# Rebuild Konzept

Wir hatten einmal einen Rebuild Button welcher den Build mit den gleichen Parametern startete wie zuvor. Nun war das Verhalten nicht immer sprechend, denn wenn wir den Head eines Branches buildeten, wollen wir nach einem Commit wieder den Head des Branches rebuilden.

Es gibt aber auch Fälle wo wir genau den gleichen Build nochmals starten wollen. Das ist dann der Fall wenn der Build aufgrund von unerklärlichen Infrastrukturproblemen nochmals laufen soll. Hier sollen einige Anwendungsfälle aufgelistet werden und so die Rebuildbuttons begründet werden.

## Build Pipeline

Bei der Build Pipeline gibt es verschiedene Gründe weshalb sie failed:

* Infrastrukturfehler
* Testfehler
* Buildfehler
* Abbruch durch User

Hier haben wir 2 Rebuild Szenarien:

1. Restart des exakt gleichen gleichen Tags/Commits (Rebuild aufgrund Infrastrukturfehler/Flunky Tests ohne Veränderung des Sources)
2. Restart des Builds mit dem letzten Tags/Commit auf dem Branch (z.B. Test wird korrigiert oder Sourcecode wird korrigiert).

### Szenario Rebuild Tag/Branch

Wird ein Build restarted, welcher auf Basis eines Tags gestartet wurde, muss der Build erneut auf Basis des Tags restartet werden. Das heisst es darf nicht der gleiche commit restartet werden, da sich das Tag ja in der Zwischenzeit verschieben könnte. Der Commit unter dem Tag muss also neu ermittelt werden vor dem Buildstart.

Wird ein branch build restartet, muss der letzte Commit (HEAD) ermittelt werden und mit diesem restartet werden.

### Szenaio exakter Rebuild

Dieser Rebuild wird dann benötigt, wenn z.B. ein Infrastrukturproblem vorlag. Das wäre dann ein simpler Restart der Pipeline wie wir es früher hatten. Also mit identischen Parametern.

## Tagging Pipeline

Die Tagging Pipeline wird über einen Dialog gestartet und verändert den Zustand im Git Repository. Nun kann es aus verschiedenen Gründen passieren, dass die Tagging Pipeline failed.

1. Infrastruktur Probleme
2. Race Condition beim Push eines Commits
3. Andere Fehler in der Pipeline

Ein Tag wird auf dem Head des selektierten Branches gemacht. Dazu wird die commit Id der Buildpipeline übergeben. Nun, wenn das Tagging fehlschlägt, würden sich 2 Rebuild Buttons anbieten:

### Szenario Restart same commit

Wenn z.B. durch ein Infrastrukturfehler oder eine Racecondition ein Tagging nicht abgeschlossen werden kann, will man nochmals den exakt gleichen Tagging Prozess durchführen. Das heisst man startet einen neuen Build basiert auf dem gleichen Commit. Das kommt einem simplen Restart mit den selben Parametern innerhalb Openshift gleich.

### Restart last branch-commit

Bei diesem Restart Knopf, soll der letzte commit des Branches ermittelt und die Pipeline nochmals restarted werden. Dieses Szenario tritt z.B. ein, wenn man eine Tagging Pipeline startete und plötzlich noch einen commit reinnehmen möchte. Man bricht die Pipeline ab, macht seinen commit und restartet die Pipeline. Das Tagging geschieht dann auf Basis des letzten Commits.

## Staging Pipeline

Die Staging Pipeline wird immer auf der gleichen Version aufbauen. Parameter sollten sich nicht verändern. Bei der Staging Pipeline macht nur ein simpler Restart mit den exakt gleichen Buildparametern Sinn. Somit hätten wir hier einfach einen simplen Restart Button.

## Fehlende GitEvent Configs

Beim klicken des Rebuild Buttons kann es sein, dass Fehlermeldungen erscheinen aufgrund fehlenden GitEvent Configs. Hier sollte im Tooltip des Buttons und auch in der Fehlermeldung des Buildstarts eine klare Fehlermeldung dargestellt werden.

## Rebuild Matrix Anforderungen

|  |  |  |  |
| --- | --- | --- | --- |
| **Pipeline Typ** | **Exakter Rebuild**  **(simpler Openshift Pipeline Restart)** | **Tag Rebuild (Im UI gleicher Button wie Branch Rebuild)**  **(Intelligenter Rebuild auf Basis eines Tags)** | **Branch Rebuild (Im UI gleicher Button wie Tag Rebuild)**  **(Intelligenter Rebuild auf Basis eines Branches)** |
| Build | :tick: | :tick: | :tick: |
| Tagging | :tick: |  | :tick: |
| Stageing | :tick: |  |  |

## Buttons auf dem UI

Die Buttons auf dem UI sollen wie folgt aussehen:

