

324-06A Branching Strategien

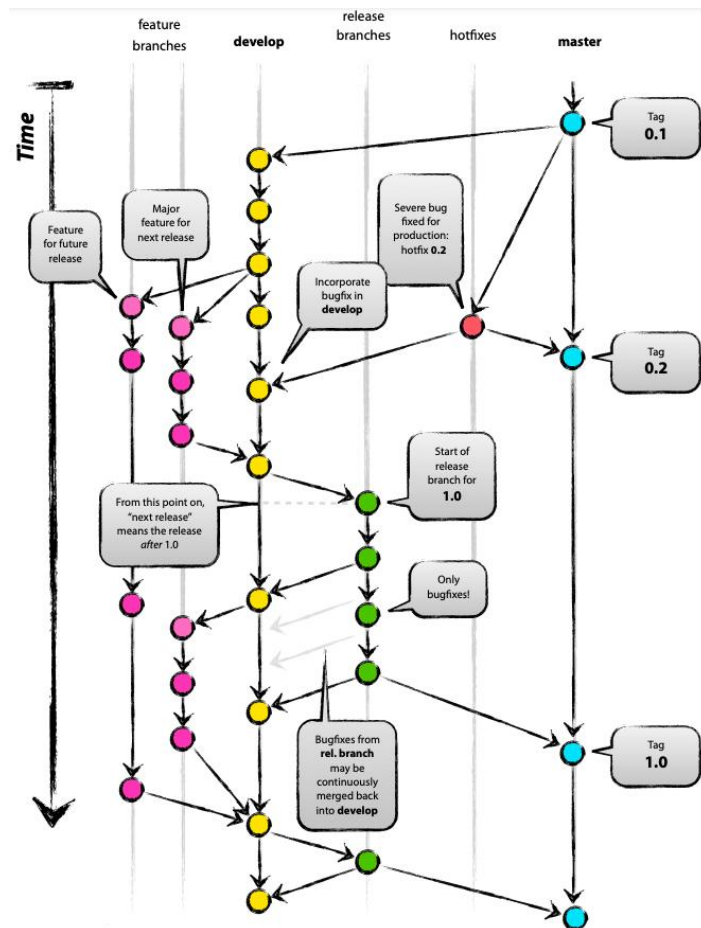
Inhalt

Branching-Strategien Analyse und Auswahl.....	2
Recherche verschiedener Branching-Strategien	2
GitFlow	2
GitHub Flow	3
GitLab Flow.....	3
Trunk-based Development	4
Vergleichstabelle	4
Abwägungen.....	4
Unsere Entscheidung: GitHub Flow	5
Branching-Strategie umsetzen.....	6
Workflow-Regeln	6
Workflow-Schritte festlegen.....	6
Änderungen der jeweiligen Teammitglieder	9
Kristian Lubina (Aufgaben-Zähler hinzufügen)	9
Daniel Kovac (Filter-Buttons hinzufügen)	11
Branch-Graph	12

Branching-Strategien Analyse und Auswahl

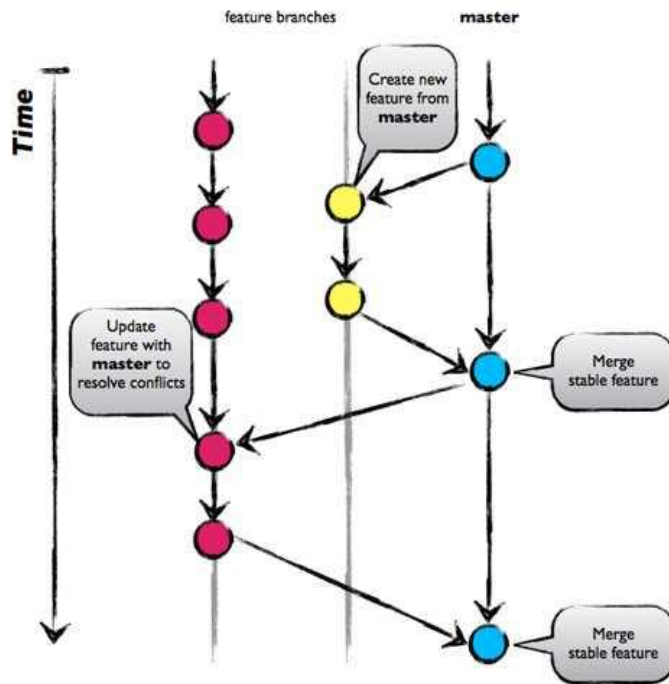
Recherche verschiedener Branching-Strategien

Wir haben gemeinsam vier verschiedene Branching-Strategien analysiert:



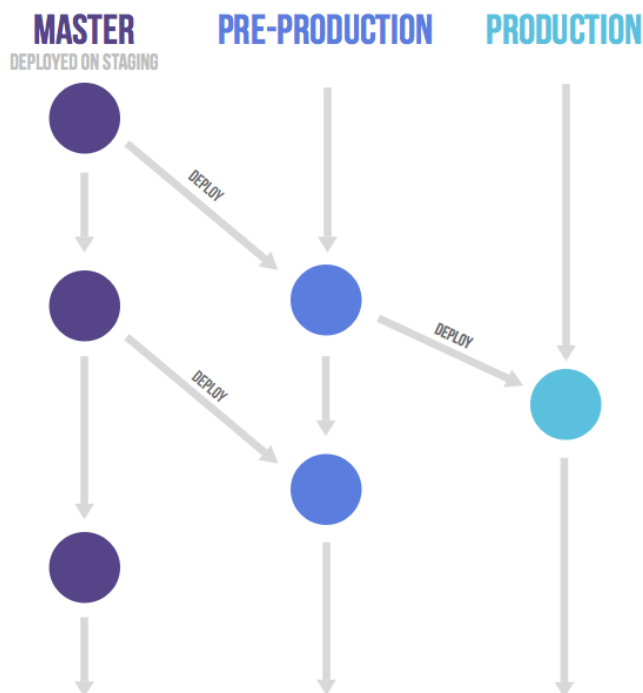
GitFlow

- **Struktur:** Master, Develop, Feature, Release, Hotfix Branches
- **Vorteile:** Parallele Entwicklung, klare Organisation, mehrere Versionen möglich
- **Nachteile:** Komplex, langsame Release-Zyklen, nicht CI/CD-freundlich



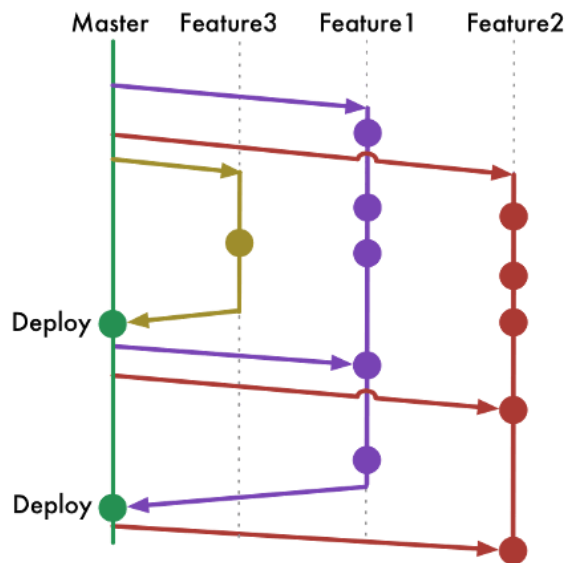
GitHub Flow

- **Struktur:** Main Branch + Feature Branches
- **Vorteile:** Einfach, schnelle Releases, CI/CD-freundlich
- **Nachteile:** Keine Multiple-Version-Verwaltung, anfällig für Bugs



GitLab Flow

- **Struktur:** Main + Production + Environment Branches
- **Vorteile:** Umgebungs-Isolation, kontrollierte Releases
- **Nachteile:** Komplexer als GitHub Flow, längere Release-Zyklen



Trunk-based Development

- **Struktur:** Ein einziger Trunk/Main Branch
- **Vorteile:** Sehr einfach, beste CI/CD-Unterstützung, keine Merge-Konflikte
- **Nachteile:** Erfordert erfahrene Entwickler, hohe Disziplin nötig

