Fachhochschule Dortmund
University of Applied Sciences and Arts

Prof. Dr. Dirk Wiesmann
Softwaretechnik D
Praktikum 4

Das Praktikum beschäftigt sich mit konstruktiven Maßnahmen zur Qualitätssicherung. Es werden im weiteren Verlauf die Themen Implementationsdokumentation, Automatisierung und Werkzeugeinsatz betrachtet.
Die professionelle Softwareentwicklung findet heute in der Regel mit Integrierten Entwicklungsumgebungen (IDE) statt. Im Rahmen des Praktikums werden wir als einen bekannten Stellvertreter Eclipse verwenden. In einem Softwareentwicklungsprojekt kann nur dann eine effiziente und effektive Teamarbeit erfolgen, wenn die erstellten Artefakte in einer Versionverwaltung abgelegt werden.
Wir werden im Praktikum *Git* einsetzen. Bei *Git* handelt es sich um ein verteiltes Versionsverwaltungssystem. Dies bedeutet, dass jeder Nutzer die komplette Änderungshistorie auf seinem lokalen Rechner verwaltet. Für die Nutzung von *Git* ist damit nicht zwingend eine Netzwerkverbindung erforderlich. Damit die Mitglieder eines Teams ihre Arbeitsergebnisse einfach austauschen können, kann jedes Team ein entferntes *Repository* (*remote repository*) einrichten. Die entfernten *Repositories* werden hier über das Werkzeug *Gitea* zur Verfügung gestellt. Über ein entferntes Repository kann dann eine Synchronisation der lokalen Repositories erfolgen.
Die Versionsverwaltung ist Bestandteil eines Konfigurationsmanagements. Wir werden ebenfalls `Gitea` als (einfaches) Werkzeug zur Verwaltung von Arbeitsaufträgen in Form von *Tickets* verwenden. Sie erreichen Gitea über das FB4-Netz unter der URL

`http://gitea.pipeline`

Für die Dokumentationsextraktion wird Javadoc verwendet.
In dem zentralen (*remote*) *Git-Repository* stehen die beiden Eclipse-Projekte `BuchV1` und `BuchRedesign` bereit. Die *Repositories* können in *Gitea* einzelnen Organisationen zugeordnet werden. Die beiden Projekte finden Sie unter der Organisation `SWTD-WS2425`.
Auf Ihrem Rechner muss also *Java SDK*, *Eclipse*, *Git* und *JavaDoc* installiert sein. Zudem brauchen Sie einen Zugriff auf `http://gitea.pipeline` im FB4-Netz. Falls dies auf Ihrem Rechner nicht möglich ist, dann beziehen Sie die Buch-Projekte aus dem Ilias-Kurs, verschaffen Sie sich danach einen (theoretischen) Überblick über die Arbeitsweise von *Git* und legen Sie sich zumindest ein lokales Repository an. Für den weiteren Verlauf sollen Sie 2er-Teams bilden. Falls Sie kein Team finden, dann können Sie die folgenden Aufgaben auch alleine bearbeiten. Dann verzichten Sie ggf. auf die Ticketvergabe.
Sie werden mit diesen beiden Projekten im weiteren Praktikumsverlauf arbeiten. Dabei sollen Sie aber ein eigenes *remote Repository* verwenden. Dafür werden Sie nun in *Gitea* eigene Repositories anlegen.

1. Legen Sie zunächst einmalig in *Gitea* einen individuellen Nutzer-Account an. Bitte wählen Sie dazu den Menüeintrag *Register*. Geben Sie die folgenden Daten ein

   - `Username` (Muster dwiesmann)
   - `Email-Address` (FH E-Mail-Adresse)

- `Password` (Hinweis: Es findet keine Synchronisation mit dem FH-Account statt)

2. Bilden Sie danach für das Praktikum Zweier-Teams.

3. Ein Teammitglied legt eine neue Organisation (*Organization*) an. Dieses Teammitglied ist dann Admin der Organisation. Bitte den Organisationsnamen nach dem Muster `SWTD-WS2425-`*Teamname* bilden. Bitte wählen Sie einen eigenen (eher kurzen) *Teamnamen*.

4. Wählen Sie als Orga-Admin die Reiter *Explore* → *Organizations*. Dort selektieren Sie Ihre Organisation. Fügen Sie dort ein neues Team „Praktikum“ hinzu. Klicken Sie die Schaltfläche *Settings* für das Team „Praktikum“ an. Dort aktivieren Sie den Eintrag *Administrator Access*. Selektieren Sie das Team und fügen Sie beide Teammitglieder hinzu.

5. Der Orga-Admin selektiert dann die eigene Organisation (z.B. über den *Explorer*-Reiter). Erzeugen Sie dann mit *New Repository* für `BuchV1` und `BuchRedesign` ein entsprechendes *Repository*.

6. Die beiden Team-Mitglieder werden sich nun gegenseitig Arbeitsaufträge erteilen, um den Source-Code der Projekte in das eigene *Repository* zu übertragen. Ein Team-Mitglied erhält einen Auftrag für `BuchV1` und das andere Team-Mitglied für `BuchRedesign`.

   Selektieren Sie den Reiter *Dashboard*. Wählen Sie ein *Repository*. Wählen Sie den Reiter *Issues* und erstellen Sie mit *New Issue* einen neuen Arbeitsauftrag. Vergeben Sie einen sprechenden Titel und verfassen Sie eine kurze Auftragsbeschreibung. Ordnen Sie den Arbeitsauftrag dem jeweils anderen Teammitglied zu (über *Assignees*). Klicken Sie dann auf *Create Issue*.

