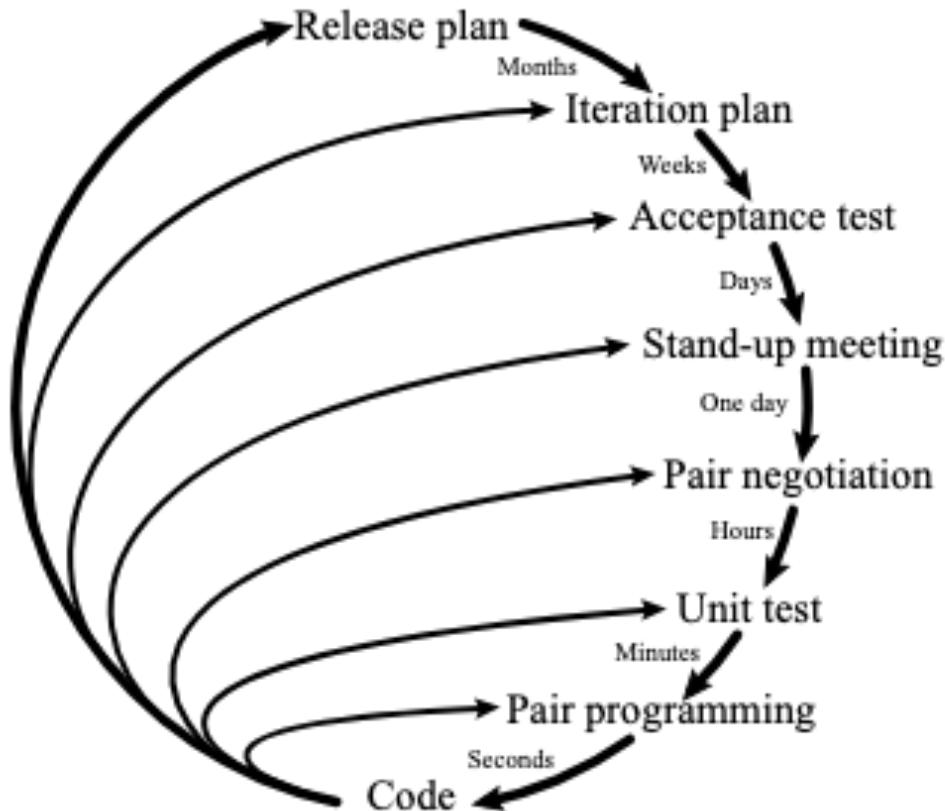


# Extreme Programming (Principles)

## Extreme Programming key points:

- Pair Programming
- Continuous Integration, Continuous Delivery
- Test-driven Development
- Shared Codebase
- Code Review
- Customer Involvement
- Simple Design

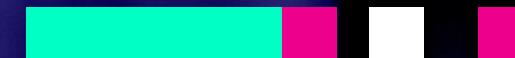
# Planning/feedback loops



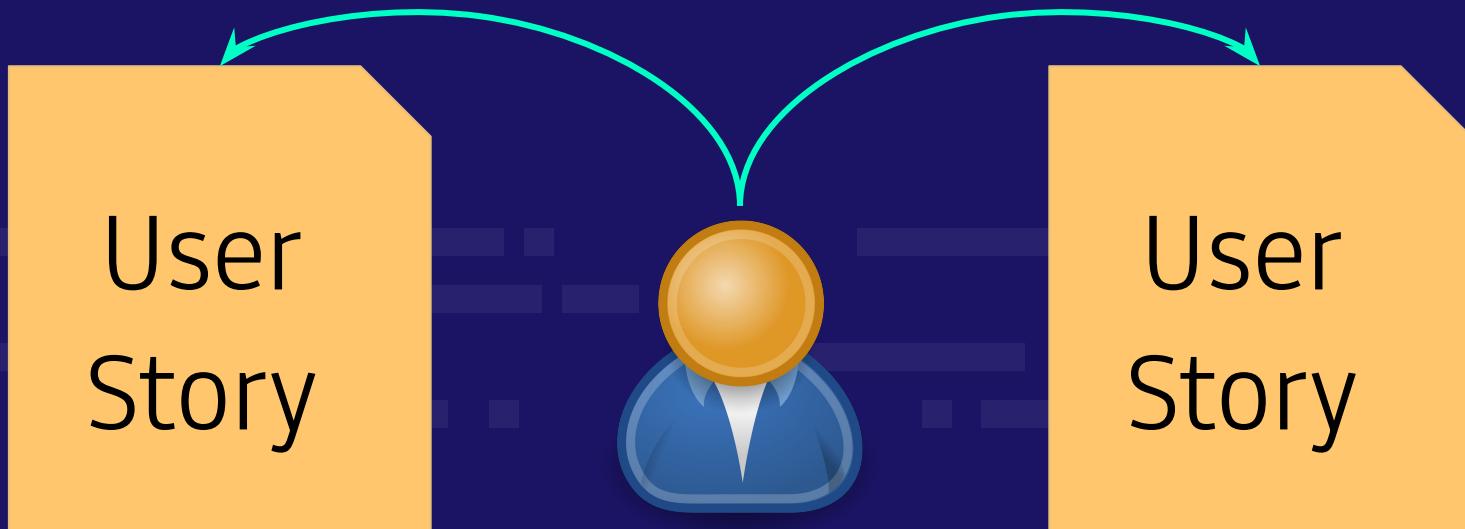
**Notera:** Agil utveckling är iterativ, vi ser här att även 'Extreme Programming' tillför ett liknande iterativt flöde.

