Universität Rostock

Traditio et Innovatio

Imperative Programmierung

Übung

Willi Brekenfelder

Fakultät für Informatik und Elektrotechnik (IEF)
Universität Rostock

Herzlich Willkommen!

Übungsleiter

	M.Sc. Willi Brekenfelder
Adresse	Architektur von Anwendungssystemen Fakultät für Elektrotechnik und Informatik Universität Rostock Albert-Einstein-Str. 22
Büro	Raum 274
Sprechzeiten	Nach Vereinbarung
Telefon	0381 / 498 7638
E-Mail	willi.brekenfelder@uni-rostock.de
WWW	https://www.ava.informatik.uni-rostock.de

Organisation – Übung

- Übung
  - Gruppe 1:
    - Wöchentlich dienstags: 09:20 – 10:50 Uhr
    - PC 201, Konrad-Zuse-Haus

- Inhalte
  - Besprechung der Hausaufgaben
  - Vertiefung des Vorlesungsstoffes
  - Vorbereitung auf die Abschlussprüfung

Prüfungsvorleistung

- Hausaufgaben
  - Aufgabenblätter über Stud.IP verfügbar
  - Wöchentliche Aufgabenserie
- Bearbeitung der Aufgaben
  - Gruppen von 3-4 Teilnehmern
  - **Eintragung: Stud.IP der Vorlesung** (Teilnehmende → Gruppen)
  - Eintragung bis 23.10, danach nur noch durch Übungsleiter möglich
  - Bearbeitungszeit beträgt 1 Woche
- Abgabe der Lösungen erfolgt über Subversion-System (SVN)
  - https://svn.informatik.uni-rostock.de/lehre/ip2024/groups/## wobei ## für Ihre Gruppennummer steht
- Bewertung
  - Voraussetzung ist das Einchecken der Lösungen bis zum angegebenen Abgabedatum

Arbeitsumgebung @ Home (Variante 1)

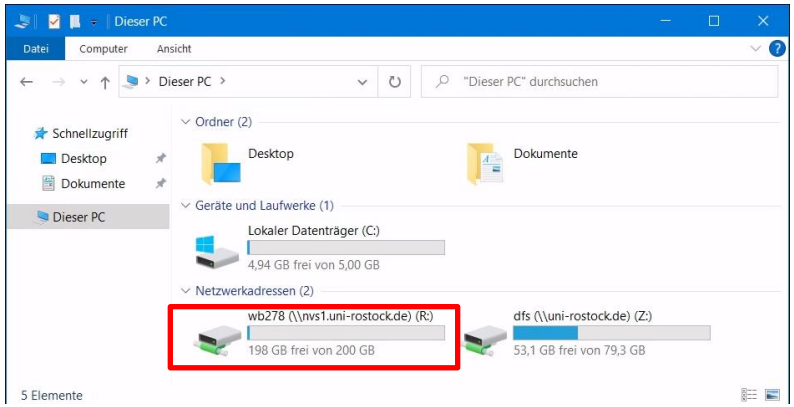
- MinGW64
  - https://github.com/brechtsanders/winlibs_mingw/releases/download/12.2.0-14.0.6-10.0.0-ucrt-r2/winlibs-i686-posix-dwarf-gcc-12.2.0-mingw-w64ucrt-10.0.0-r2.zip
  - Entpacken und Pfad des „bin“-Verzeichnisses zu PATH System-Umgebungsvariablen hinzufügen
- SmartSVN
  - https://www.smartsvn.com/download/
  - Unten auf der Seite die SVN Binaries herunterladen, entpacken und Pfad des Verzeichnisses zu PATH System-Umgebungsvariablen hinzufügen
- Visual Studio Code mit Erweiterungen
  - https://code.visualstudio.com/
  - Nach der Installation notwendige Erweiterungen installieren:
    - C/C++ (ID: ms-vscode.cpptools)
    - SVN-SCM (ID: johnstoncode.svn-scm)
    - PDF Preview (ID: analytic-signal.preview-pdf)

Alternative Arbeitsumgebung

- Remote auf unicomp
  - RDP auf unicomp.uni-rostock.de
  - Anmelden mit der Uni-E-Mail (<vorname>.<nachname>@uni-rostock.de)
  - Software aus Arbeitsumgebung Variante 1 vorinstalliert

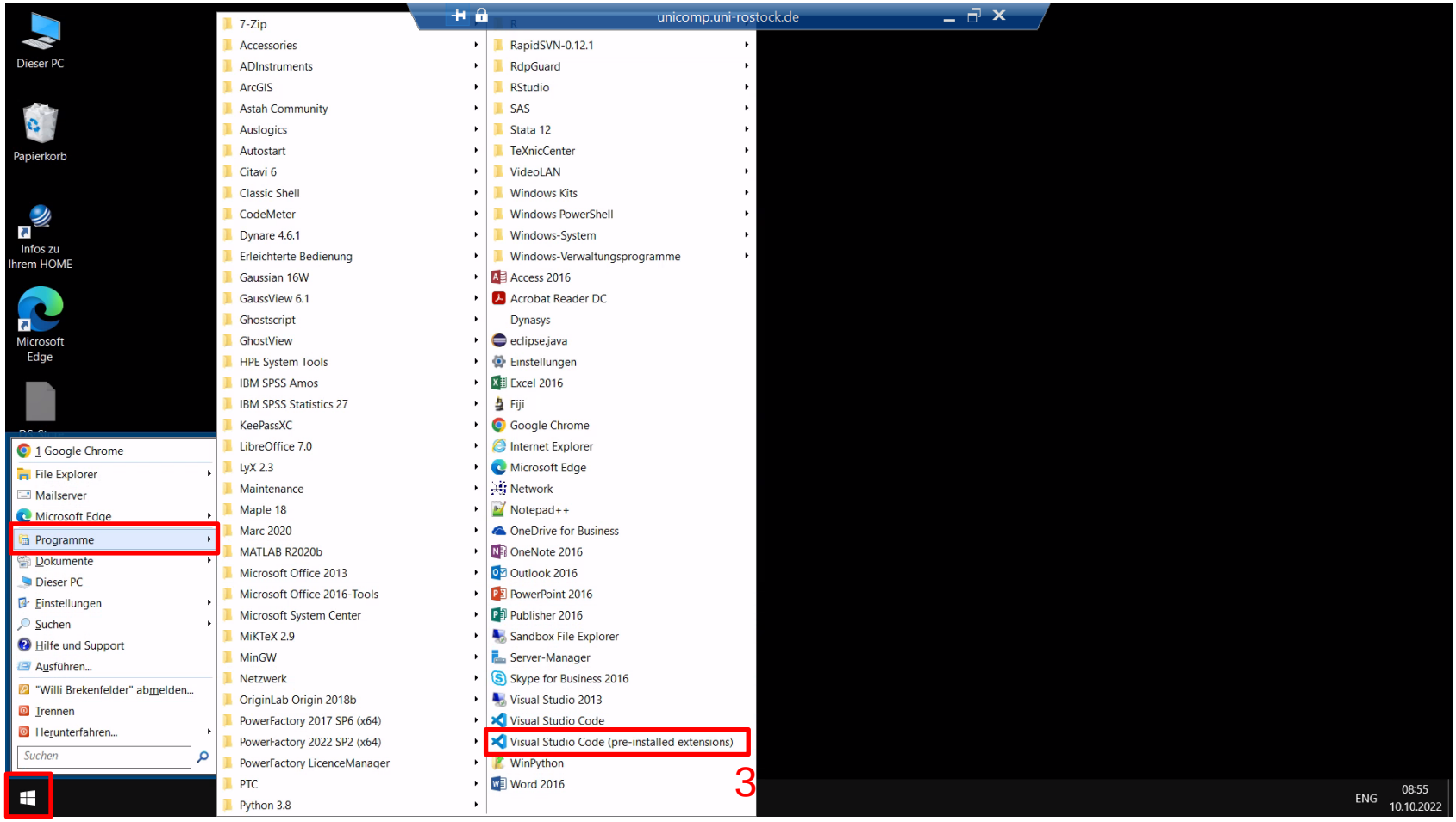
Arbeitsumgebung auf unicomp

- Remotedesktopververbindung öffnen
  - Computer: unicomp.uni-rostock.de
  - Benutzername: <vorname>.<nachname>@uni-rostock.de
  - → Verbinden
- Arbeitsordner anlegen
  - Explorer öffnen
  - Laufwerk mit eurem Nutzerkürzel (in meinem Fall "R:") finden und Ordner für "imperative_programmierung" erstellen.



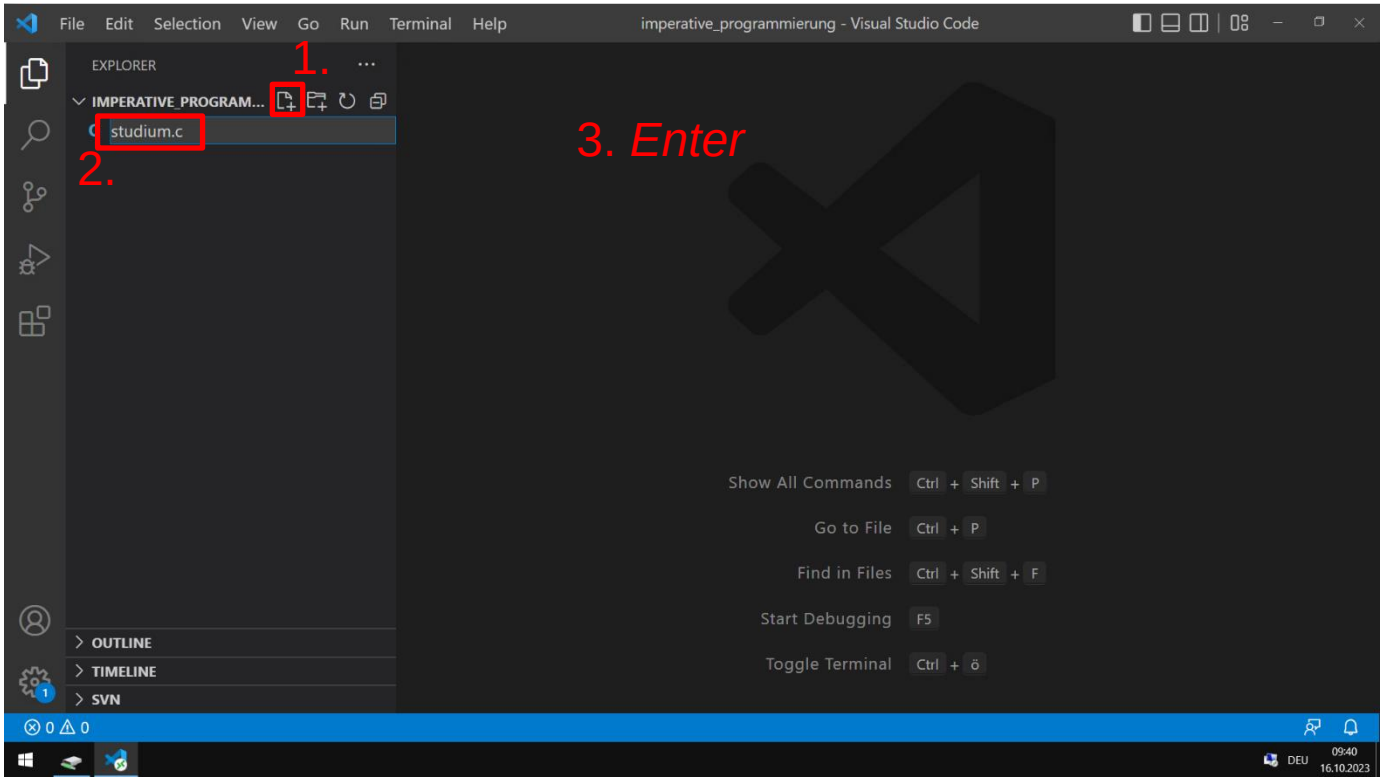
Arbeitsumgebung auf unicomp

> Visual Studio Code starten (mit "pre-installed extensions")



Arbeitsumgebung auf unicomp

- Arbeitsordner mit VS Code öffnen
  - *File* → *Open Folder…* → zu angelegtem Arbeitsordner navigieren → *Select Folder*
- Erste C-Datei anlegen



Mein erstes C-Programm

- Programm anlegen
  - File → New Text File
  - File → Save → Ordner wählen, studium.c eingeben → speichern
- Programm/Klasse schreiben

