

Praktikumstage bei der msg DAVID GmbH

Mai 2024

Willkommen!

Wir sind ...



Michael Kohrs

Fabian Goethert

Joel-Tyron Beck

Eugen Stepanov

Leon Alexander Widerspan

Fatemeh Rahiminezhad Horand

Jack Kuenstler



Erster Tag



08:45 Uhr – Vorgehensmodelle (Wasserfall und Scrum)



09:15 Uhr - Projektvorstellung: Zeugnismanagement-Software & Arbeiten am Projekt



12:00 Uhr – Rundgang durch das Gebäude



12:45 Uhr - Mittagspause



13:05 Uhr – Vorstellung von Git u. Arbeiten am Projekt



15:15 Uhr - Feierabend



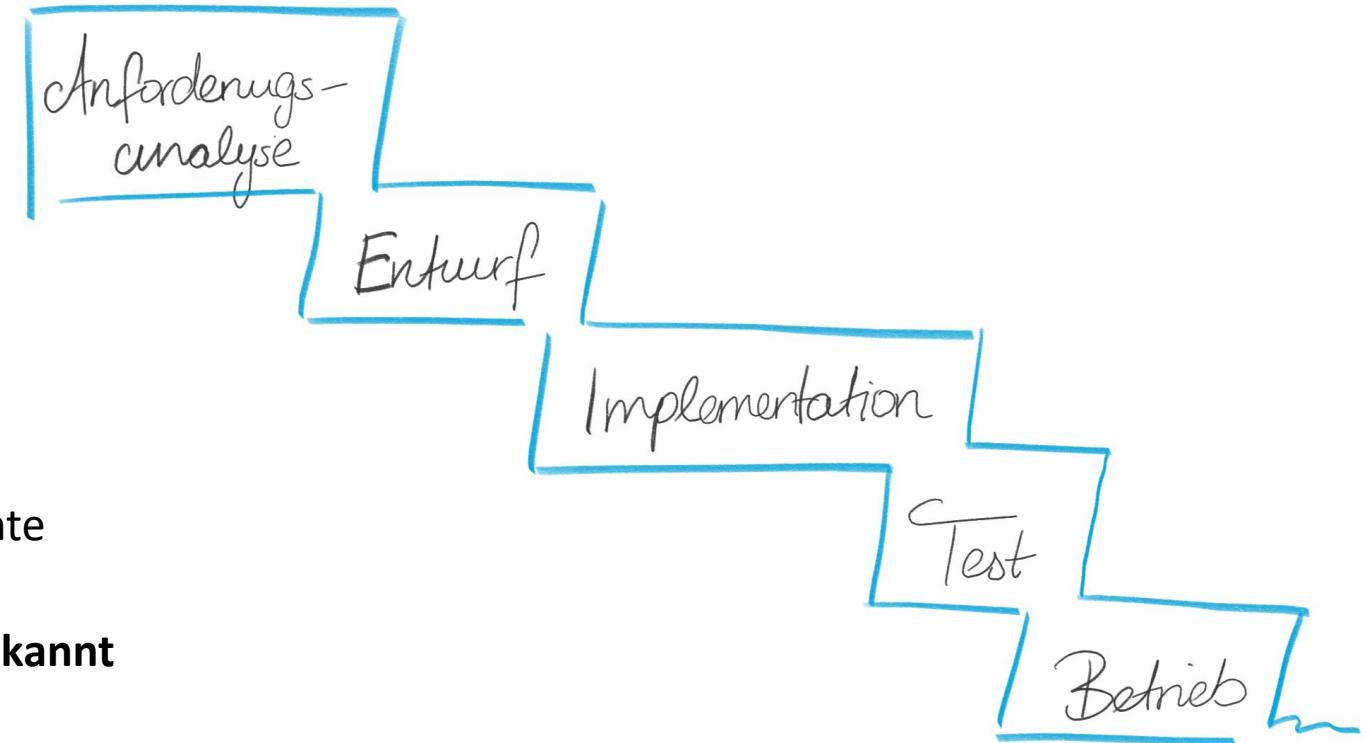
Vorgehensmodelle: Wasserfall & Scrum

Leon & Virginia

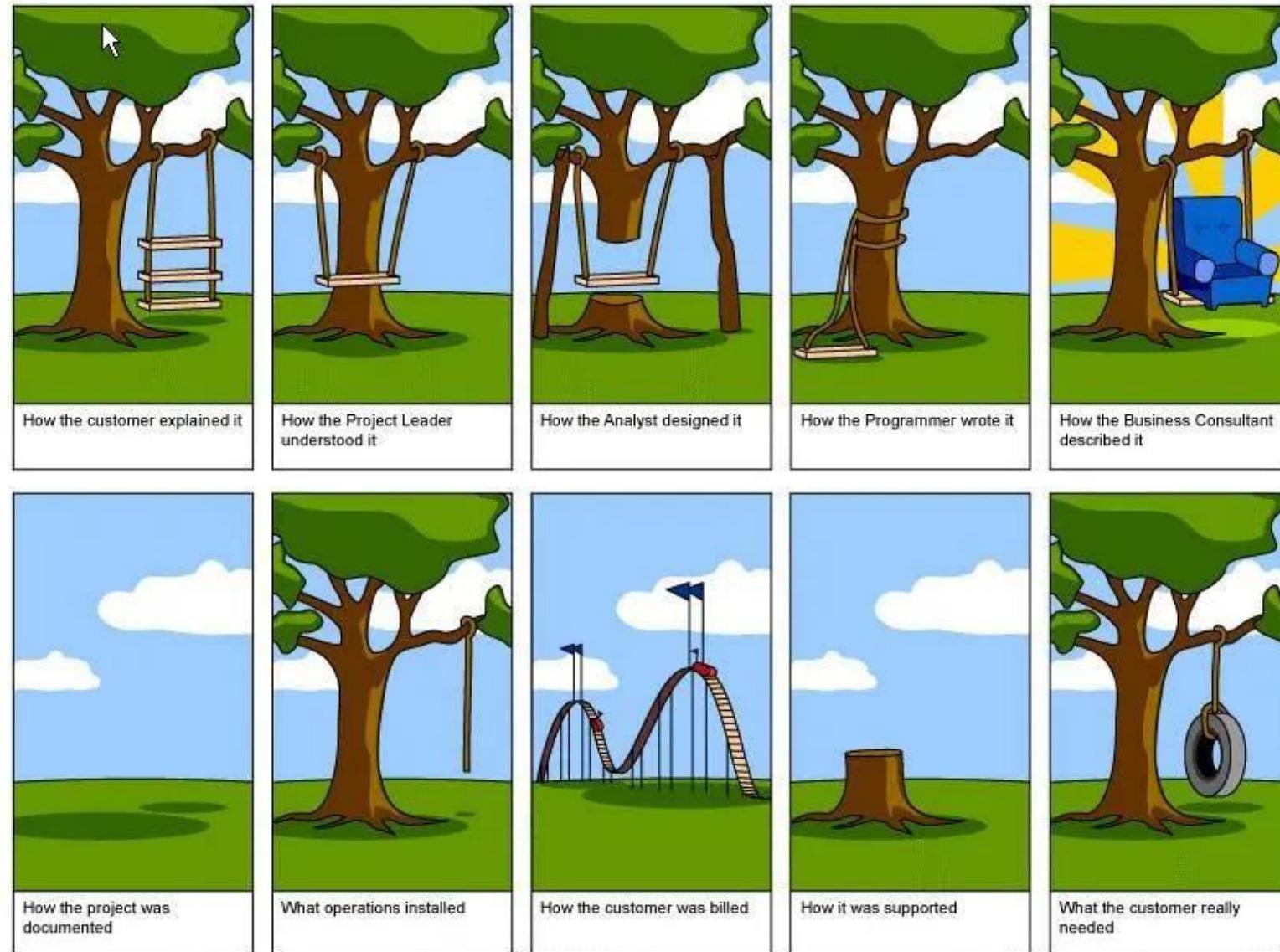


Wasserfallmodell

- **Sequielles Vorgehensmodell** für Software-Entwicklung
- Entworfen in 1956
- Erstes Vorgehensmodell in der Geschichte
- Pro: **Aufgaben vor Implementierung bekannt**
- Con: Sehr **starr**
- Heutzutage selten verwendet



Der Weg zu agilen Modellen ...