**Observera:** Extreme Programming är mer orienterat kring utveckling.

# Code Review & Shared Codebase (bonus)



# Shared Codebase (Without Extreme Programming)



**Benny** tar hand om 2  
'User Story Features'

# Shared Codebase (With Extreme Programming)

User  
Story



User  
Story

**Teamet** tar hand om  
ALLA 'user stories'

# Code Review (Close Team)

“Hej, kan du kolla  
på min kod här?”

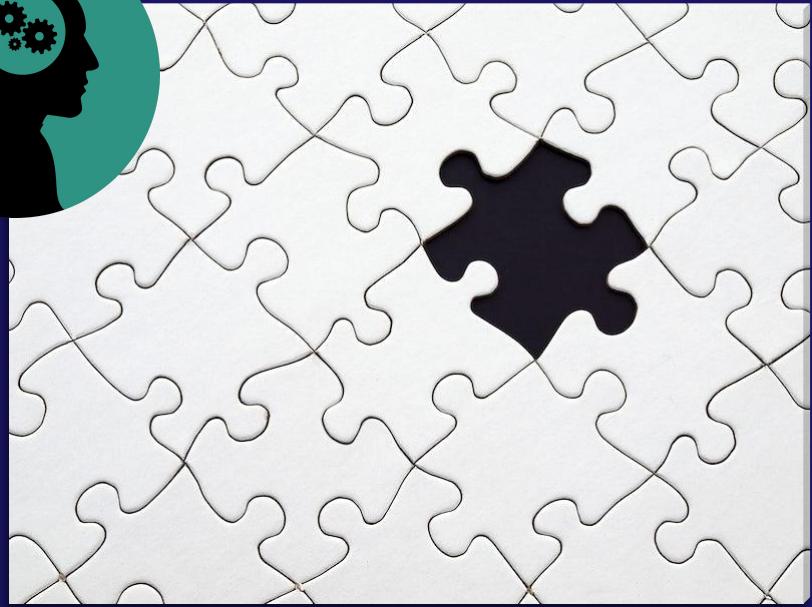


“Absolut, det är ‘vår’  
kod trots allt” –



**Notera:** Bättre helhetsförståelse för kodbas i sin helhet. Enklare att bidra. Högre kvalité på kod. Upptäcker buggar tidigt i livslängden av projekt.

*Frågor?*



# Protecting Git Branches (bonus) Tutorial





Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

General

Access

Collaborators

Moderation options

Code and automation

→ Branches

Tags

Rules

Actions

## Branch protection rules



### Classic branch protections have not been configured

Define branch rules to disable force pushing, prevent branches from being deleted, or require pull requests before merging. Learn more about [repository rules](#) and [protected branches](#).

[Add branch ruleset](#)[Add classic branch protection rule](#)

**Ruleset Name \***

main - require PR before merge

**Enforcement status**



Active ▾

Du kan stänga av regeln när som helst med 'enforcement status'

## Bypass list

+ Add bypass ▾

Exempt roles, teams, and apps from this ruleset by adding them to the bypass list.

**Bypass list is empty**

Någon i Teamet som vi inte behöver tillämpa sagd regel till?

**Lämnar detta tomt**

## Target branches

Which branches should be matched?

Branch targeting criteria

Add target ▾



Default



Default == 'main' or 'master'

### **Require a pull request before merging**

Require all commits be made to a non-target branch and submitted via a pull request before they can be merged.

[Hide additional settings ^](#)

### **Required approvals**

1 ▾

The number of approving reviews that are required before a pull request can be merged.

#### **Dismiss stale pull request approvals when new commits are pushed**

New, reviewable commits pushed will dismiss previous pull request review approvals.

#### **Require review from Code Owners**

Require an approving review in pull requests that modify files that have a designated code owner.

#### **Require approval of the most recent reviewable push**

Whether the most recent reviewable push must be approved by someone other than the person who pushed it.

#### **Require conversation resolution before merging**

All conversations on code must be resolved before a pull request can be merged.

#### **Automatically request Copilot code review**

Request Copilot code review for new pull requests automatically if the author has access to Copilot code review.

### **Allowed merge methods**

Merge, Squash ▾

When merging pull requests, you can allow any combination of merge commits, squashing, or rebasing. At least one option must be enabled.

```
kristofferjohansson@Kristoffers-MacBook-Pro demo_5 % git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 549 bytes | 549.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote: error: GH013: Repository rule violations found for refs/heads/main.
remote: Review all repository rules at https://github.com/Krillinator/EE\_Lektion\_5\_Collab/rules?ref=refs%2Fheads%2Fmain
remote:
remote: - Changes must be made through a pull request.
remote:
To https://github.com/Krillinator/EE\_Lektion\_5\_Collab.git
 ! [remote rejected] main -> main (push declined due to repository rule violations)
error: failed to push some refs to 'https://github.com/Krillinator/EE\_Lektion\_5\_Collab.git'
```

Can't push to main directly, must go through PR first!



Krillinator wants to merge 1 commit into `main` from `file/readme`

Conversation 0

-o Commits 1

Checks 0

Files changed 1



Krillinator commented now

Owner ...

Added instructions for forking and completing the project.



-o Add README with project instructions ...

Verified

4b0759d



Krillinator commented now

Owner Author ...

Verification required



**Review required**

At least 1 approving review is required by reviewers with write access.



**Merging is blocked**

New changes require approval from someone other than the last pusher.

Code and automation

Branches

Tags

Rules

Rulesets

Actions

Active ▾

## Bypass list

+ Add bypass ▾

Exempt roles, teams, and apps from this ruleset by adding them to the bypass list.

Repository admin Role

Always allow ...

**Setup:** Bypass for Admin



## Review required

At least 1 approving review is required by reviewers with write access.



## Merging is blocked

New changes require approval from someone other than the last pusher.



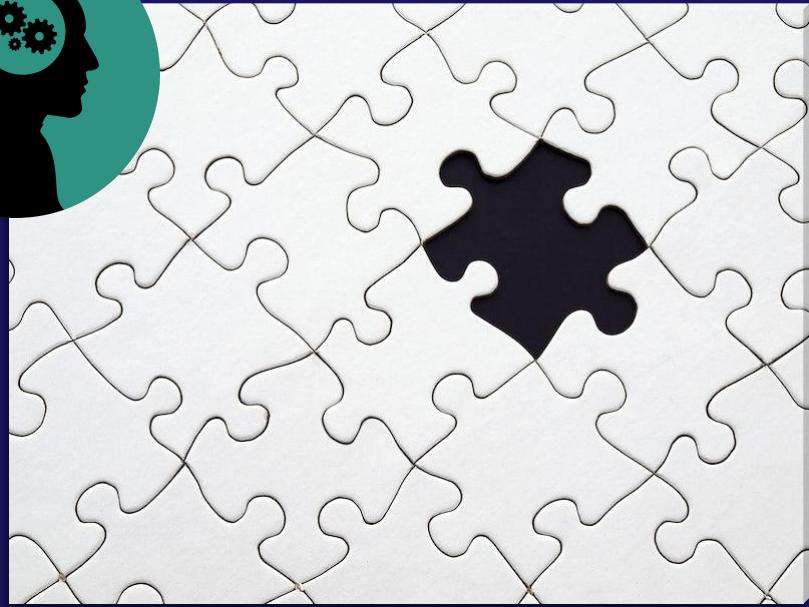
Merge without waiting for requirements to be met (bypass rules)

Merge pull request



You can also merge this with the command line. [View command line instructions](#).

*Frågor?*



# Forking (bonus) Github Repo'

## About forks

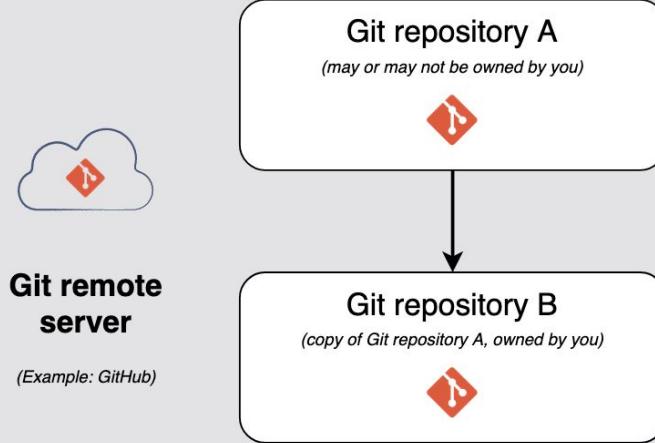
---

A fork is a new repository that shares code and visibility settings with the original “upstream” repository. Forks are often used to iterate on ideas or changes before they are proposed back to the upstream repository, such as in open source projects or when a user does not have write access to the upstream repository. For more information, see [Working with forks](#).

