

Modul Softwareentwicklung im Team

* Erforderlich

* Dieses Formular wird Ihren Namen aufzeichnen. Bitte tragen Sie Ihren Namen ein.

Teil 1: Theoriefragen - bitte jeweils allein beantworten

1. Welche Aufgaben gehören nicht zu den Kernfunktionalitäten einer Versionsverwaltung? * (1 Punkt)

- ☐ Protokollierungen der Änderungen.
- ☐ Wiederherstellung von alten Ständen einzelner Dateien.
- ☐ Archivierung der einzelnen Stände eines Projektes.
- ☐ Sicherstellen von einer hohen Testabdeckung.
- ☐ Koordinierung des gemeinsamen Zugriffs von mehreren Entwicklern auf eine Datei.
- ☐ Gleichzeitige Entwicklung mehrerer Entwicklungszweige (engl. Branch) eines Projektes.
- ☐ Optimierung der im Code vorhandenen Algorithmen

2. Markieren Sie alle Arten von Versionsverwaltungen, die es nicht gibt.

* (1 Punkt)

- ☐ Zentrale Versionsverwaltung
- ☐ Kontinuierliche Versionsverwaltung
- ☐ Maschinelle Versionsverwaltung
- ☐ Lokale Versionsverwaltung
- ☐ Verteilte Versionsverwaltung
- ☐ Neuronale Versionsverwaltung

3. Welcher Art von Versionsverwaltung ist Git in Verbindung mit GitHub zuzuordnen?

* (1 Punkt)

- ☐ Zentrale Versionsverwaltung
- ☐ Verteilte Versionsverwaltung
- ☐ Lokale Versionsverwaltung

4. Markieren Sie alle korrekten Aussagen. (Gerne können Sie diese mit Ihrem Cheat-Sheet abgleichen) * (1 Punkt)

- ☐ `git clone [url]` = retrieve an entire repository from a hosted location via URL
- ☐ `git branch [name]` = create a new branch at the current commit
- ☐ `git checkout` = switch to another branch and check it out into your working directory
- ☐ `git merge [branch]` = merge the specified branch's history into the current one
- ☐ `git fetch [alias]` = fetch down all the branches from that Git remote
- ☐ `git merge [alias]/[branch]` = merge a remote branch into your current branch to bring it up to date
- ☐ `git push [alias] [branch]` = Transmit local branch commits to the remote repository branch
- ☐ `git pull` = fetch and merge any commits from the tracking remote branch
- ☐ memorizing: Memorizing stubbornly things is a lot more fun than using a cheat sheet

5. Markieren Sie alle korrekten Aussagen, die auf Pull-Requests zutreffen:
* (1 Punkt)

- ☐ Code-Reviews dienen dazu, verschiedene Meinungen einzuholen, Wissen zu teilen und nicht zuletzt den Fortschritt im Projekt konkret und greifbar zu machen.
- ☐ Code-Reviews kosten aber auch Zeit und können manchmal zu unangenehmen Konflikten im Team führen.
- ☐ Das Code-Review von Pull-Request sollten vom den Entwickler durchgeführt werden, der auch den Pull-Request gestellt hat.
- ☐ Code-Review über Pull Requests tragen zu einer höheren Konsistenz der Umsetzung (einheitlicher Stil, Lesbarkeit) bei.
- ☐ Durch die Verwendung von Pull-Requests verringert sich letztlich auch die langfristig immer entstehende technische Schuld.

6. Unterscheiden Sie die Befehle "git fetch" und "git pull" voneinander.

7. Stark vereinfacht betrachtet könnte ein Softwareentwicklung mit Pull Request wie folgt erfolgen. Bringen Sie die Punkte in die richtige Reihenfolge.

* (1 Punkt)

Ein Thema wird in einem Ticket (Story) hinsichtlich der Anforderungen konkretisiert

Product Owner legt Prioritäten der Storys im Rahmen des Sprints fest

Entwicklung startet und ein neuer Branch wird ausgehend vom Entwicklungsweig erstellt

Nach Abschluss der Entwicklung erfolgt das Review via Pull Requests

Nach Abnahme des Pull Requests wird der Branch wieder mit dem Hauptzweig zusammengeführt (Merge)

Gegebenenfalls Deployment auf Testumgebung und Abnahme/QA

Je nach Releasezyklus werden Entwicklungsweig und Master-Branch zusammengeführt und später released.

Teil 2: Projektbezogene Fragen - bitte im Projektteam gemeinsam beantworten

8. Bitte laden Sie ein Bild(kann auch Screenshot sein) Ihrer Lösung zu folgender Aufgabenstellung hoch:

"Git Branching: **Aufgabe:** Erstellen Sie, nachdem Sie die vorgeschlagenen Kapitel des Onlinetutorial erledigt haben, eine Abbildung, die folgende Begriffe in einem Zusammenhang darstellt: *git clone, git add, git commit, git push, git pull, git branch, git merge*" *

↗ Datei hochladen

Limit für Dateianzahl: 1 Größenlimit für eine einzelne Datei: 10MB Zulässige Dateitypen: Word, Excel, PPT, PDF, Bild, Video, Audio

9. Notieren Sie im nachfolgenden Feld, ob Sie und ggf. mit welcher Review-Konstellationen (siehe Artikel) Sie Pull-Request in Ihrem Projekt verwenden wollen. *

10. Für einen angehenden Fachinformatiker ist das **Thema Softwareentwicklung im Team** relevant:



Dieser Inhalt wurde von Microsoft weder erstellt noch gebilligt. Die von Ihnen übermittelten Daten werden an den Formuläreigentümer gesendet.