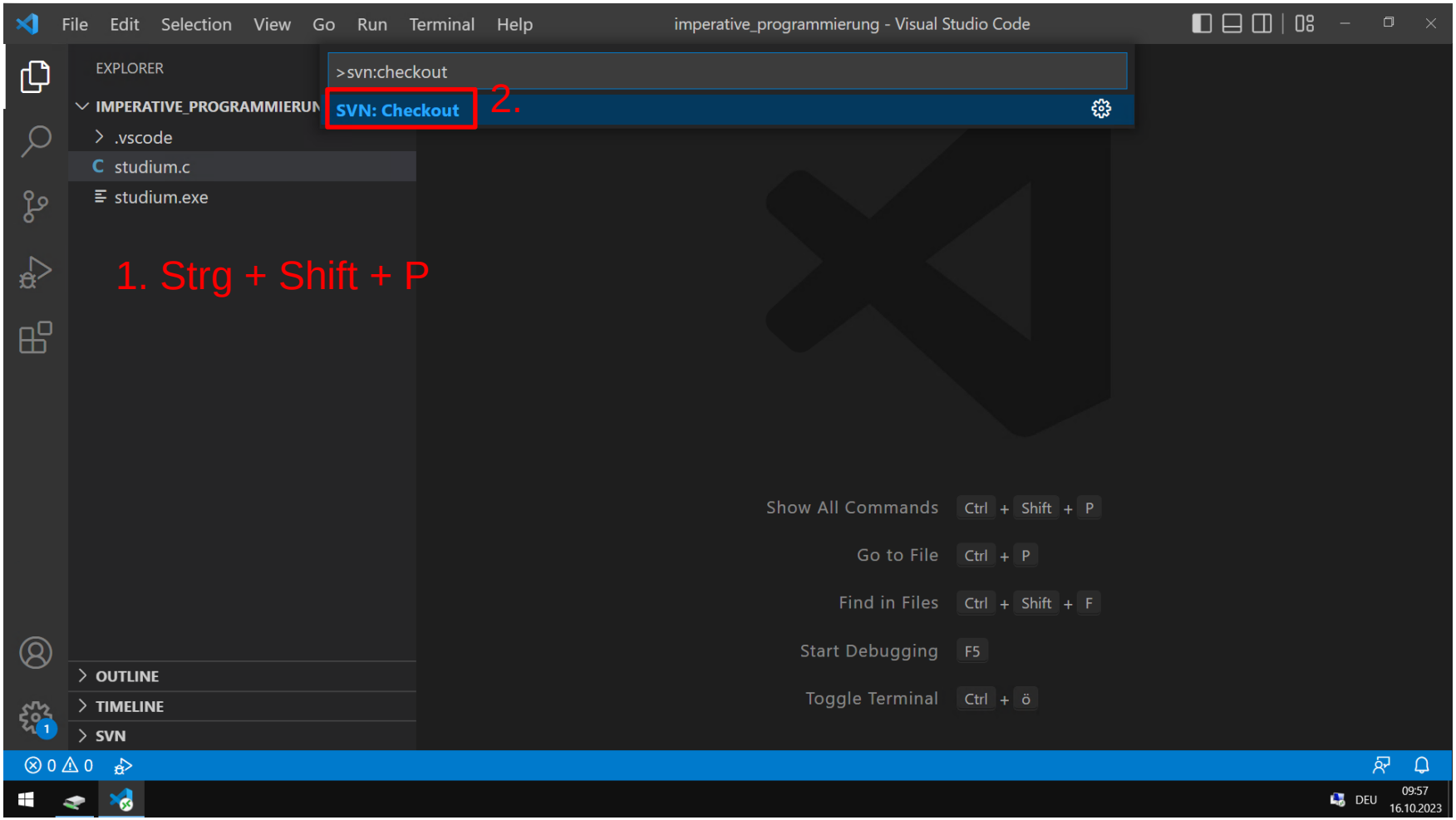
```c
#include <stdio.h>

int main() {
    char studiengang[] = "Informatik";
    printf("Ich freue mich sehr auf mein Studium der %s!", studiengang);
    return 0;
}
```

- Programm ausführen
  - Run → Run Without Debug → C/C++ (GDB/LLDB) → C/C++: gcc.exe Akive Datei erstellen und debuggen
  - Ausgabe in Terminal betrachten → Herzlichen Glückwunsch

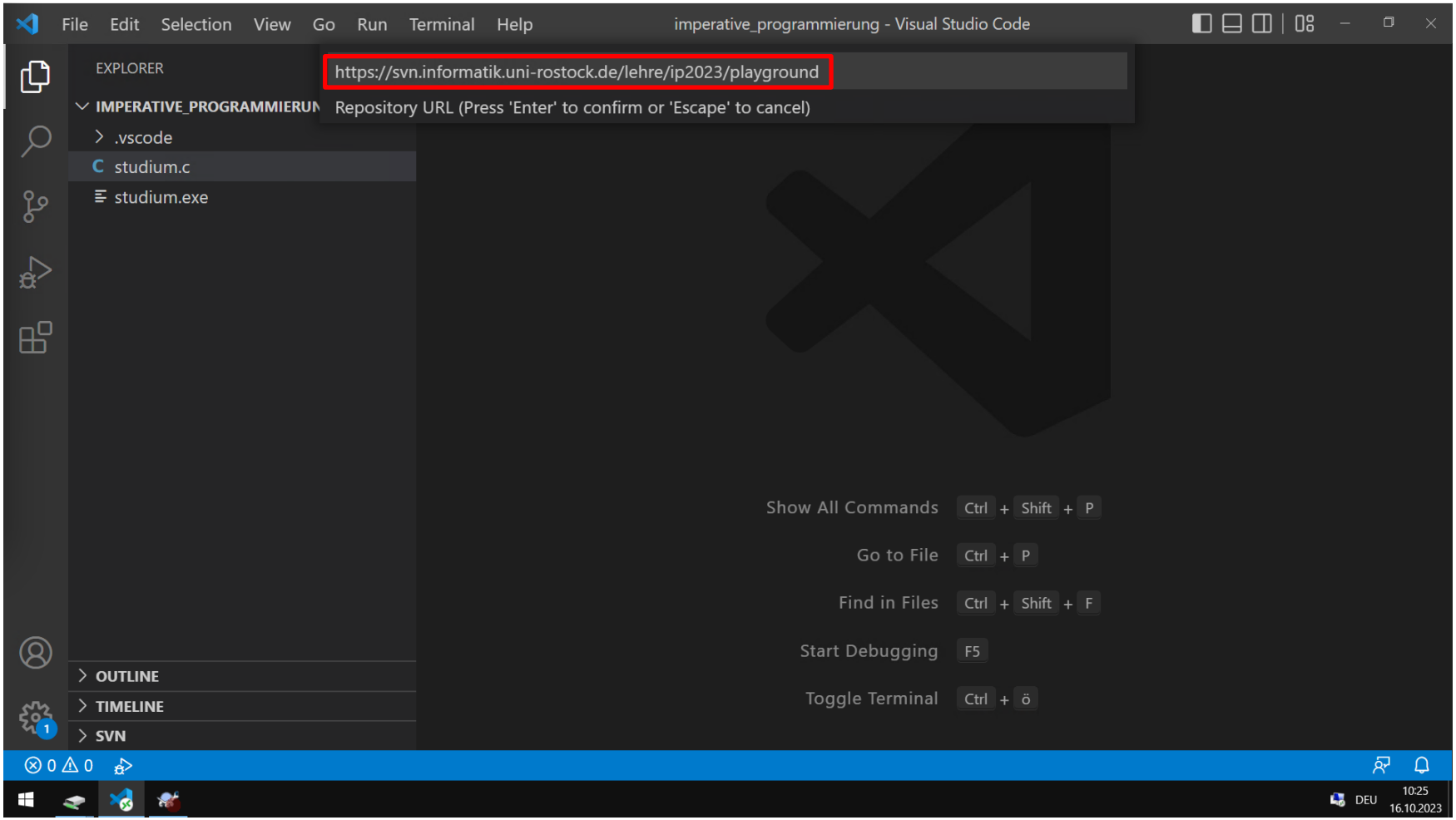
SVN-Repository hinzufügen

> VS Code Kommandos anzeigen (*Strg* + *Shift* + *P*) → SVN checkout



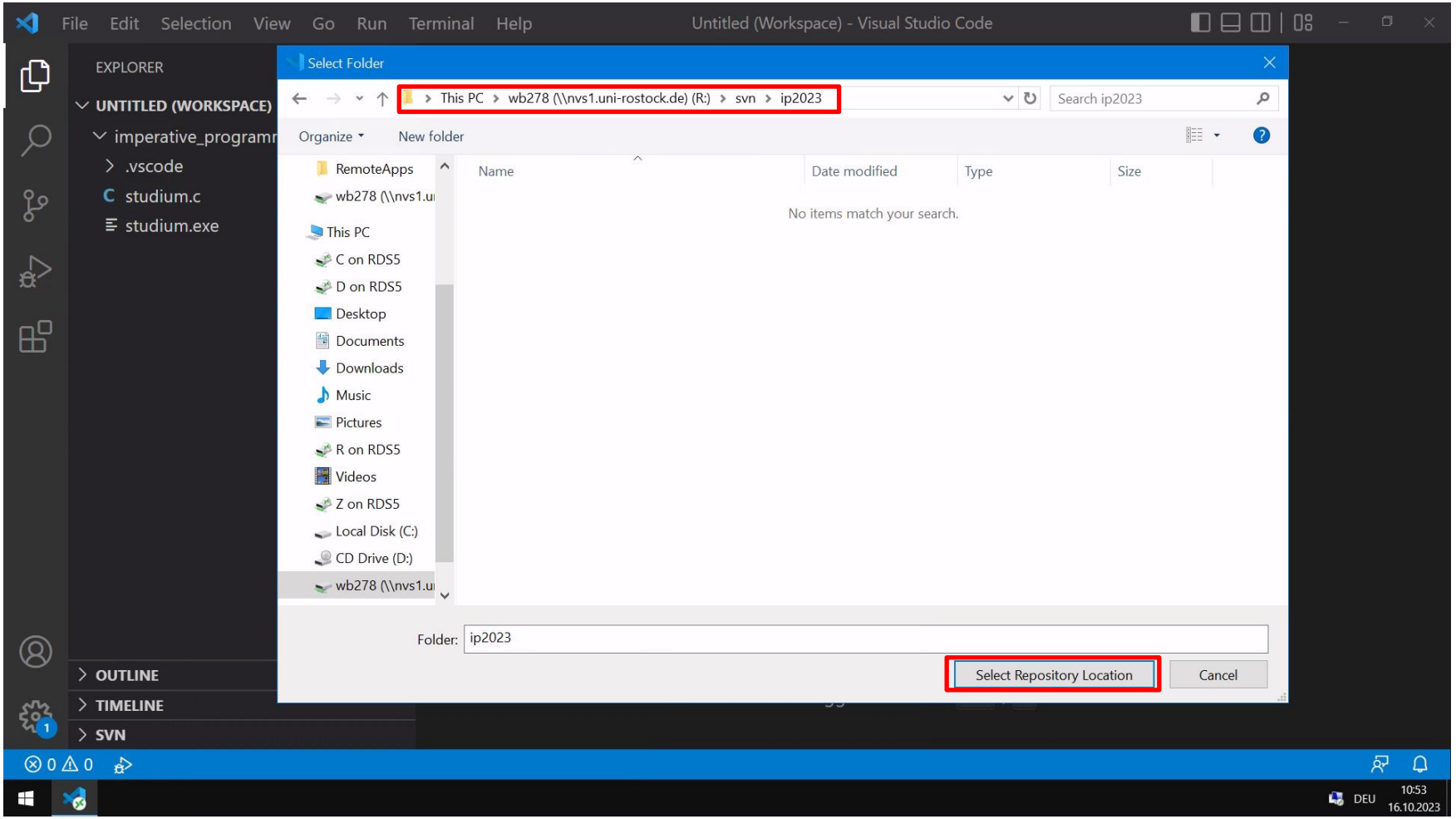
SVN-Repository hinzufügen

> **https://svn.informatik.uni-rostock.de/lehre/ip2024/playground**



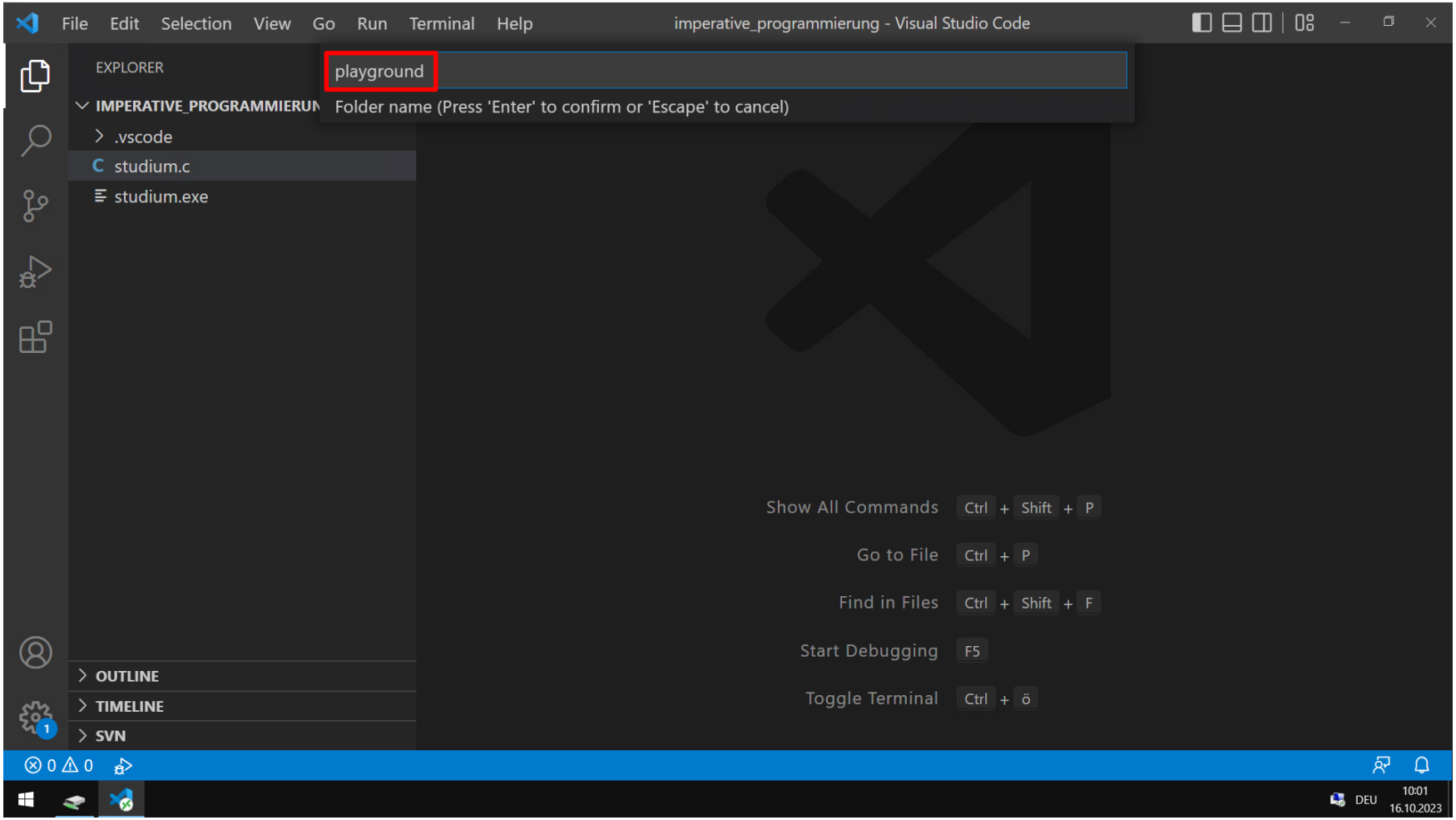
SVN-Repository hinzufügen

> Pfad auswählen (möglichst eigener SVN-Ordner)