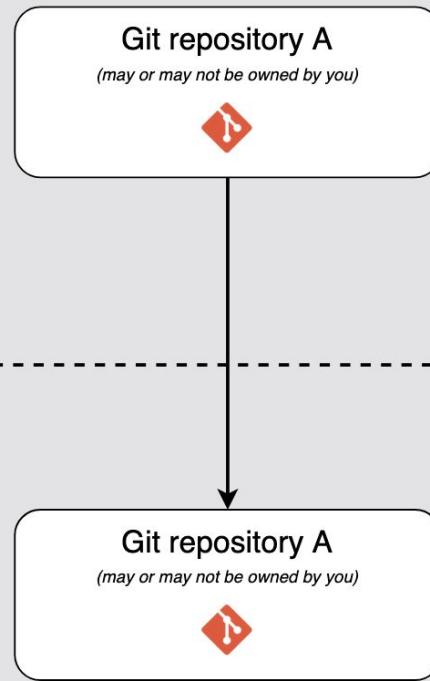
Source:

<https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/working-with-forks/fork-a-repo#about-forks>

## Forking



## Cloning



**FORK** = New Feature,  
new repository

**CLONE** = Collab, local  
project & files

# 04

## Uppgifter

&

## Eget Arbete

```
1          // -Uppgift #1- //
2          Pair Programming
3
4  /* INSTRUCTIONS
5
6      Uppgift
7
8          • Du får en lista
9          • Ett tal kommer två gånger
10         • Skriv ut det första talet som upprepas
11
12     numbers = [1, 3, 5, 3, 7, 5]
13
14     Output: 3
15
16     */
17
18     // HINT & Examples
19     hint("Navigator: vägleder, tänker högt, säger inte
20         exakt kod")
21     hint("Driver: kodar, skriver bara det som sägs,
22         läser koden högt, ändrar inte logiken själv")
23
```



## Uppgift #1

Kom igång enkelt med uppgift #1

```
1          // -Uppgift #2- //
2          Pair Programming
3
4  /* INSTRUCTIONS
5
6      Uppgift
7
8          • Du får en sträng
9          • Räkna hur många bokstäver den har
10         • Skriv ut antalet
11
12     text = "python"
13
14     Output: 6
15
16 */
17
18 // HINT & Examples
19 hint("Navigator: vägleder, tänker högt, säger inte
20       exakt kod")
21 hint("Driver: kodar, skriver bara det som sägs,
22       läser koden högt, ändrar inte logiken själv")
23
```



## Uppgift #2

```
1           // -Uppgift #3- //
2           Versionshantering
3
4 /* INSTRUCTIONS
5
6     Skapa ett nytt Repository
7
8     VIKTIGT: lägg till en README fil direkt i github (pusha inte från
9     terminalen)
10
11
12     Skapa ett nytt lokalt projekt. Lägg till remote. Skapa nu en ny fil
13     lokalt. Försök pusha förändringarna online - lös problemet!
14
15 */
16
17 // HINT & Examples
18 hint("Detta är ett typiskt versionshanterings
19 problem, läs mer här:
20 https://krillinator.github.io/developer-fundamentals/terminal,-git-&-github/git-&-github-basics/git-&-github-common-issues.html ")
```

## Uppgift #3

```
1          // -Uppgift #4- //
2          Versionshantering
3
4      /* INSTRUCTIONS
5
6          Inom det lokala projekt:
7
8              Skapa en helt egen Readme fil.
9
10             Lägg till valfri text och pusha.. Försök nu på 'merge konflikt'!
11
12
13         // HINT & Examples
14         hint("https://docs.github.com/en/pull-requests/colla
15         borating-with-pull-requests/addressing-merge-conflic
16         ts/resolving-a-merge-conflict-using-the-command-line
17         ")
18
19
20
21
22
23
```

## Uppgift #4

# THANKS !

Do you have any questions?  
[kristoffer.johansson@sti.se](mailto:kristoffer.johansson@sti.se)

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik.