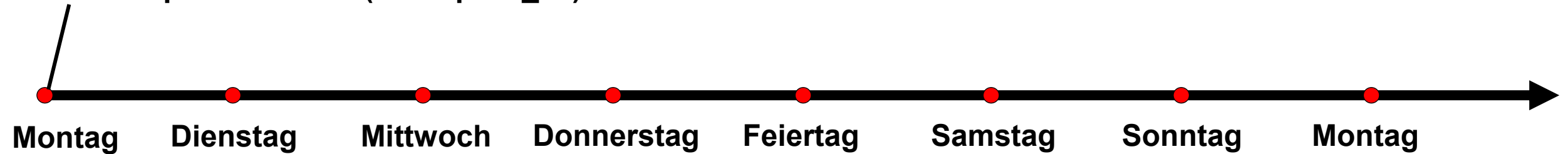


Sprintplanung

1 Schritt (ca. Montag, 10 Uhr):

Es wird ein Branch vom Main erstellt und nach dem Sprint benannt (z.B. Sprint_01)



Sprintplanung

2 Schritt:

Jede Änderung bedarf ein Branch vom Sprint Branch, wobei der Branch wie folgt benannt werden sollte:

[Feature/Bug] <Issue ID: Issue Name>



2 Alternativen



```
graph TD; A[2 Alternativen] --> B[Entwicklungsphase: Montag – Freitag<br/>Testtag: Samstag<br/>Tester: 1 Person]; A --> C[Entwicklungsphase: Montag – Sonntag<br/>Testtag: Montag<br/>Tester: Mehrere Personen];
```

Entwicklungsphase: Montag – Freitag
Testtag: Samstag
Tester: 1 Person

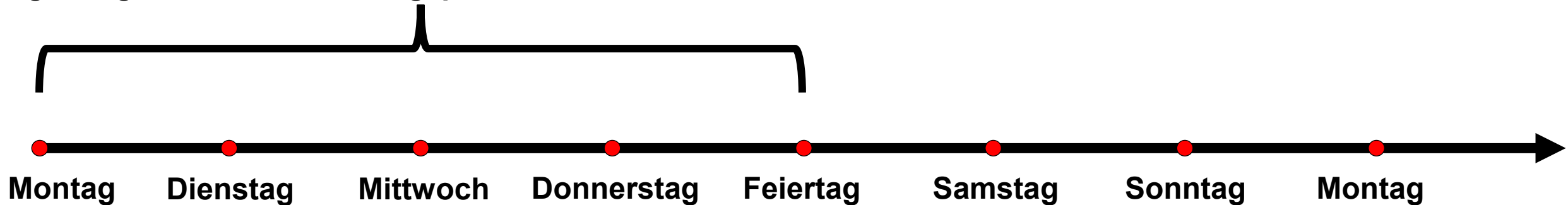
Entwicklungsphase: Montag – Sonntag
Testtag: Montag
Tester: Mehrere Personen

Sprintplanung – Alternative 1

3 Schritt:

Entwicklungsphase von Montag bis Freitag

Alle Pull Requests müssen bis Freitag (23:59 Uhr) erstellt und bearbeitet worden, und auf den Sprint Branch gemergt sein, um auf Main gepusht zu werden



Sprintplanung – Alternative 1

4 Schritt:

Testtag – Eine Person testet alle Szenarien. Test gilt als erfolgreich, wenn alle Szenarien mindestens genau so gut funktionieren, wie in dem vorherigen Sprint.

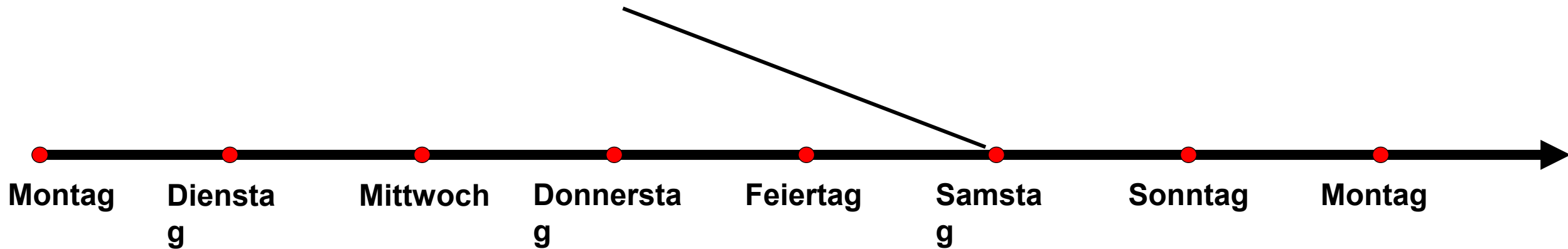
Zuständige Person rotiert unter allen Teilnehmern, bei 6 Personen, muss jeder im Laufe des Projekts 2 - 3 mal testen



Sprintplanung – Alternative 1

5 Schritt:

Testtag schreibt kurzes Protokoll (Checkliste in Github), welche Szenarien fehlschlugen und vergleicht mit dem Protokoll der Vorwoche. Wenn Test erfolgreich ist erstellt der Tester ein Pull Request (Sprintbranch → Main)



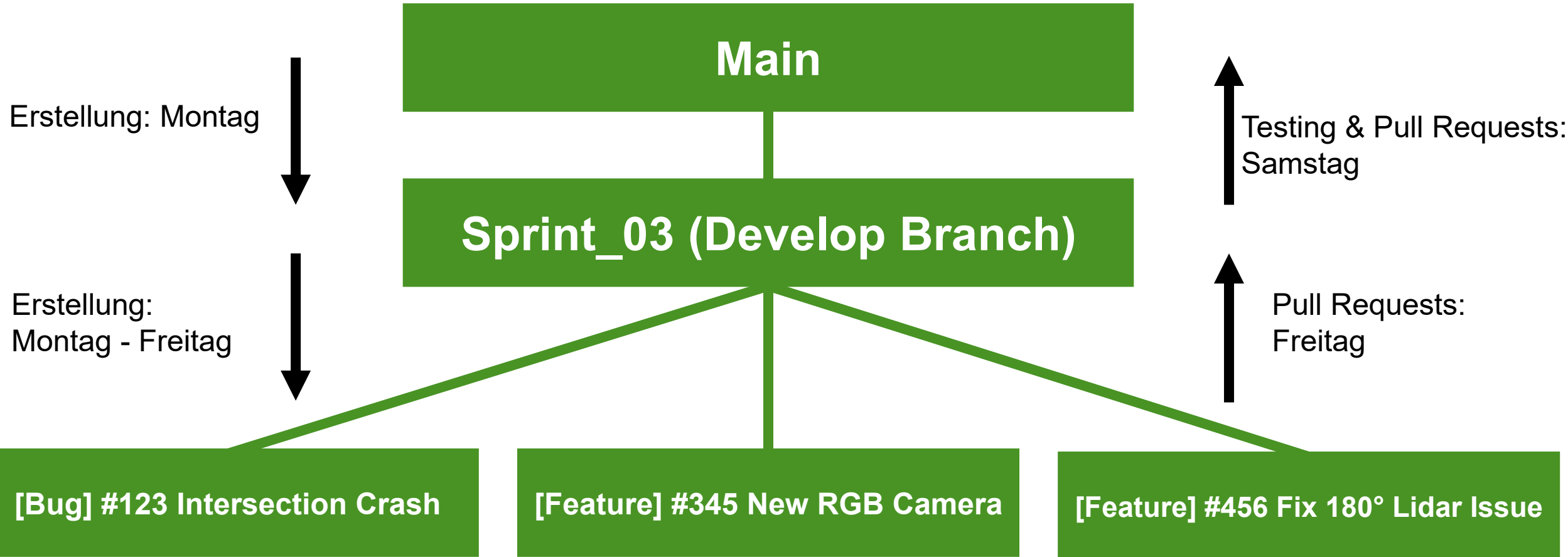
Sprintplanung – Alternative 1

6 Schritt (Montag ca. 10 Uhr):

Sprint Branch wird gemerged und neuer Branch wird erstellt



Überblick Git Branches



Vorteile

- 1 Klare Strukturen & geregelter Ablauf ermöglicht einfache Planung
- 2 Sichere Tests & geregelter Rollback bei Bugs
- 3 Erkennung von Dependency Bugs (Feature A hat Auswirkung auf Feature B im selben Sprint)
- 4 Verbessert Zusammenspiel zwischen Codeänderungen und Github Backlog/ Dokumentation

Nachteile

- 1 Overhead im Testing (Tests in Pull Request + Test auf Develop Branch)
- 2 Teammitglieder sind alles Studenten mit unterschiedlichen Arbeitszeiten (z.B. Samstags)
- 3 Rollbacks würden neuen Sprint stoppen & verzögern
- 4 Hoher manueller Testaufwand