Vergleichstabelle

Kriterium	GitFlow	GitHub Flow	GitLab Flow	Trunk-based
Komplexität	Hoch	Niedrig	Mittel	Sehr niedrig
Teamgröße	Groß	Klein	Mittel	Klein-Mittel
CI/CD Eignung	Schlecht	Gut	Gut	Ausgezeichnet
Release-Zyklen	Langsam	Schnell	Mittel	Sehr schnell
Lernkurve	Steil	Flach	Mittel	Flach

Abwägungen

Wir haben folgende Faktoren berücksichtigt:

- **Projekttyp:** Web-Anwendung mit kontinuierlicher Bereitstellung
- **Teamgröße:** Kleines Team im Bildungskontext
- **Erfahrungslevel:** Lernende mit begrenzter Git-Erfahrung
- **CI/CD:** Bestehende GitLab Pipeline soll genutzt werden

Unsere Entscheidung: GitHub Flow

Begründung:

1. **Projektpassung:** Unsere ToDo-App ist eine einfache Web-Anwendung ohne Multiple-Version-Anforderungen
2. **Team-Kontext:** Als lernendes Team profitieren wir von der einfachen Struktur und flachen Lernkurve
3. **CI/CD-Kompatibilität:** Unsere bestehende GitLab Pipeline (only: merge_requests) funktioniert perfekt mit GitHub Flow
4. **Praktische Vorteile:**
 - Main-Branch bleibt immer deploybar
 - Feature Branches für User Stories
 - Schnelle Integration neuer Features
 - Automatisierte Builds bei Merge Requests

GitHub Flow bietet uns die optimale Balance zwischen Einfachheit und professioneller Entwicklungspraxis für unser Bildungsprojekt.

Branching-Strategie umsetzen

Workflow-Regeln

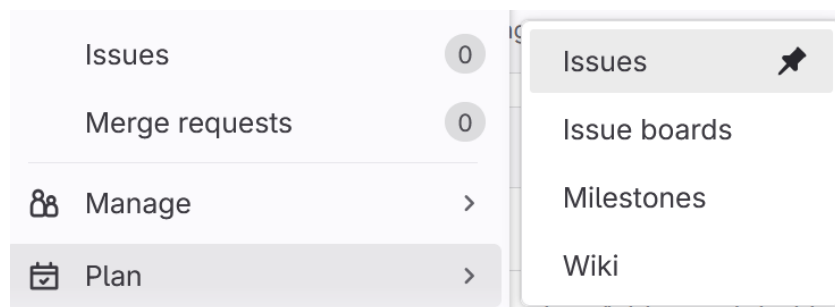
- Jeder Feature-Branch muss von einem Issue stammen
- Pipeline muss erfolgreich durchlaufen vor Merge

Workflow-Schritte festlegen

Wir haben folgende Workflow-Schritte für unser Projekt festgelegt:

Schritt 1: Issue eröffnen

- Neues Issue im GitLab Webbrowser erstellen



- Klare Beschreibung der geplanten Änderung/Feature

me / M324_PROJEKT_TODOLIST-Gemeinsam2 / Issues / New

Type
Issue

Title (required)
Add task counter display

Description
Add [description templates](#) to help your contributors communicate effectively!

Preview | **B** *I*

Add a simple counter that shows:

- Total tasks
- Completed tasks
- Open tasks

Display above the input form for quick overview.

Switch to rich text editing

Assignee Edit
 Kristian Lubina

Labels Edit
Implementierung

Parent Edit
None

Milestone Edit
None

Schritt 2: Branch erstellen

- Direkt aus dem Issue heraus: **"Create Branch"** klicken

me / M324_PROJEKT_TODOLIST-Gemeinsam2 / Issues / #9

Add task counter display

Open Issue created just now by Kristian Lubina

Add a simple counter that shows:

- Total tasks
- Completed tasks
- Open tasks

Display above the input form for quick overview.

0 0 0

Add design Create merge request

Child items 0

No child items are currently assigned. Use child items to break down work into smaller parts.

Assignee: Kristian Lubina

Labels: Implementierung

Parent: None

Milestone: None

Dates: Start: None

- Automatische Verknüpfung zwischen Issue und Branch entsteht

Schritt 3: Entwicklung

- In der erstellten Feature-Branch alle Änderungen vornehmen
- Commits mit aussagekräftigen Nachrichten

Schritt 4: Merge Request

- Feature-Branch über GitLab Webbrowser in die **main-Branch** mergen

2 All 10

Bulk edit New merge request

Search or go to...

Project

- Merge requests 0
- Manage >
- Plan >
- Code >
- Merge requests 0
- Repository
- Branches

There are no open merge requests

New merge request

New merge request

Source branch

me7554020/m324_projekt_to... 9-add-task-counter-displ...

add task counter display
Kristian Lubina authored May 27, 2025

e0b707cb

Target branch

me7554020/m324_projekt_to... main

undo main-commit
Kristian Lubina authored May 27, 2025

11bc8db9

Compare branches and continue

- Automatische Pipeline-Ausführung durch bestehende CI/CD

The screenshot shows a GitHub pull request titled "add task counter display" by Kristian Lubina, requesting a merge into the main branch. The interface includes tabs for Overview, Commits, Pipelines, and Changes. The Overview tab is active, showing a "Merge request pipeline #1838313366 passed" status. Below this, there is an "Approve" button and a "Ready to merge!" section with checkboxes for "Delete source branch", "Squash commits", and "Edit commit message". The right sidebar contains sections for Assignees, Reviewers, Labels (with a label "Implementierung"), Milestone, Time tracking, and a list of participants.

Schritt 5: Automatischer Abschluss

- Issue schließt sich automatisch nach erfolgreichem Merge
- Feature gilt als erfolgreich implementiert

Änderungen der jeweiligen Teammitglieder

Kristian Lubina (Aufgaben-Zähler hinzufügen)

Zuerst einmal wurde ein Issue eröffnet in dem die Erstellung des «Task Counter Display» jederzeit zurück verfolgt werden kann.

Add task counter display

Closed Issue created 1 hour ago by Kristian Lubina

Add a simple counter that shows:

- Total tasks
- Completed tasks
- Open tasks

Display above the input form for quick overview.

0 0

Add design

Child items 0

Add ▾ ⋮ ^

No child items are currently assigned. Use child items to break down work into smaller parts.

Linked items 0

Add ⋮ ^

Link items together to show that they're related.

Assignee

Kristian Lubina

Edit

Labels

Implementierung x

Edit

Parent

None

Edit

Milestone

None

Edit

Dates

Start: None

Due: None

Edit

Time tracking

+
Add an estimate or time spent

Nach dem die Änderungen im Code durchgeführt wurden wurde ein Merge-Request eröffnet. Dort wurde zunächst einmal überprüft ob die Codeänderungen gut sind und ob das Projekt noch build-bar ist. Diese Prüfung hat es bestanden und somit konnte die feature-Branch ins main-branch gemerged werden.

me / M324_PROJEKT_TODOLIST-Gemeinsam2 / Merge requests / !11

add task counter display

Merged Kristian Lubina requested to merge 9-add-task-counter-display into main 1 hour ago

Overview 0 Commits 1 Pipelines 1 Changes 1

Closes #9 (closed)

0 0



Merge request pipeline #1838313366 passed



Merge request pipeline passed for e0b707cb 1 hour ago

8✓

Approval is optional

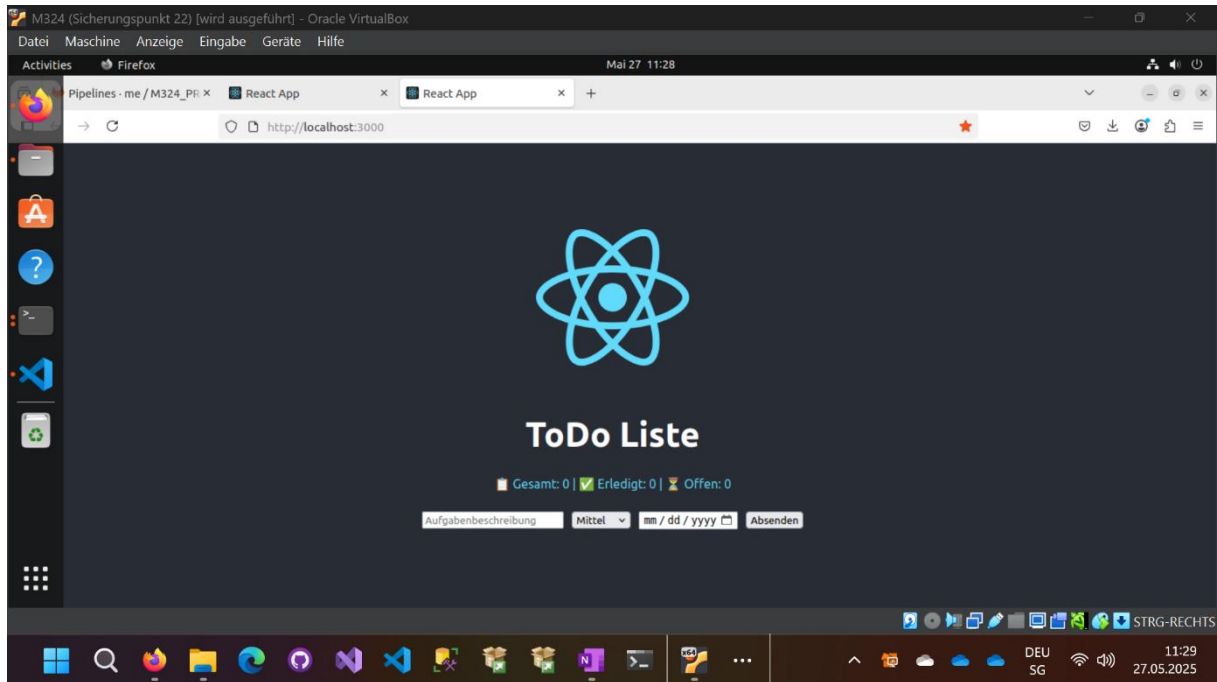


Merged by Kristian Lubina 1 hour ago

Revert

Cherry-pick

Der Aufgabenzähler wurde durch eine `getTaskCounts()` Funktion implementiert in der `App.js`-Klasse, die das `todos` Array mit `filter()` durchläuft und die erledigten/offenen Aufgaben zählt, wobei die Ergebnisse dann in einem `<div>` über dem Eingabeformular angezeigt werden.



Daniel Kovac (Filter-Buttons hinzufügen)

Zunächst einmal wurde das Issue für das Hinzufügen der Filter-Buttons hinzugefügt.

Aufgaben filter

🔒 Closed 📅 Issue created 1 hour ago by Daniel

- Ich möchte, dass Aufgaben gefiltert werden können.

Edited 28 minutes ago by Daniel

👍 0 💡 0 😊 📎 Add design

Child items 0 Add ⋮ ^
No child items are currently assigned. Use child items to break down work into smaller parts.

Linked items 0 Add ⋮ ^
Link items together to show that they're related.

Assignee Edit
Daniel

Labels Edit
Implementierung ✕

Parent Edit
None

Milestone Edit
None

Dates Edit
Start: None
Due: None

Nachdem die dementsprechende Änderungen im Code gemacht wurden, wurde dann das Feature-Branch in das main-Branch gemerged.

me / M324_PROJEKT_TODOLIST-Gemeinsam2 / Merge requests / #12

Filter hinzugefügt für Aufgaben.

🔗 Merged Daniel requested to merge `Aufgaben-filter` into `main` 25 minutes ago

Overview 0 **Commits** 1 **Pipelines** 1 **Changes** 1 Add a to-do item

Filter hinzufügen für die Aufgaben.

👍 0 💡 0 😊

✓ **Merge request pipeline #1841581575 passed** ✓ 📄
Merge request pipeline passed for `ff5a1cfa` 24 minutes ago ?

8✓ Approval is optional ⌵

🔗 Merged by Daniel 24 minutes ago Revert Cherry-pick

Assignee Edit
Daniel

0 Reviewers Edit
None - assign yourself

Labels Edit
None

Milestone Edit
None

Die Filter-Buttons wurden durch einen neuen «filter»-State, eine «handleFilterChange()» Funktion zum Setzen des aktiven Filters und eine «getFilteredTodos()» Funktion implementiert, die das «todos» Array basierend auf dem aktuellen Filter (alle, offen, erledigt) filtert und nur die entsprechenden Aufgaben für die Anzeige zurückgibt.



Branch-Graph

Die Visualisierung des der Impelemntierung nach unserem Workflow sieht nun wie folgt aus.