7. Bearbeiten Sie nun den Ihnen zugewiesenen Arbeitsauftrag. Wählen Sie den Reiter *Issues* und selektieren Sie den zugewiesenen Arbeitsauftrag. Aktivieren Sie den *Time Tracker* und bearbeiten Sie den Auftrag (wie im Folgenden beschrieben).

8. Sie werden nun eine Kopie des Projekts in Ihrem lokalen Arbeitsbereich erstellen. Sie können dafür die Konsole (falls `Git` entsprechend installiert ist) oder Eclipse verwenden. Falls Sie die Konsole verwenden (empfohlen), dann gehen Sie wie folgt vor.

   1. Wechseln Sie in das gewünschte *Workspace*-Verzeichnis.
   2. Verwenden Sie `git clone` *`url`*, um das Projekt zu klonen. Die passende URL finden Sie, wenn Sie unter *Gitea* unter der Organisation `SWTD-WS2425` das gewünschte *Repository* auswählen. Im Anzeigebereich finden Sie dann das Feld `HTTP`. Mit dem benachbarten *Icon* können Sie die Adresse in die Zwischenablage kopieren.

3. Nun müssen Sie Ihr eigenes *Repository* als *remote Repository* einstellen. Dazu wechseln Sie in das Projektverzeichnis und verwenden dann `git remote set-url origin` ***`url`***. Verwenden Sie hier die URL Ihres eigenen *Repository*. Die Adresse erfahren Sie wieder über `Gitea`. Ob die Einstellung korrekt erfolgt ist, können Sie mit `git remote -v` prüfen.
4. Übertragen Sie nun mit `git push -u origin master` die lokale Kopie in Ihr *remote Repository*.
5. Sie können dann die lokale Kopie als Projekt in Eclipse importieren (*Import* → *Git* → *Projects form Git* → *Existing local repository*).

Falls Sie nur Eclipse verwenden, dann gehen Sie wie folgt vor.

1. Navigieren Sie auf der *Gitea*-Weboberfläche zur Organisation `SWTD-WS2425` und wählen dort das gewünschte *Repository* aus.
2. Kopieren Sie sich den angezeigten Link in Ihre Zwischenablage.
3. Wählen Sie in Eclipse die Option *Import* → *Git* → *Projects from Git* → *Clone URI* aus und tragen Sie dort den zuvor kopierten Link ein. Bestätigen Sie die weiteren Abfragen.
4. Nun haben Sie eine lokale Kopie des Projekts in Ihrem *Workspace*. Nun müssen Sie Ihr eigenes *Repository* als *remote Repository* einstellen. Klicken Sie in der Eclipse-Projektansicht mit der rechten Maustaste auf das Projekt. Wählen Sie dann im Kontextmenü *Team* → *Remote* → *Configure Push to Upstream*.
5. Tragen Sie dort die URL Ihres *remote Repository* ein. Die Adresse erfahren Sie wieder über `Gitea`.
6. Übertragen Sie nun die lokale Kopie in Ihr *remote Repository*. Dazu wählen Sie im Kontextmenü des Projekts *Team* → *Push to Upstream*.

9. Prüfen Sie über die *Gitea*-Weboberfläche, ob die Dateien in Ihrem zentralen Projekt-*Repository* angekommen sind.

10. Importieren Sie nun das Buch-Projekt aus dem *Gitea*-Repository, welches von Ihrem Team-Mitglied dort abgelegt wurde. Der entsprechende Arbeitsauftrag in *Gitea* muss natürlich fertiggestellt sein. Denken Sie also daran, auch Ihren eigenen Arbeitsauftrag zu beenden, indem Sie den *Time Tracker* stoppen und den Arbeitsauftrag abschließen (ggf. mit einem Eintrag in das Kommentarfeld).

    1. Gehen Sie unter *Gitea* in das gewünschte *Repository*. Kopieren Sie die angezeigte URL.
    2. Importieren Sie in Eclipse ein Projekt aus Git (*Projects from Git*). Wählen Sie als Quelle den Eintrag *Clone URI*. Tragen Sie dort die kopierte Adresse aus *Bitbucket* ein. Führen Sie den Import aus. Alternativ können Sie auch wieder über die Konsole arbeiten.

11. Jedes Team-Mitglied erstellt nun in *Gitea* einen weiteren Auftrag der folgenden Form.

    - *Ein Javadoc-Kommentar für die Klasse* `Buch` *erstellen.*
    - *Ein Javadoc-Kommentar für die Klasse* `Buchverwaltung` *erstellen.*

    Weisen Sie den Auftrag einem anderen Team-Mitglied zu. Führen Sie den Ihnen zugewiesenen Aufrag durch und schreiben Sie die Änderungen in das *Gitea*-Repository.

    Hinweise: Änderungen müssen erst in den (lokalen) *Staging*-Bereich geschrieben werden, bevor ein *Commit* erfolgen kann. Der *Staging*-Bereich wurde in älteren Git-Versionen als Index bezeichnet. Im Kontext-Menü von Eclipse findet sich daher noch der Eintrag *Team* → *Add to Index*. Eclipse bietet auch verschiedene *Git-Views* (z.B. *Git Stagin*).

    Im Kontext-Menü des Projekts findet sich zudem der Eintrag *Team* → *Remote*. Über die enthaltenen Menüeinträge können Daten mit dem entfernten *Repository* ausgetauscht werden (oder direkt *Team* → *Fetch from Upstream*. Mit einem *fetch* werden die dort festgeschriebenen Artefakte in das lokale Repository übernommen. Dort werden die Artefakte aber zunächst in einem eigenen Zweig (*branch*) gespeichert (*remote branch*), um die Änderungen zunächst noch prüfen zu können.

    Mit dem Eintrag *Switch To* kann zwischen verschiedenen Zweigen des lokalen *Repository* gewechselt werden.