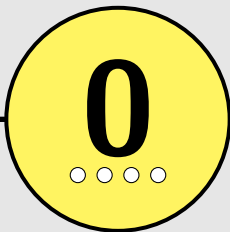




refactoring



„git push – refactoring!“

- Mit diesem Commit übernimmst du die vorherigen drei Commits als deine eigenen. Schieb sie unter diese Karte, alle diese Commits gelten ab jetzt als einer.

1

1

commit

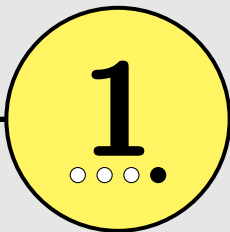
1



„git push!“



conflict



„git push – merge conflict!“

Mit diesem Commit verursachst und löst du einen Merge Conflict:

- Übernimm den vorherigen Commit als deinen eigenen. Schieb ihn unter diese Karte, diese beiden Commits gelten ab jetzt als einer.

2

2

commit

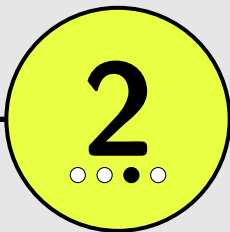
2



„git push!“



conflict



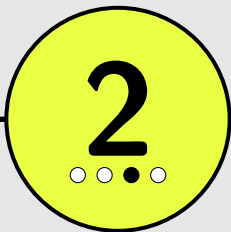
„git push – merge conflict!“

Mit diesem Commit verursachst und löst du einen Merge Conflict:

- Übernimm den vorherigen Commit als deinen eigenen. Schieb ihn unter diese Karte, diese beiden Commits gelten ab jetzt als einer.



request



„git push – merge request!“

Mit diesem Commit *bittest* du deinen Kollegen um ein Code Review, dass er natürlich auch durchführt:

- Der nächsten Spieler muss ein Revert ausspielen oder einmal aussetzen.

3

3

commit

3



„git push!“

3

3

commit

3



„git push!“



conflict



„git push – merge conflict!“

Mit diesem Commit verursachst und löst du einen Merge Conflict:

- Übernimm den vorherigen Commit als deinen eigenen. Schieb ihn unter diese Karte, diese beiden Commits gelten ab jetzt als einer.



request



„git push – merge request!“

Mit diesem Commit *bittest* du deinen Kollegen um ein Code Review, dass er natürlich auch durchführt:

- Der nächsten Spieler muss ein Revert ausspielen oder einmal aussetzen.

4

4

commit

4



„git push!“

4

4

commit

4



„git push!“



conflict



„git push – merge conflict!“

Mit diesem Commit verursachst und löst du einen Merge Conflict:

- Übernimm den vorherigen Commit als deinen eigenen. Schieb ihn unter diese Karte, diese beiden Commits gelten ab jetzt als einer.



request



„git push – merge request!“

Mit diesem Commit *bittest* du deinen Kollegen um ein Code Review, dass er natürlich auch durchführt:

- Der nächsten Spieler muss ein Revert ausspielen oder einmal aussetzen.

5

5

commit

5



„git push!“

5

5

commit

5



„git push!“



request



„git push – merge request!“

Mit diesem Commit *bittest* du deinen Kollegen um ein Code Review, dass er natürlich auch durchführt:

- Der nächsten Spieler muss ein Revert ausspielen oder einmal aussetzen.

6

6

commit

6



„git push!“

6

6

commit

6



„git push!“

7

7

commit

7



„git push!“

7

7

commit

7



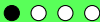
„git push!“

8

8

commit

8



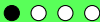
„git push!“

8

8

commit

8



„git push!“

9

9

commit

9



„git push!“

10

10

commit

10
● ● ● ●

„git push!“

12

12

commit

12
... ..

„git push!“



release



v 1.x

„git tag ... und Lieferung!“

- Werte alle Karten im Repository aus und leg sie sowie die aktuellen Issues und alle Handkarten auf den Abwurfstapel.
- Leg diese Karte auf den Abwurfstapel und ziehe einen neuen CR.



release



v 1.x

„git tag ... und Lieferung!“

- Werte alle Karten im Repository aus und leg sie sowie die aktuellen Issues und alle Handkarten auf den Abwurfstapel.
- Leg diese Karte auf den Abwurfstapel und ziehe einen neuen CR.



release



v 1.x

„git tag ... und Lieferung!“

- Werte alle Karten im Repository aus und leg sie sowie die aktuellen Issues und alle Handkarten auf den Abwurfstapel.
- Leg diese Karte auf den Abwurfstapel und ziehe einen neuen CR.



release



v 1.x

„git tag ... und Lieferung!“

- Werte alle Karten im Repository aus und leg sie sowie die aktuellen Issues und alle Handkarten auf den Abwurfstapel.
- Leg diese Karte auf den Abwurfstapel und ziehe einen neuen CR.



release



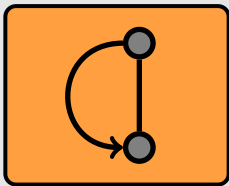
v 1.x

„git tag ... und Lieferung!“

- Werte alle Karten im Repository aus und leg sie sowie die aktuellen Issues und alle Handkarten auf den Abwurfstapel.
- Leg diese Karte auf den Abwurfstapel und ziehe einen neuen CR.



revert



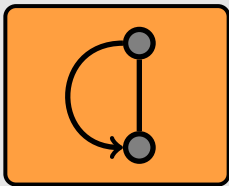
„git revert!“

Der gerade geschriebene Code ist fehlerhaft und muss wieder weg:

- Muss unmittelbar auf einen Commit folgen und negiert ihn.
- Leg diese Karte und den Commit auf den Abwurfstapel.



revert



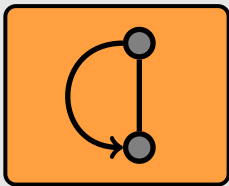
„git revert!“

Der gerade geschriebene Code ist fehlerhaft und muss wieder weg:

- Muss unmittelbar auf einen Commit folgen und negiert ihn.
- Leg diese Karte und den Commit auf den Abwurfstapel.



revert



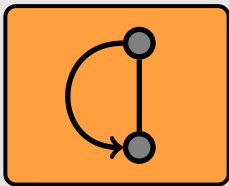
„git revert!“

Der gerade geschriebene Code ist fehlerhaft und muss wieder weg:

- Muss unmittelbar auf einen Commit folgen und negiert ihn.
- Leg diese Karte und den Commit auf den Abwurfstapel.



revert



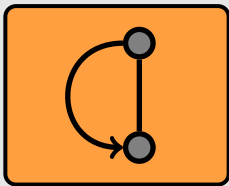
„git revert!“

Der gerade geschriebene Code ist fehlerhaft und muss wieder weg:

- Muss unmittelbar auf einen Commit folgen und negiert ihn.
- Leg diese Karte und den Commit auf den Abwurfstapel.



revert



„git revert!“

Der gerade geschriebene Code ist fehlerhaft und muss wieder weg:

- Muss unmittelbar auf einen Commit folgen und negiert ihn.
- Leg diese Karte und den Commit auf den Abwurfstapel.

??

??

help

???

„git help!“

Du weißt gerade nicht weiter und brauchst Hilfe:

- Leg diese Karte auf den Abwurfstapel.
- Zieh drei Handkarten zusätzlich, nachdem du **aufgestockt** hast.

?

?

status

?

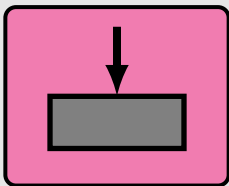
„git status!“

Du hast gerade keine Idee und brauchst erstmal einen Überblick:

- Leg diese Karte auf den Abwurfstapel.
- Zieh eine Handkarte zusätzlich, nachdem du **aufgestockt** hast.



stash

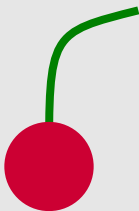


„git stash!“

- Leg bis zu zwei Handkarten verdeckt vor Dir ab.
- Sie können weiterhin wie Handkarten ausgespielt werden, zählen aber nicht beim **Aufstocken** mit und gehen bei einem Release nicht verloren.



cherry-pick

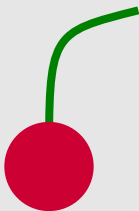


„git cherry-pick!“

- **Stocke** zuerst deine Handkarten auf.
- Wähle dir danach bei einem beliebigen Spieler aus seinen Handkarten eine aus und füge sie deinen hinzu oder spiele sie sofort aus.



cherry-pick

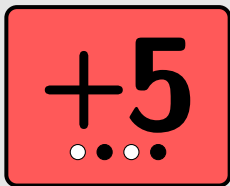


„git cherry-pick!“

- **Stocke** zuerst deine Handkarten auf.
- Wähle dir danach bei einem beliebigen Spieler aus seinen Handkarten eine aus und füge sie deinen hinzu oder spiele sie sofort aus.

5**5**

Bug



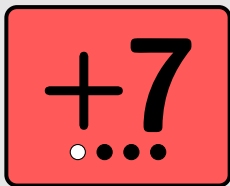
„Ich habe einen Fehler gefunden, der auch noch behoben werden muss.“

- Leg die Karte zum aktuellen CR.
- Für die aktuelle Lieferung müssen noch 5 Codezeilen zusätzlich geschrieben werden.

7

7

Bug



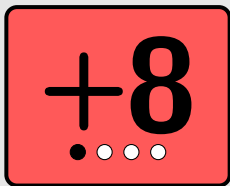
„Ich habe einen Fehler gefunden, der auch noch behoben werden muss.“

- Leg die Karte zum aktuellen CR.
- Für die aktuelle Lieferung müssen noch 7 Codezeilen zusätzlich geschrieben werden.

8

8

Bug

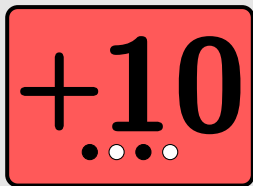


„Ich habe einen Fehler gefunden, der auch noch behoben werden muss.“

- Leg die Karte zum aktuellen CR.
- Für die aktuelle Lieferung müssen noch 8 Codezeilen zusätzlich geschrieben werden.

10**10**

Bug



„Ich habe einen Fehler gefunden, der auch noch behoben werden muss.“

- Leg die Karte zum aktuellen CR.
- Für die aktuelle Lieferung müssen noch 10 Codezeilen zusätzlich geschrieben werden.

10**10**

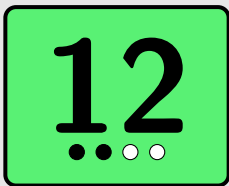
Issue: CR

10

- Für diese Lieferung müssen mindestens 10 Codezeilen geschrieben werden.
- Bei der Lieferung erhält der Auslieferer 5 Zusatzpunkte.

12**12**

Issue: CR



- Für diese Lieferung müssen mindestens 12 Codezeilen geschrieben werden.
- Bei der Lieferung erhält der Auslieferer 6 Zusatzpunkte.

15**15**

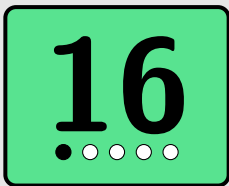
Issue: CR

15
...

- Für diese Lieferung müssen mindestens 15 Codezeilen geschrieben werden.
- Bei der Lieferung erhält der Auslieferer 7 Zusatzpunkte.

16**16**

Issue: CR



- Für diese Lieferung müssen mindestens 16 Codezeilen geschrieben werden.
- Bei der Lieferung erhält der Auslieferer 8 Zusatzpunkte.

20**20**

Issue: CR



- Für diese Lieferung müssen mindestens 20 Codezeilen geschrieben werden.
- Bei der Lieferung erhält der Auslieferer 10 Zusatzpunkte.

24**24**

Issue: CR



- Für diese Lieferung müssen mindestens 24 Codezeilen geschrieben werden.
- Bei der Lieferung erhält der Auslieferer 12 Zusatzpunkte.

26**26**

Issue: CR



- Für diese Lieferung müssen mindestens 26 Codezeilen geschrieben werden.
- Bei der Lieferung erhält der Auslieferer 13 Zusatzpunkte.

30**30**

Issue: CR



- Für diese Lieferung müssen mindestens 30 Codezeilen geschrieben werden.
- Bei der Lieferung erhält der Auslieferer 15 Zusatzpunkte.

32**32**

Issue: CR



- Für diese Lieferung müssen mindestens 32 Codezeilen geschrieben werden.
- Bei der Lieferung erhält der Auslieferer 16 Zusatzpunkte.