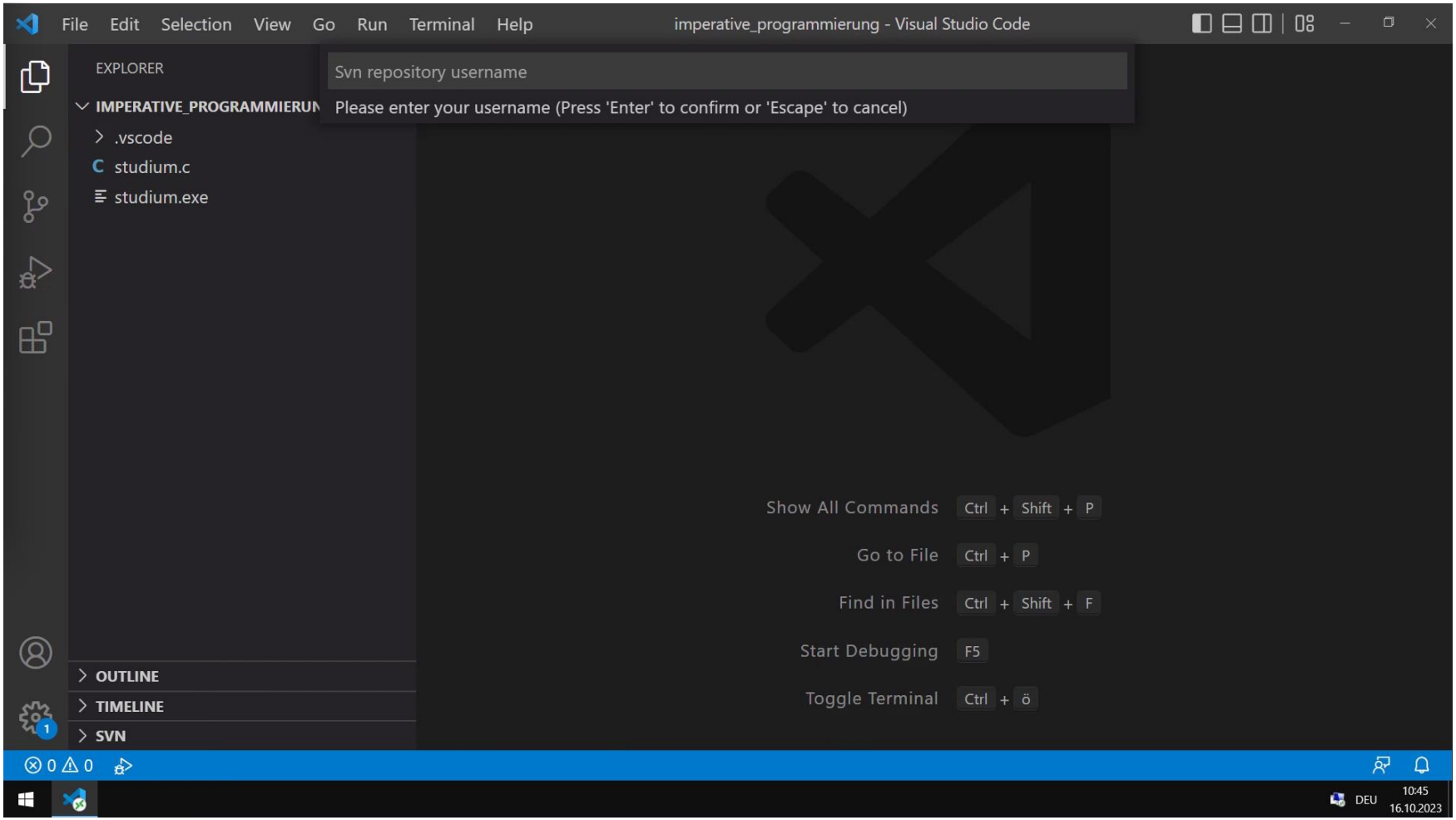
SVN-Repository hinzufügen

> Name des Unterordners wählen



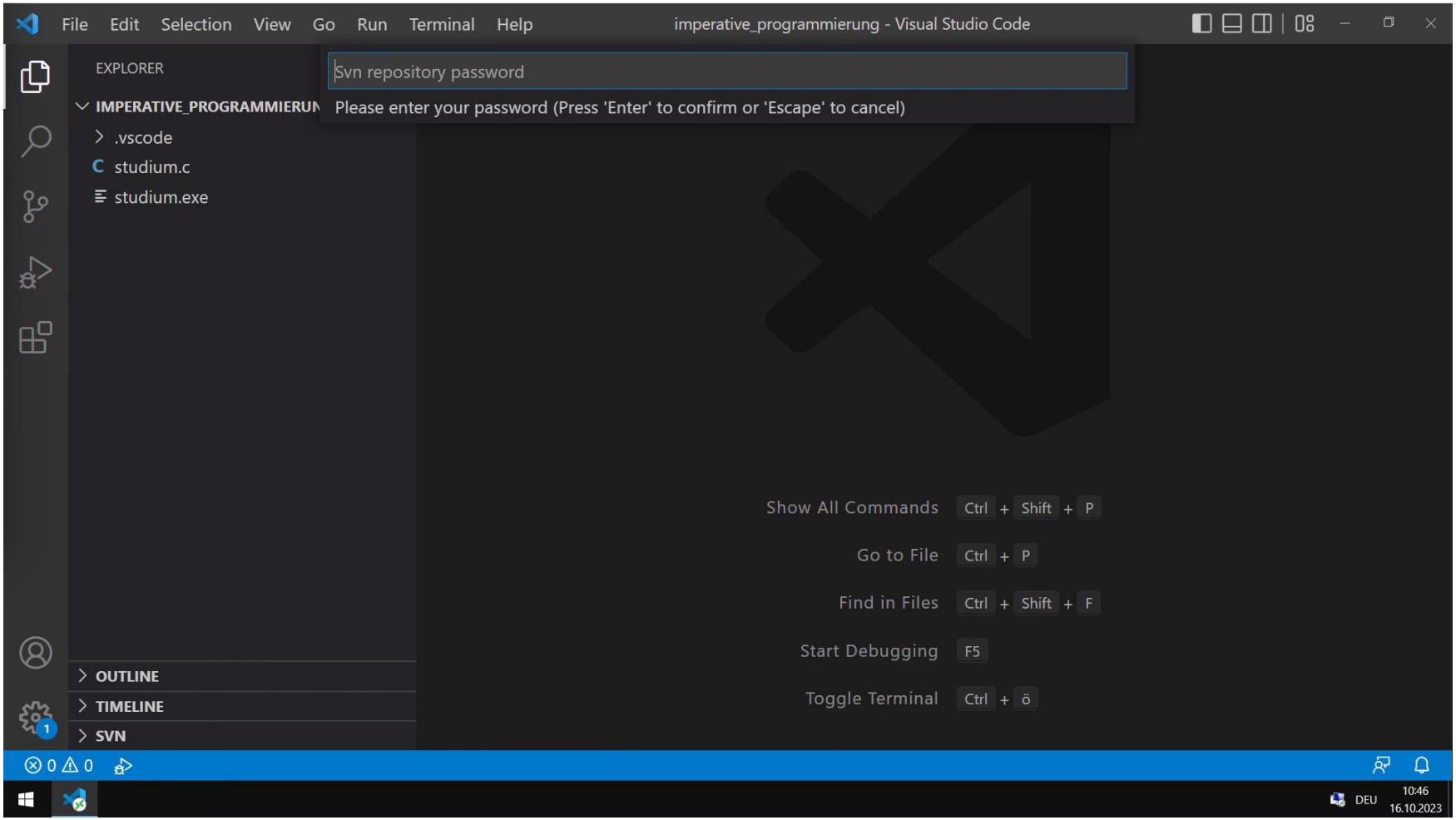
SVN-Repository hinzufügen

> Benutzername: <nutzerkürzel>



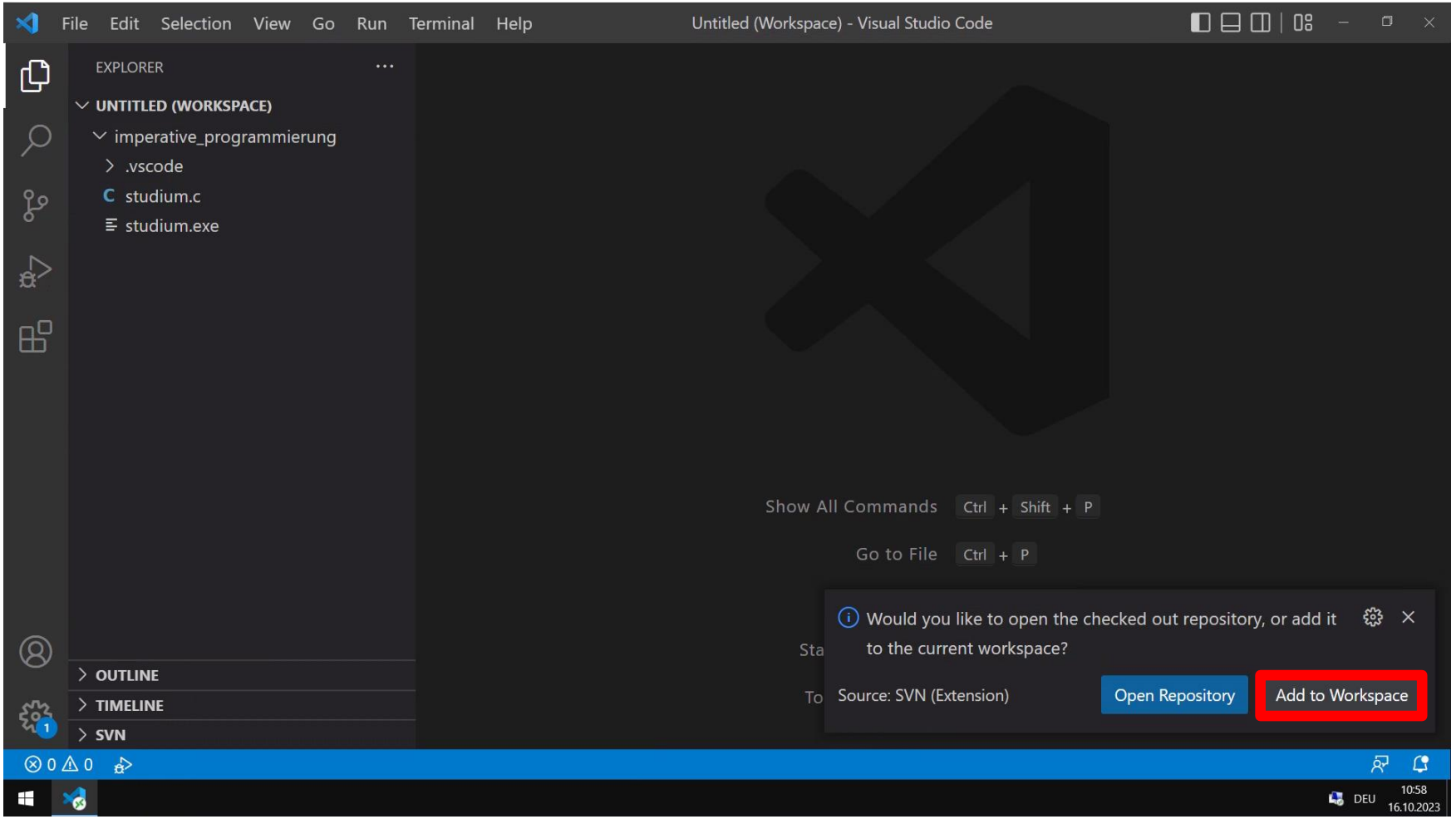
SVN-Repository hinzufügen

> Passwort: ITMZ-Account-Passwort (E-Mail, StudIP, …)



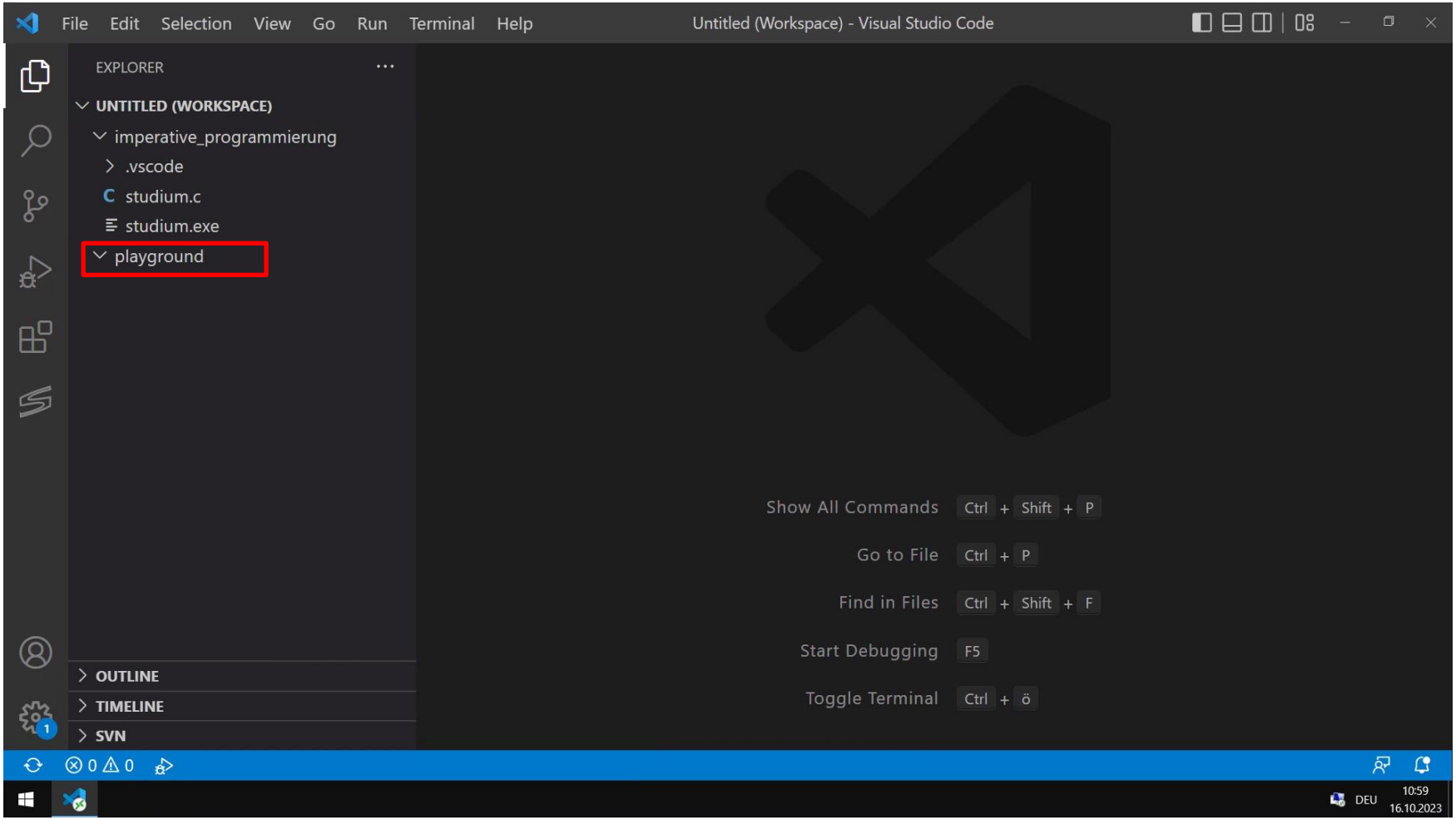
SVN-Repository hinzufügen

> SVN-Ordner zu Workspace hinzufügen

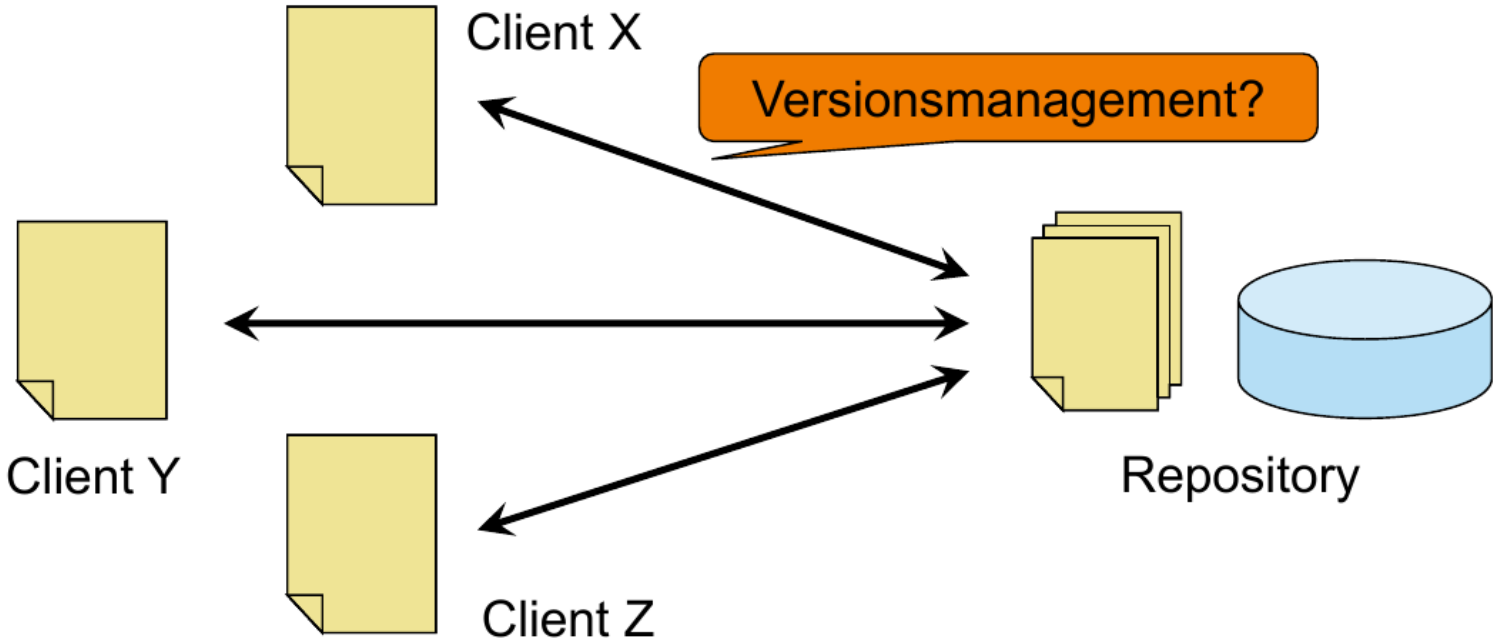


SVN-Repository hinzufügen

> Fertig



Verteilte Entwicklung



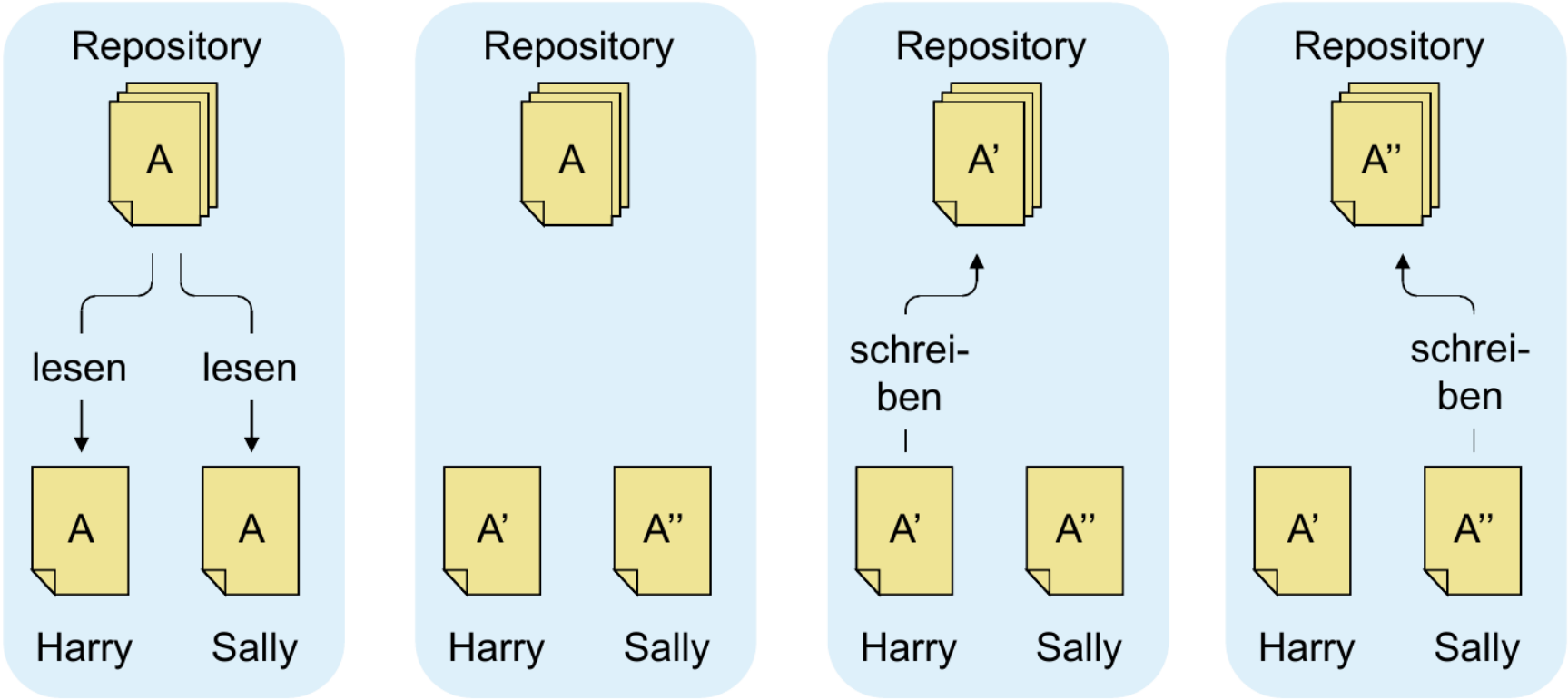
Arbeitskopie

> Lokale Kopie zur Bearbeitung beim Client
> Kann Teile oder gesamtes Repository umfassen

Repository

> Lager für alle Ressourcen eines Projektes (inklusive Historie)
> Oft an zentralem Ort realisiert → Repository Server

Wozu eine Versionskontrolle?



Lesen

> Harry und Sally erzeugen eigene Kopien

Editieren

> Beide Arbeiten gleichzeitig auf ihren Kopien

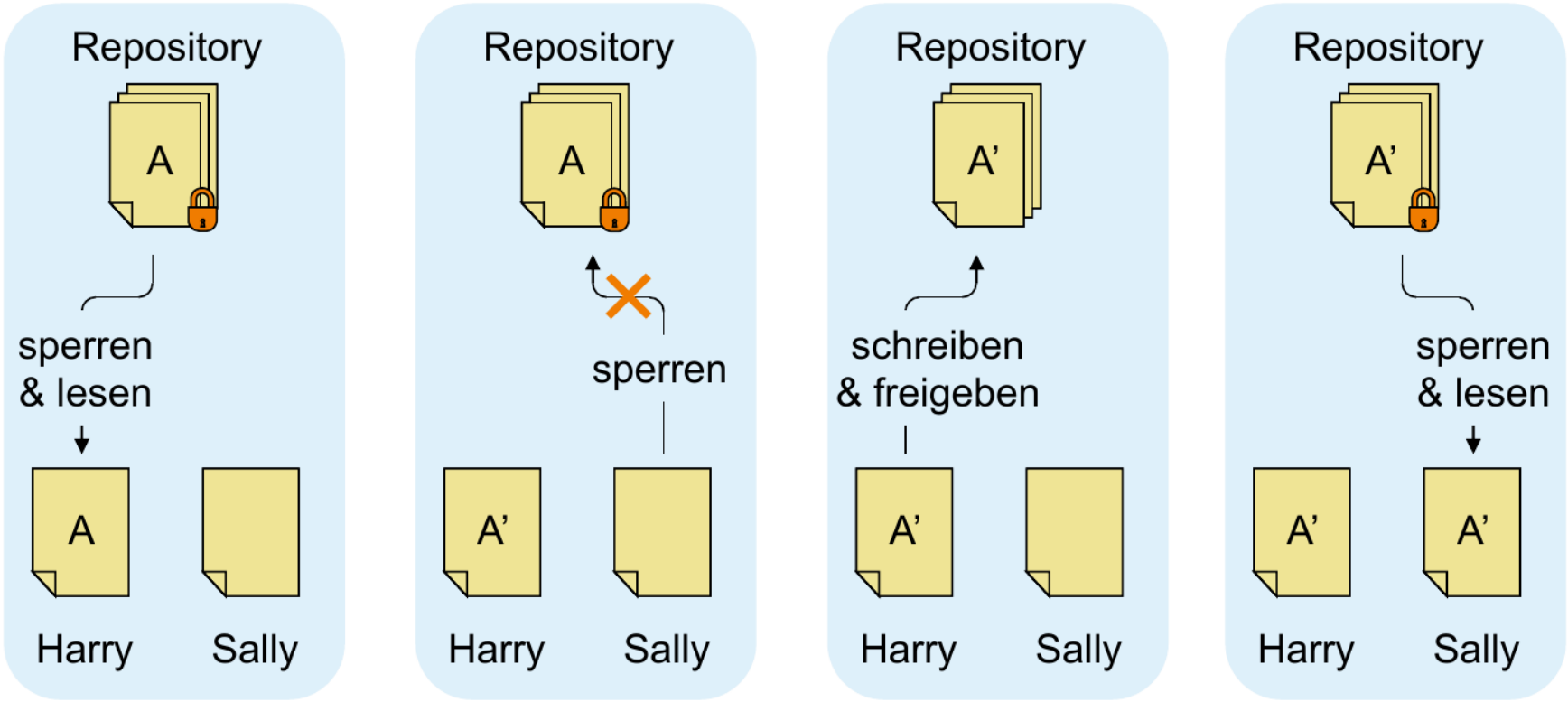
Schreiben

> Zurückschreiben von Harrys Änderungen

Überschreiben

> Verlust von Harrys Änderungen

Lösung: Sperren → Ändern → Freigeben



Sperren

> Harry sperrt Dokument zur Bearbeitung

Ändern / Warten

> Sallys Sperrversuch scheitert
> Warten auf Freigabe

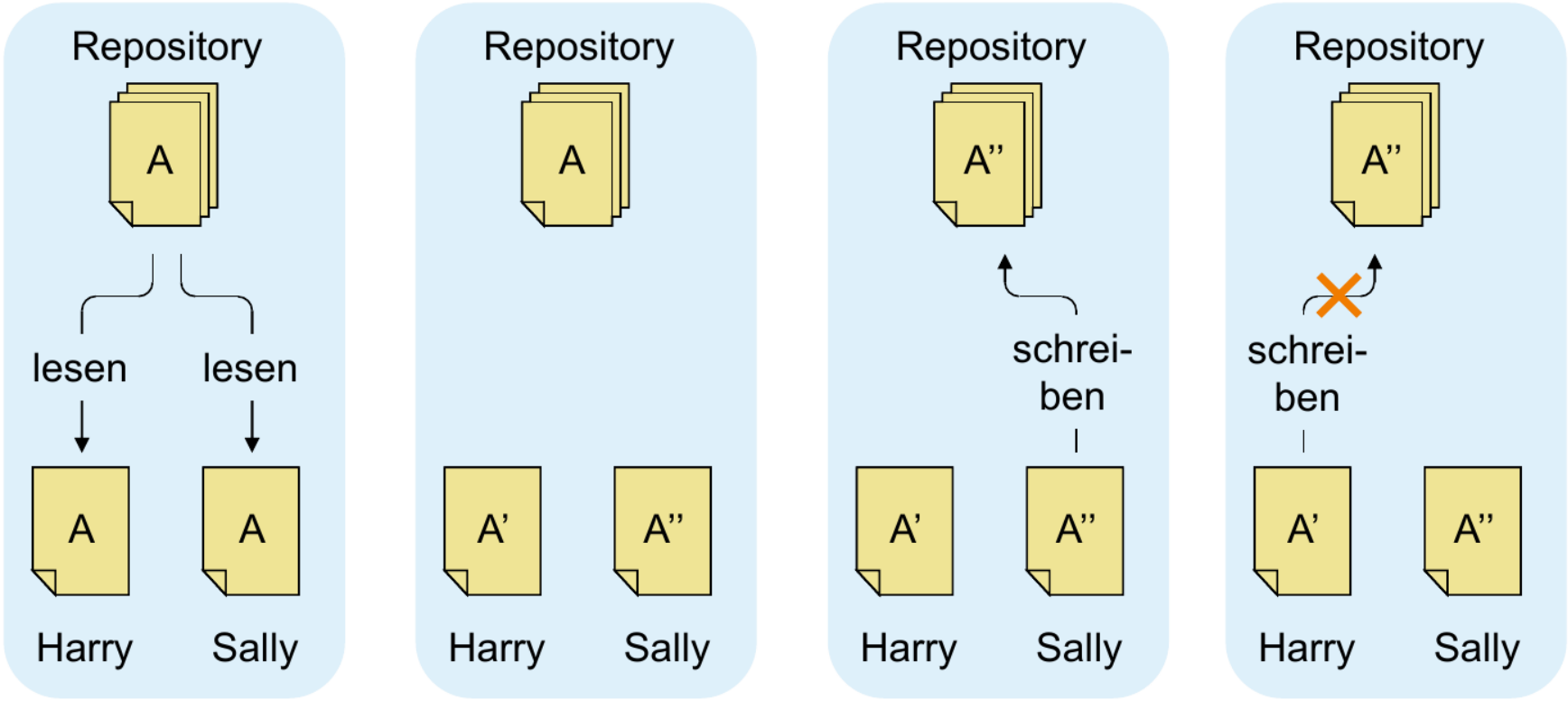
Freigeben

> Harrys Freigabe erfolgt nach Änderung

Sperren

> Sally sperrt Dokument zur Bearbeitung

Lösung: Kopieren → Ändern → …



Kopieren

> Harry und Sally erzeugen eigene Kopien

Editieren

> Beide arbeiten gleichzeitig auf ihren Kopien

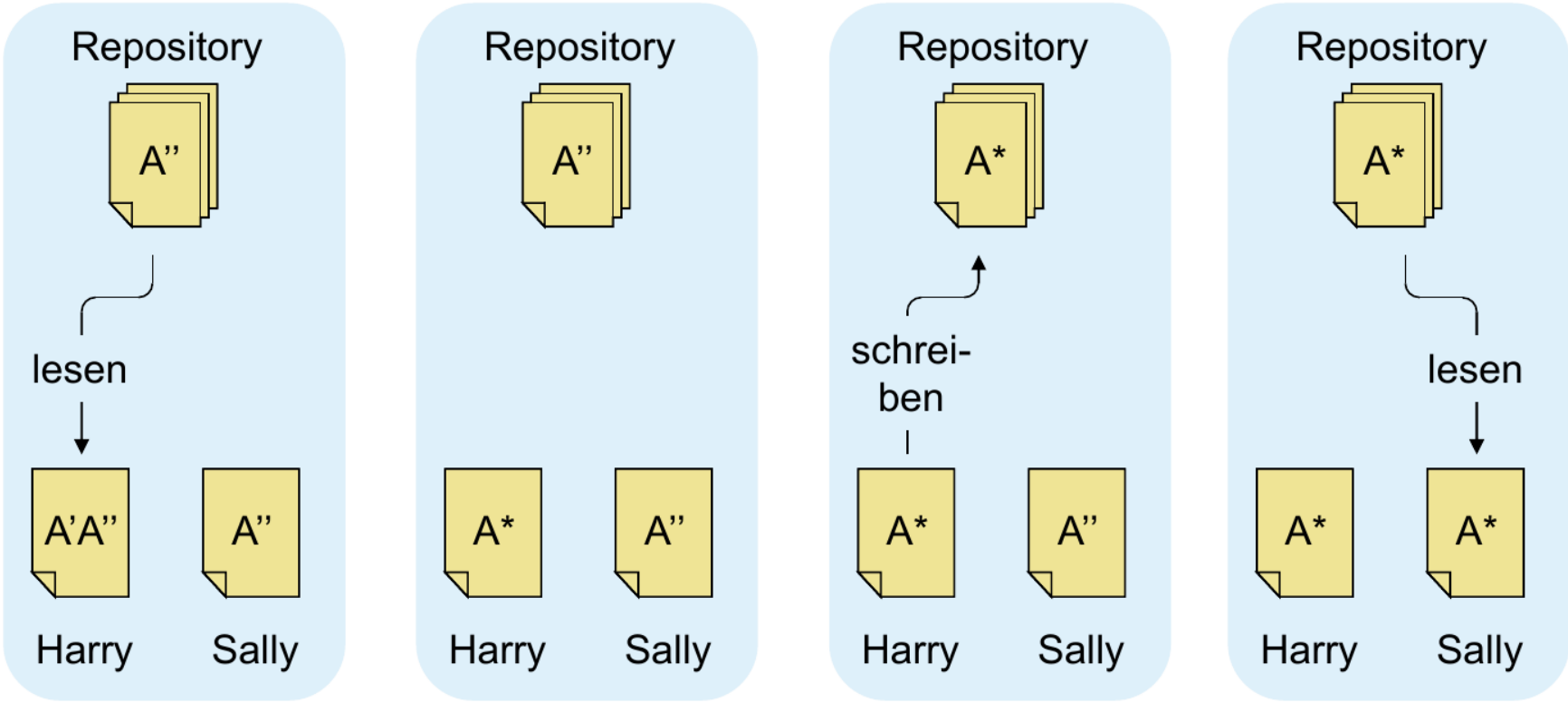
Schreiben

> Sally schreibt Änderungen zuerst zurück

Schreiben

> Harrys Schreibversuch schlägt fehl (da nicht mehr aktuell)

… → Zusammenführen



Update

> Harry vergleicht seine Version mit aktueller

Zusammenführen

> Erstellung einer gemeinsamen Version
> Konfliktpotential

Schreiben

> Harry schreibt zusammengeführte Version zurück

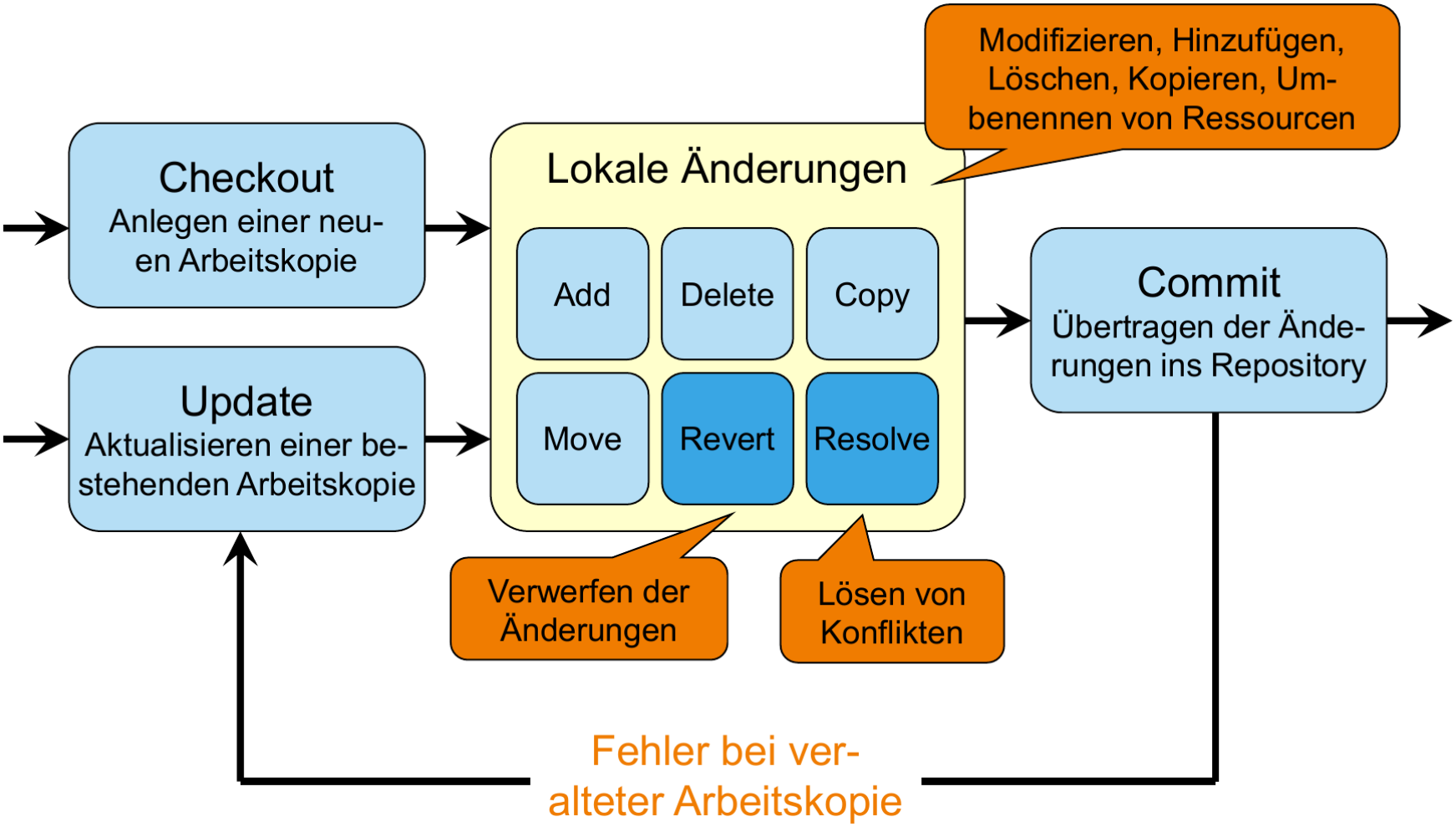
Lesen

> Sally erhält neue Version vom Repository

Subversion

- Versionsmanagementsystem
  - Zentrales Repository für Ordner und Dateien
  - Versionsmanagement → Revisionen des gesamten Repositories
  - Historie (aller Revisionen) einer Ressource verfügbar
    → Speicherung des Originals und sämtlicher Änderungen
  - Unterstützung von Kopieren / Ändern / Zusammenführen

Subversion Workflow



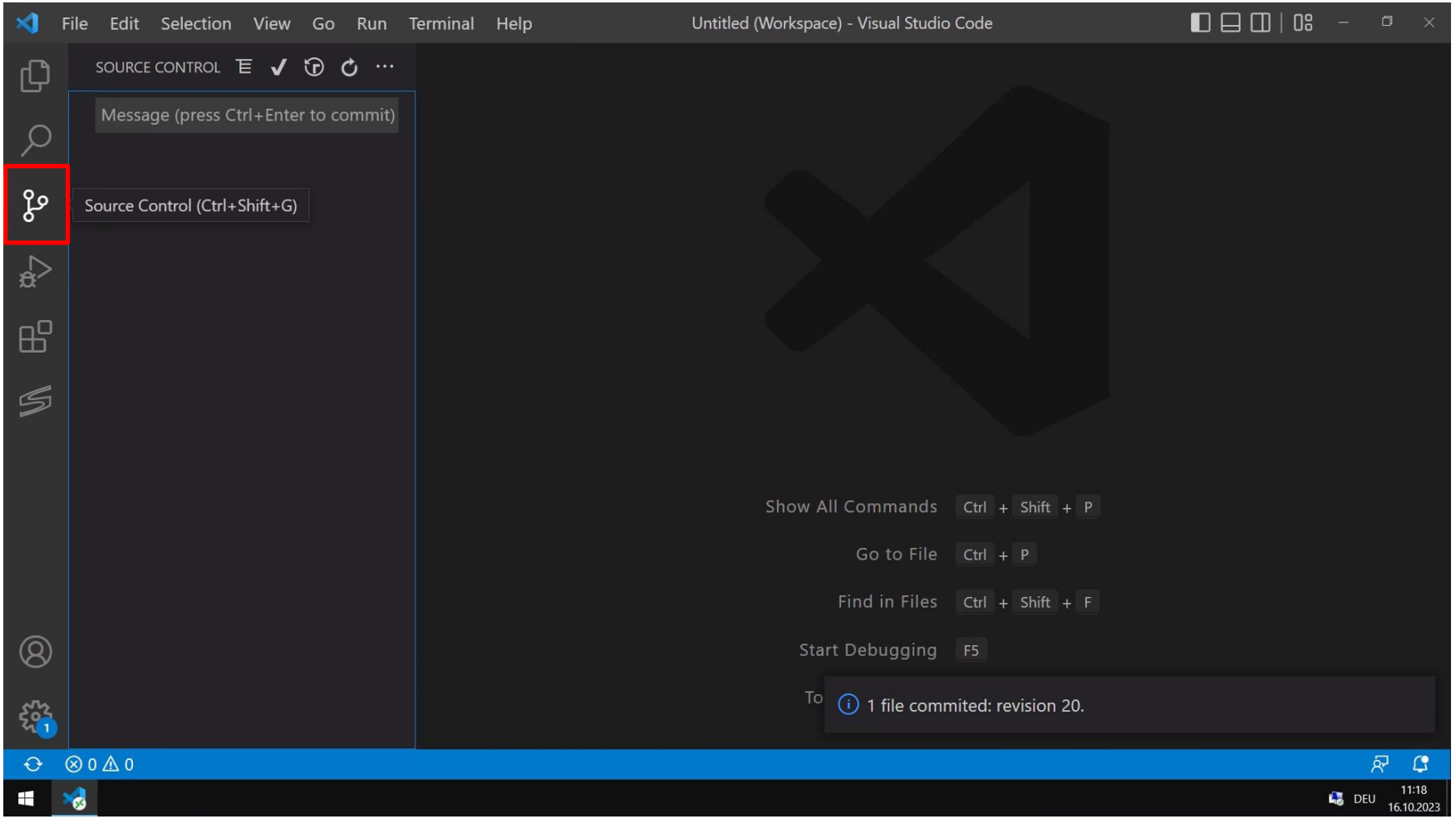
Konflikte

- Scheitern der automatischen Zusammenführung von Revisionen beim Aktualisieren → Konflikt
- Zusätzliche Kopien bei Dateien mit Konfliktstatus
  - DATEI: Datei mit enthaltenen Konfliktmarkierungen
  - DATEI.mine: Eigene Arbeitskopie der Datei vor Aktualisierung
  - DATEI.rXX: Aktuelle Revision XX aus dem Repository
  - DATEI.rYY: Ausgangsrevision YY (< XX) der Arbeitskopie

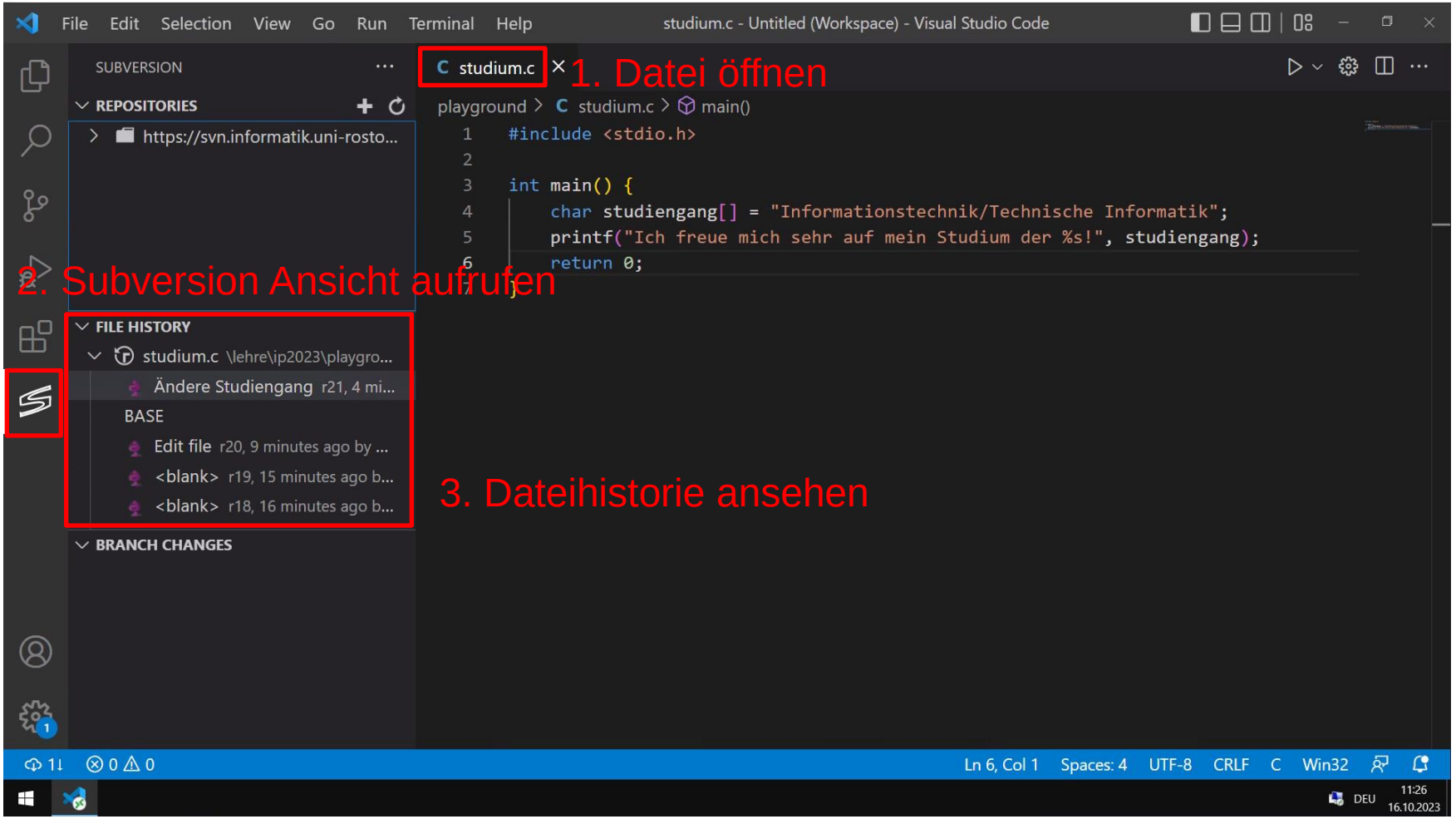
- Konflikte werden abschnittsweise innerhalb der Datei markiert.

```
... Textzeilen der Datei ohne Konflikt
<<<<<<< .mine
Zeilen der Arbeitskopie
=======
Zeilen der aktuellen Repository-Revision
>>>>>>> .r42
Textzeilen der Datei ohne Konflikt ...
```

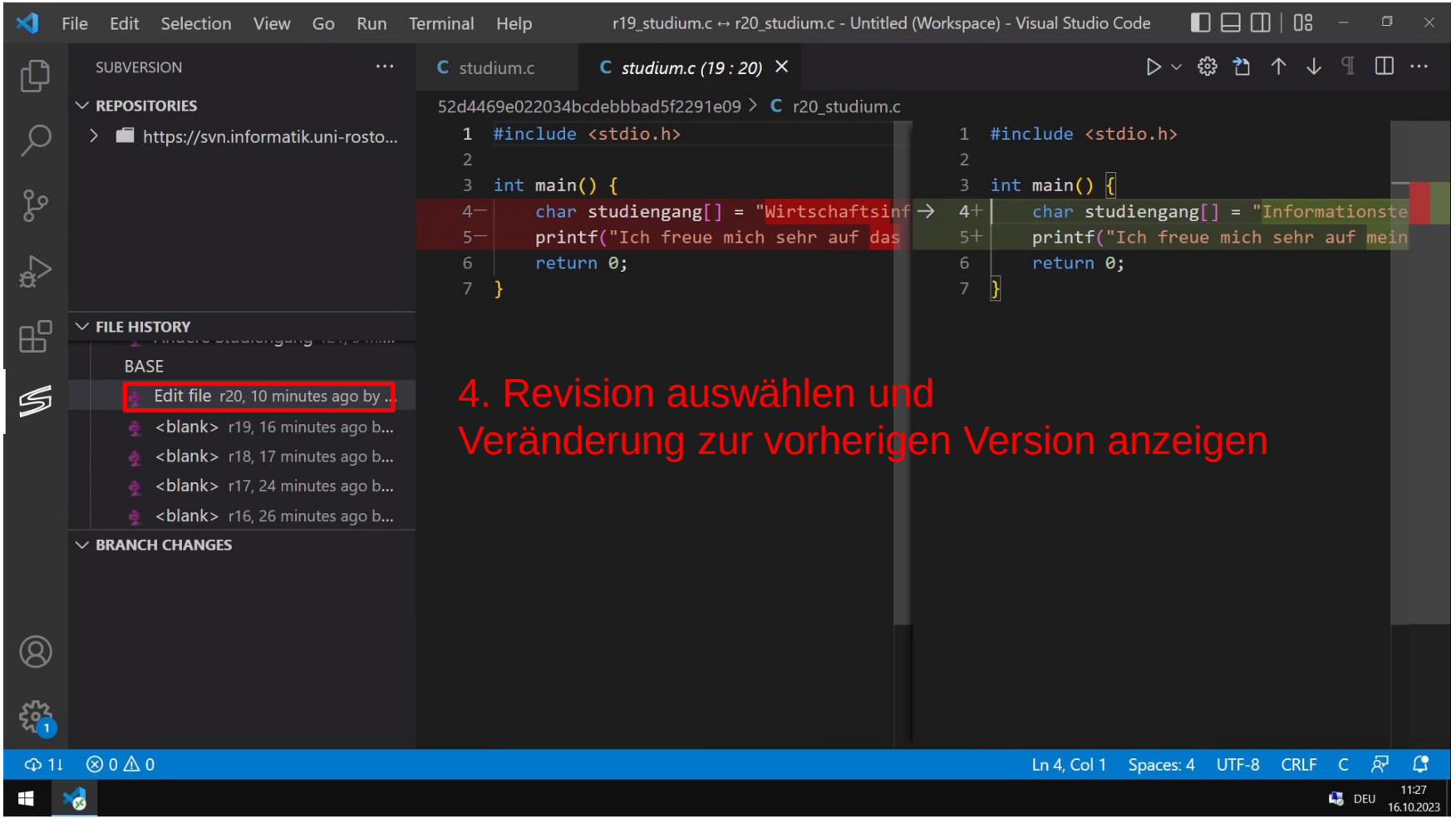
SVN in VS Code



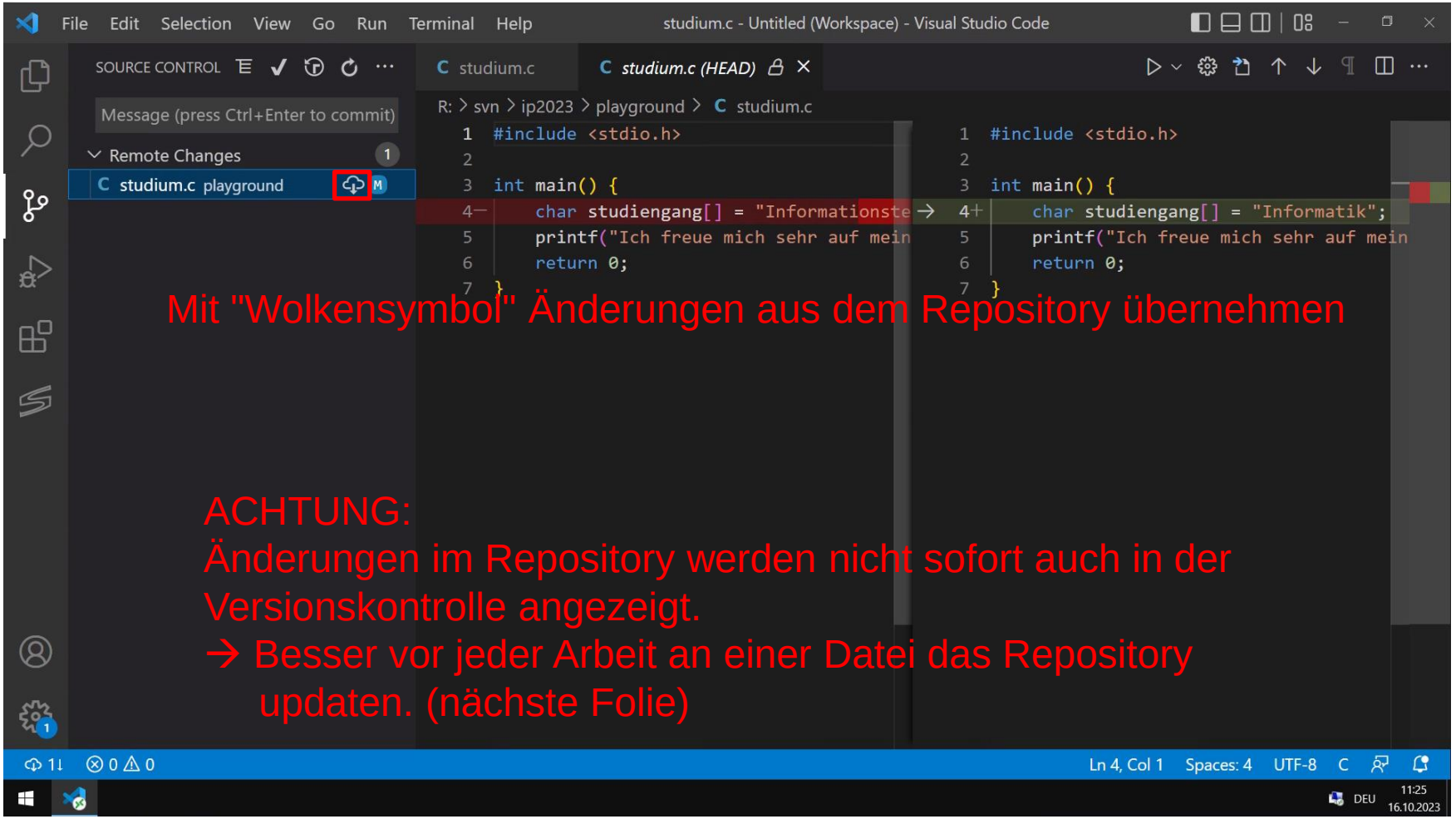
SVN Revisionen



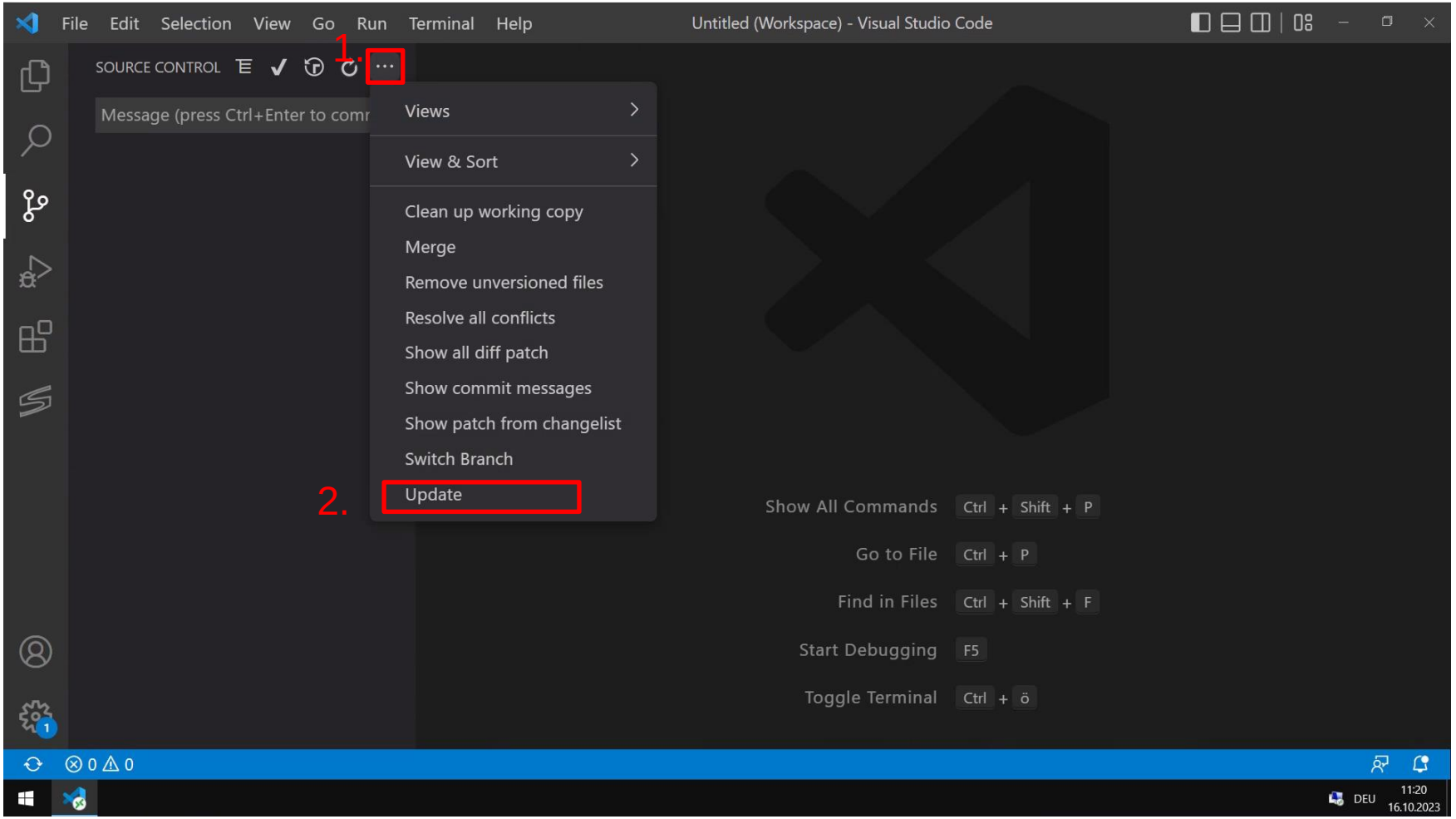
SVN Revisionen



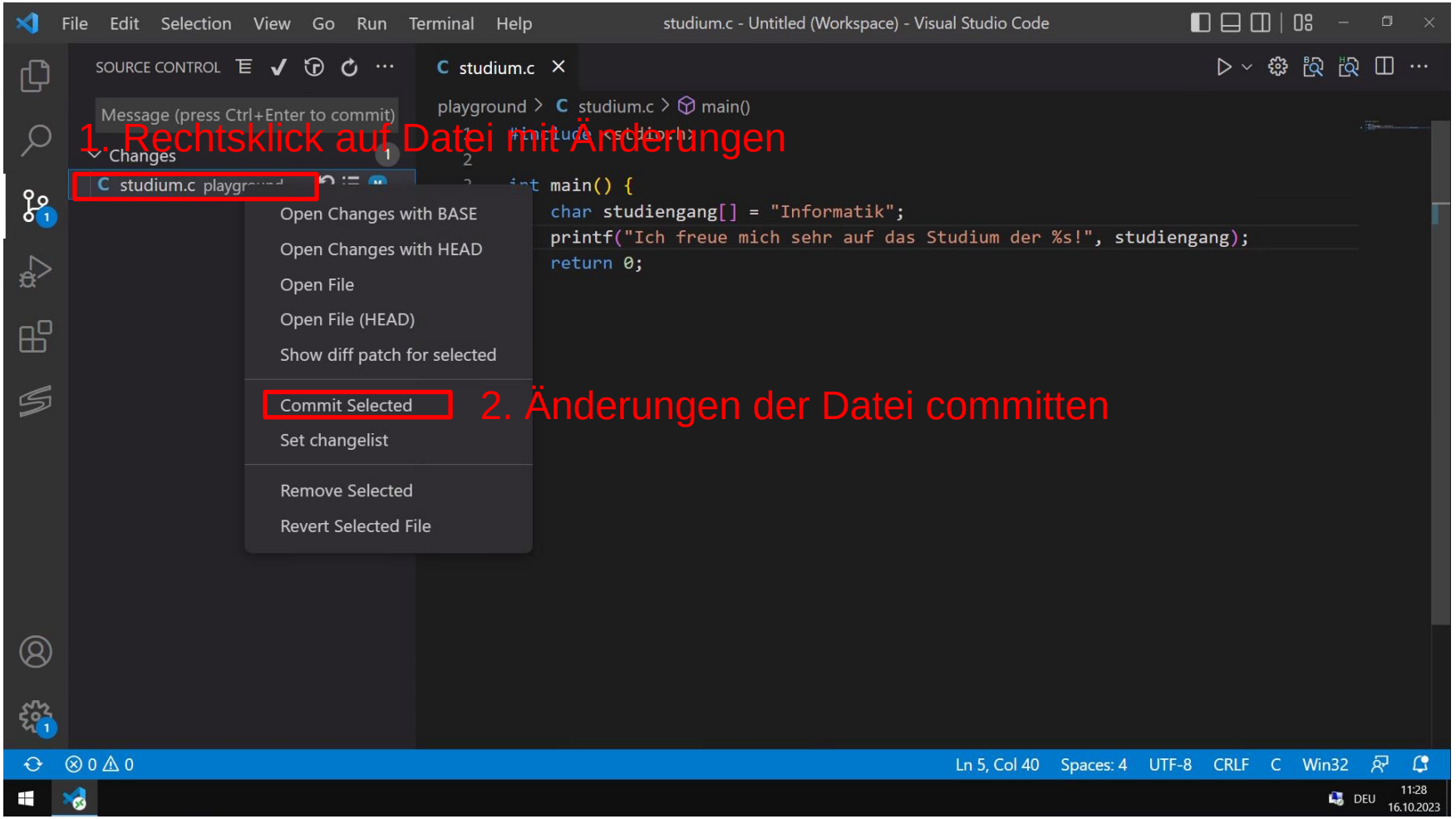
SVN Update


File Edit Selection View Go Run Terminal Help
studium.c - Untitled (Workspace) - Visual Studio Code
SOURCE CONTROL
Message (press Ctrl+Enter to commit)
Remote Changes 1
studium.c playground M
studium.c
studium.c (HEAD)
R: > svn > ip2023 > playground > studium.c
#include <stdio.h>
int main() {
char studiengang[] = "Informationste
printf("Ich freue mich sehr auf mein
return 0;
}
#include <stdio.h>
int main() {
char studiengang[] = "Informatik";
printf("Ich freue mich sehr auf mein
return 0;
}
Mit "Wolkensymbol" Änderungen aus dem Repository übernehmen
ACHTUNG:
Änderungen im Repository werden nicht sofort auch in der Versionskontrolle angezeigt.
→ Besser vor jeder Arbeit an einer Datei das Repository updaten. (nächste Folie)
Ln 4, Col 1 Spaces: 4 UTF-8 C
DEU 11:25 16.10.2023

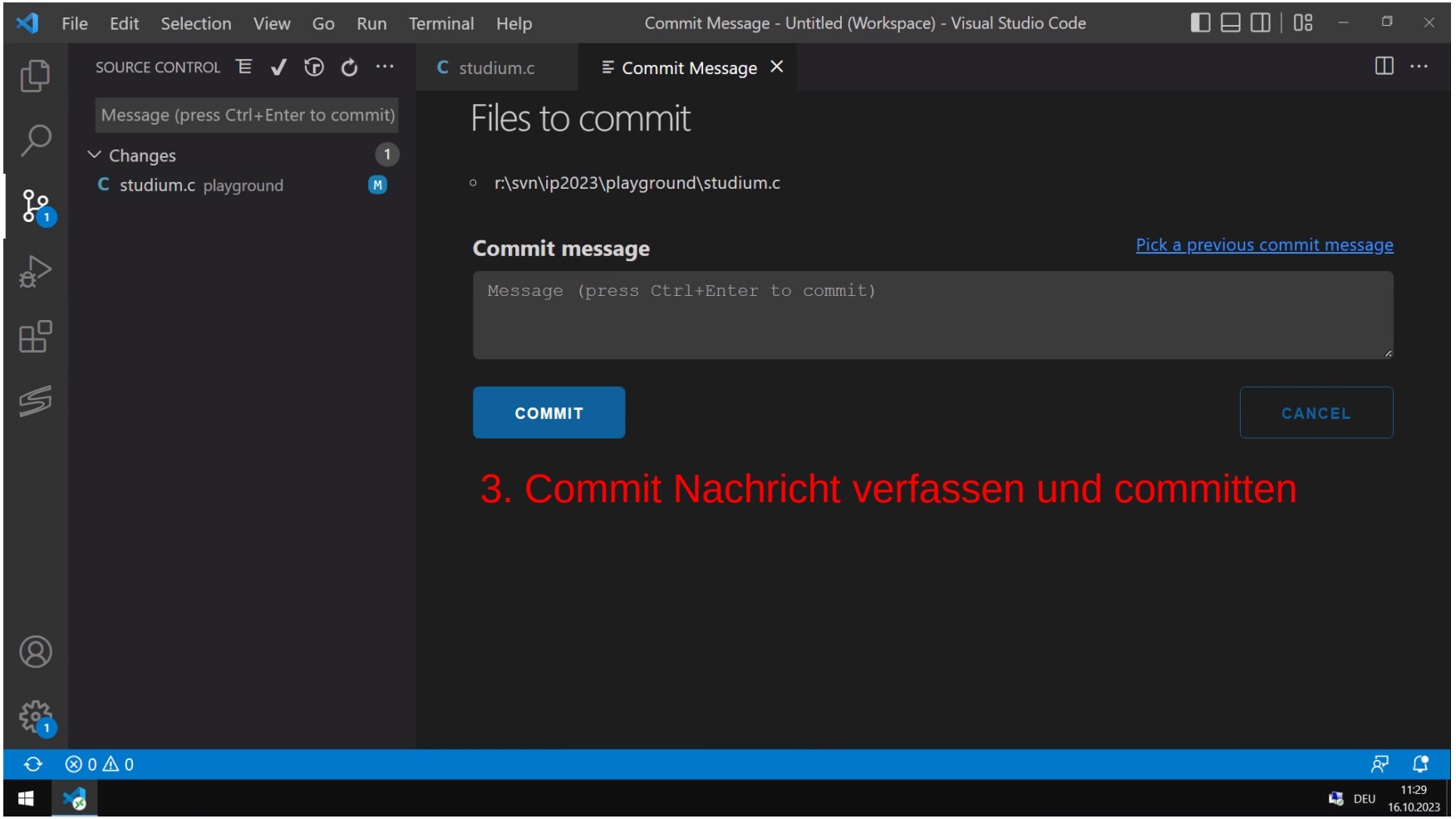
SVN Update ohne angezeigte Änderungen



SVN Commit (lokale Änderungen hochladen)

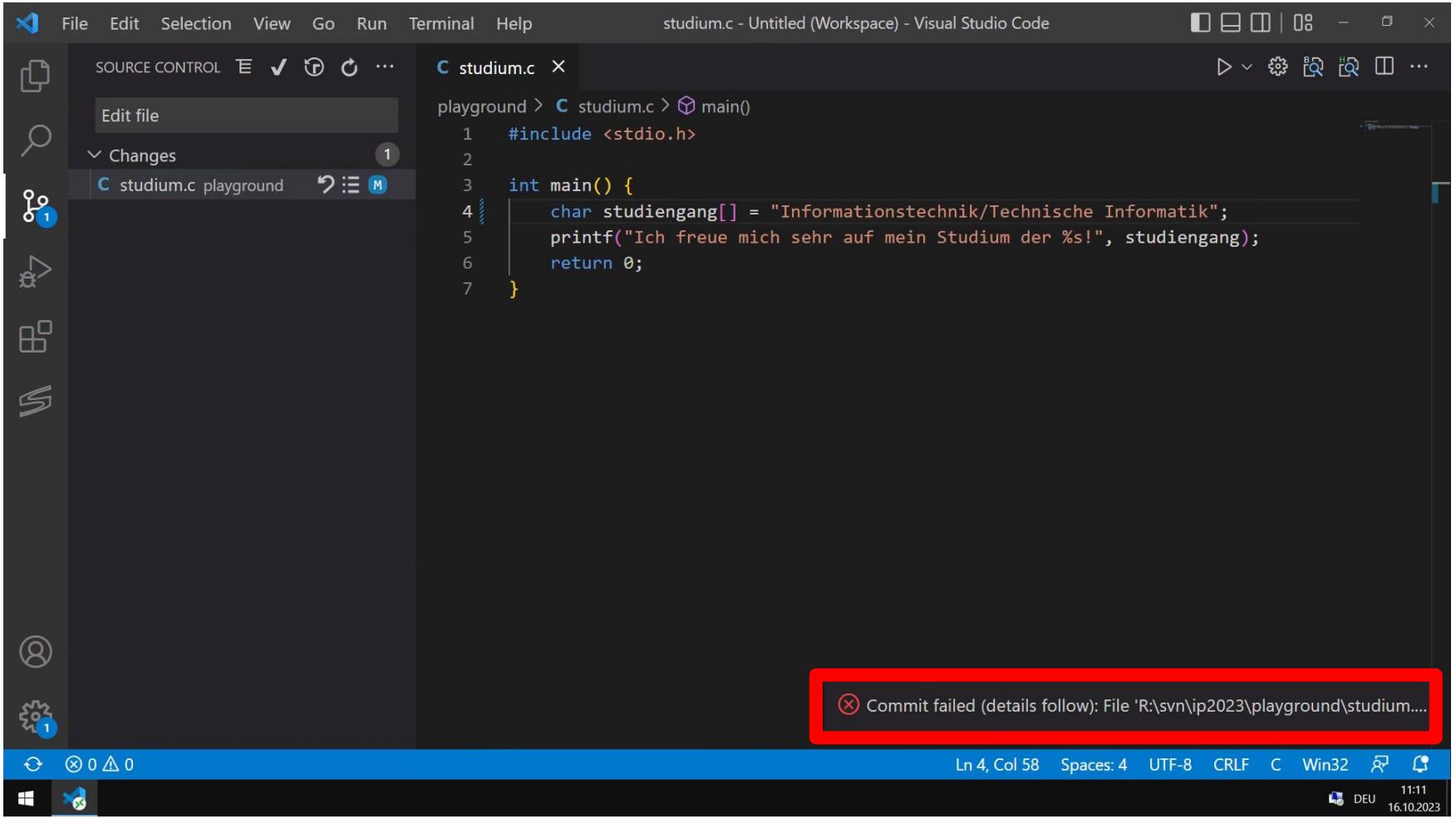


SVN Commit (lokale Änderungen hochladen)

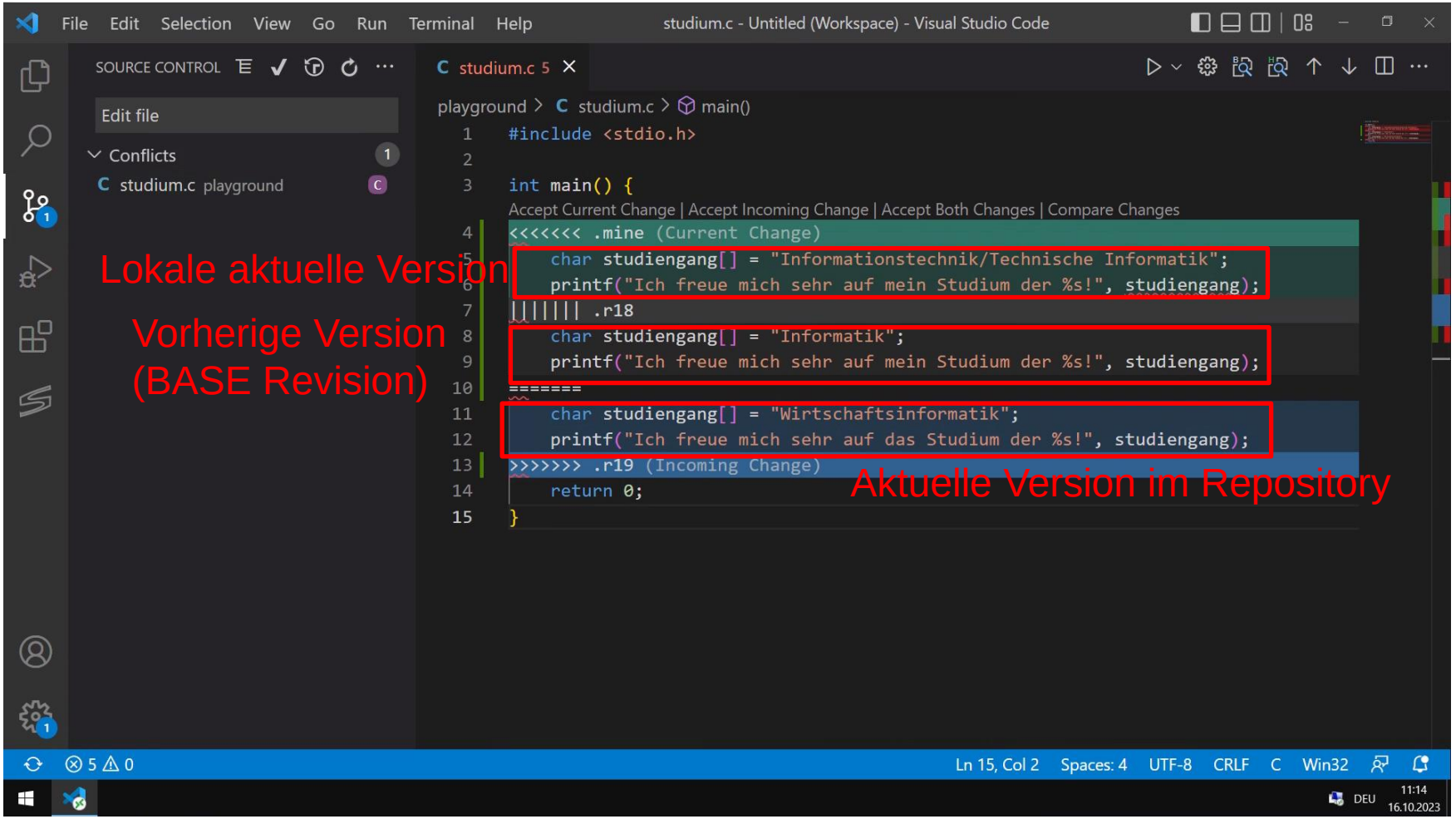


Konflikte lösen

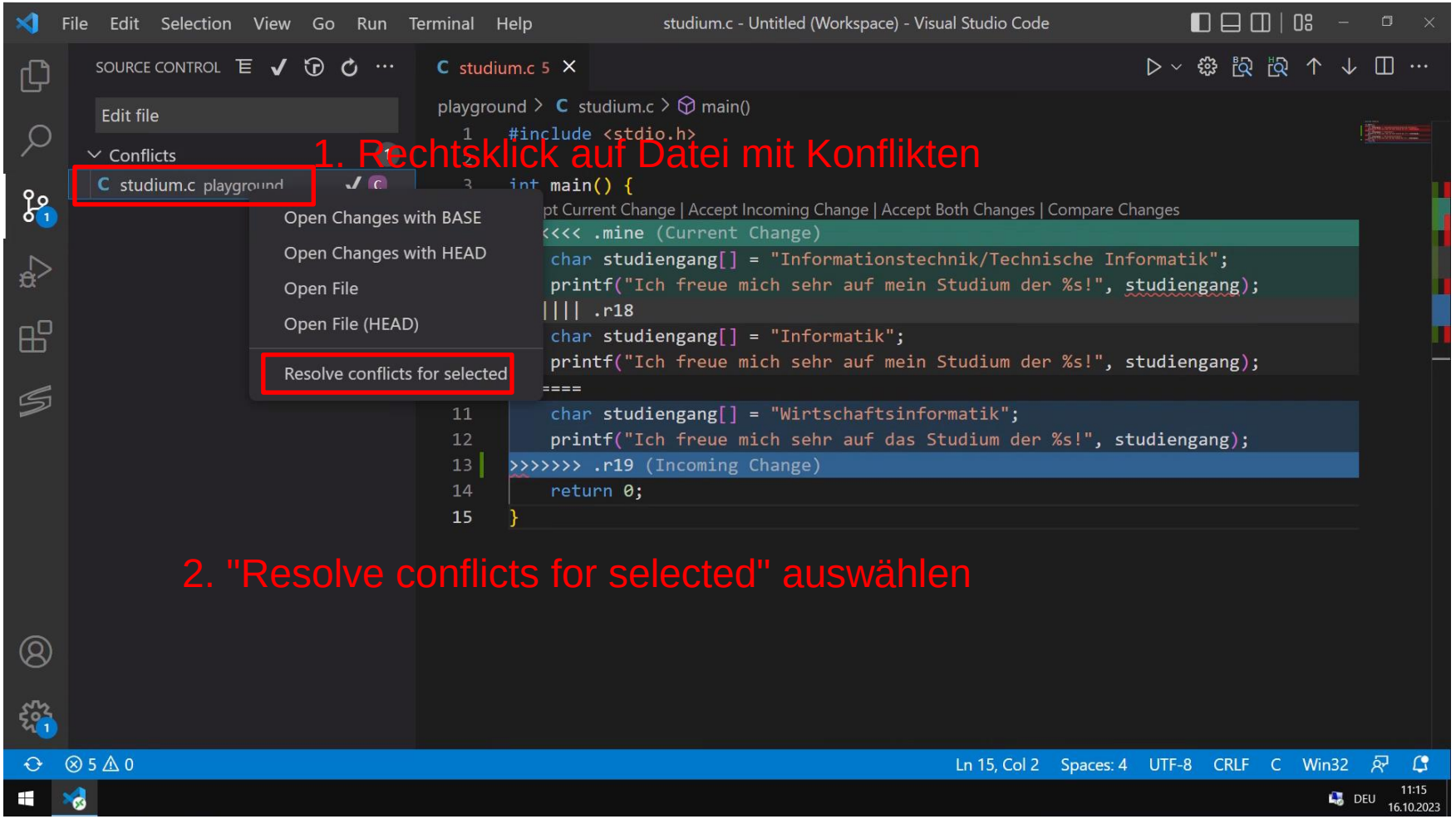
> Falls ein Commit scheitert → Datei updaten



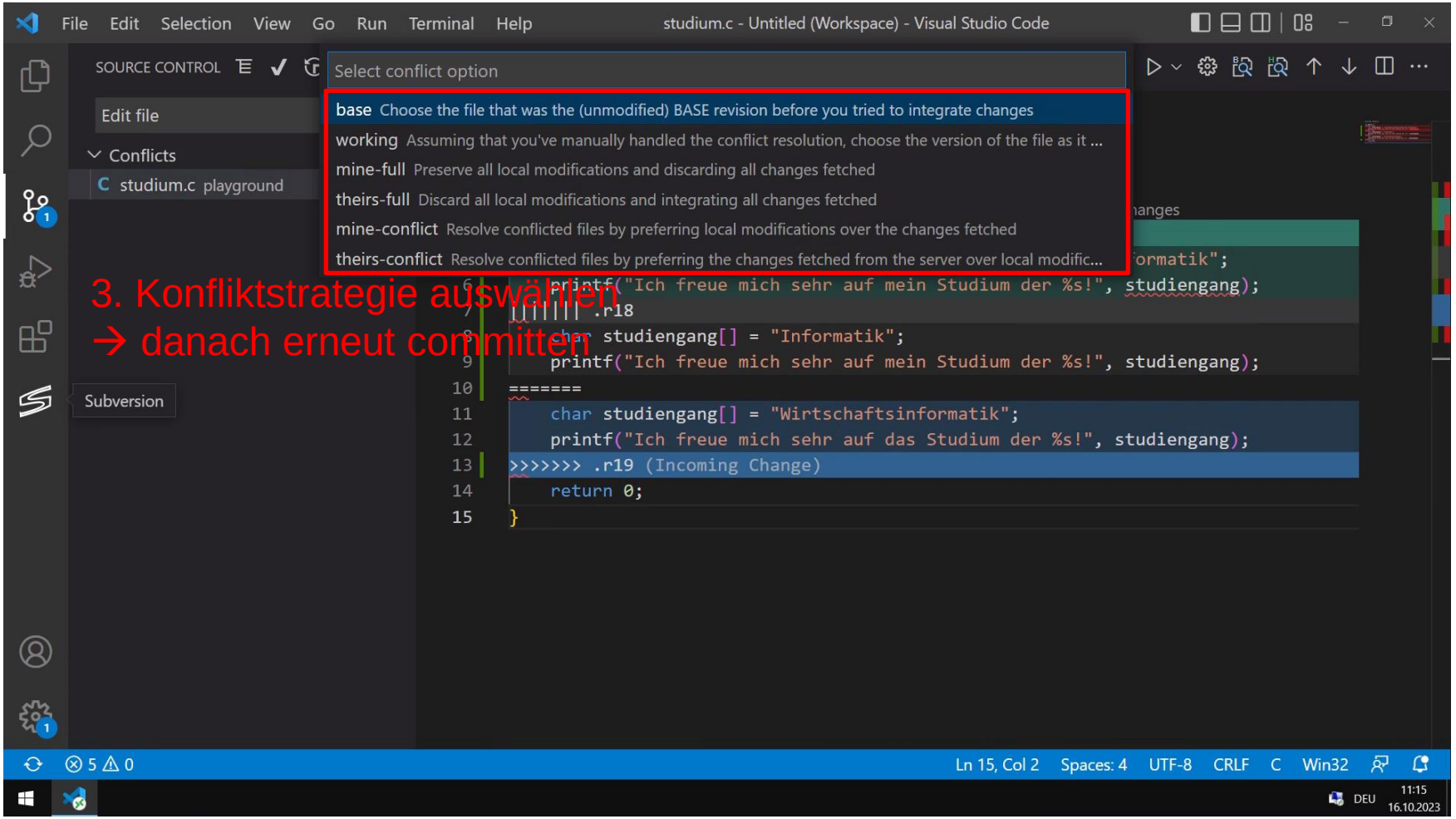
Konflikte lösen



Konflikte lösen



Konflikte lösen



Aufgaben:

- Comitten Sie eine Datei in das playground Repository!
- Lassen Sie die Datei durch einen Mitstudierenden verändern und provozieren Sie einen Konflikt! Lösen Sie den Konflikt.
- Finden Sie heraus welche Änderungen an der Datei studium.c mit der Commit-Nachricht "Kleine Aenderung" vorgenommen wurden.
- Welche Änderungen wurden an der Datei studium.c beim Schritt von Revision 12 zu 13 vorgenommen?

Fragen?

Vielen Dank für die Aufmerksamkeit!