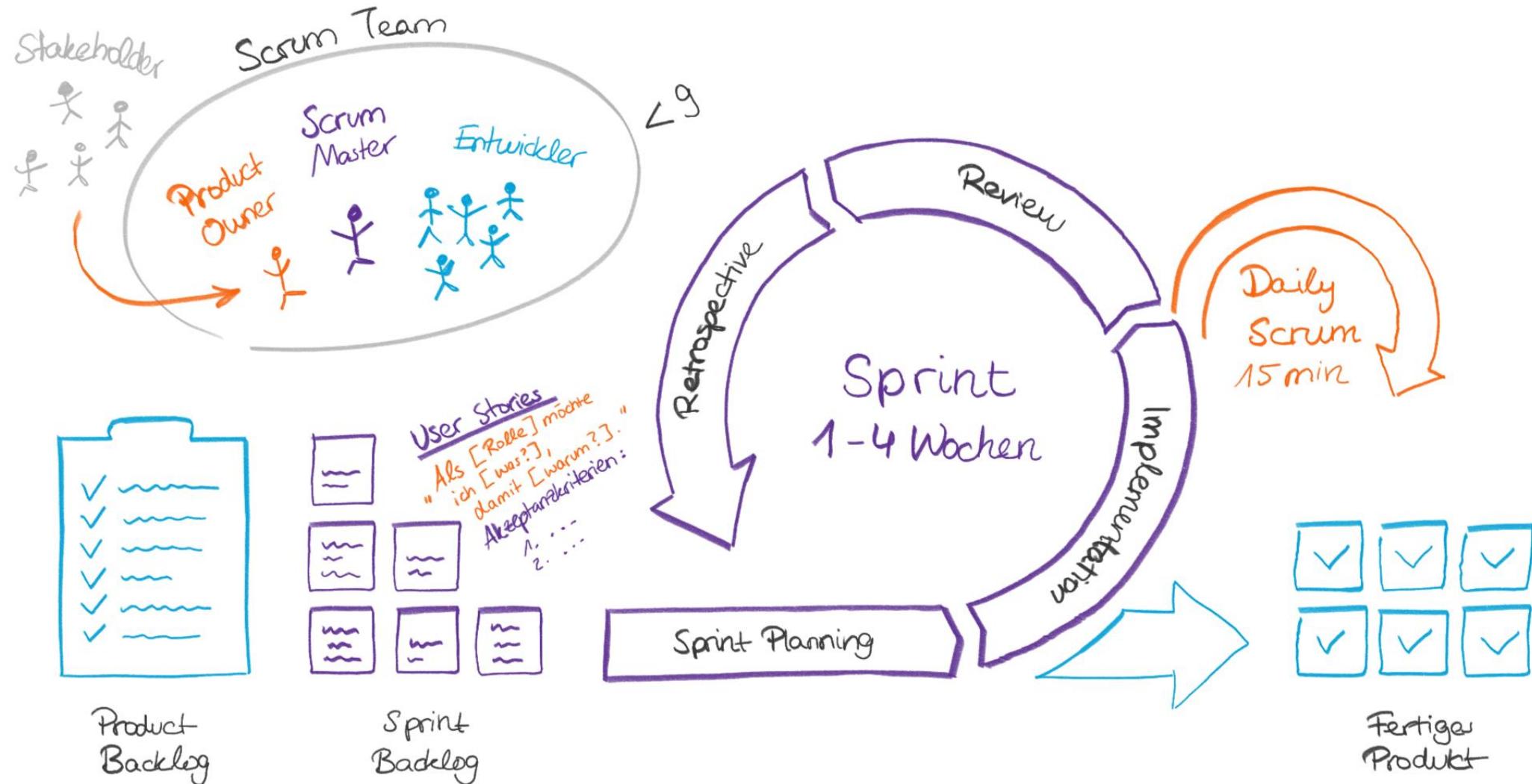
Scrum

- **Agiles Vorgehensmodell** für Software-Entwicklung
- Entworfen in 1995
- Heute: Nicht nur bei Software-Projekten eingesetzt
- **Agil, transparent, fokussiert, kollaborativ, time-boxed**
- Heutzutage sehr häufig verwendet
- Idee
 - **Kurze iterative Entwicklungszyklen (Sprint)**
 - **Häufige Absprachen** mit dem Kunden
 - **Selbstorganisierte Teams**
 - Großes Projekt in **kleine Aufgaben** herunterbrechen
 - **Minimal Viable Product** schaffen und erweitern



<https://www.visual-paradigm.com/tw/scrum/what-is-the-evolution-of-scrum/>

Scrum



Scrum-Rollen

- **Product Owner**
 - Eine einzelne Person
 - Verantwortlich für das Produktziel
 - Verantwortlich für das Product Backlog
 - Meist jemand von der Kundenseite
 - Kennt die Anforderungen
- **Stakeholder**
 - Gehören nicht zum Scrum-Team
 - Haben Interesse am Erfolg des Projekts
 - Anwender, Management, Projektleiter, ...
 - Teilen Anforderungen an Product Owner mit
- **Scrum Master**
 - Eine einzelne Person
 - Agiert als Moderator und vermittelt
 - Stellt sicher, dass Scrum umgesetzt wird
 - Schafft vertrauensvolle Arbeitsatmosphäre
- **Entwickler**
 - Idealerweise fünf Personen
 - Realisieren das Projekt
 - Stellen Projektergebnisse dem PO vor
 - Können beim Schreiben von Items mitwirken
(Ratschläge geben, verfeinern, ...)

Scrum-Elemente

- **Product Backlog**
 - Liste mit Items (User Storys, Bugs, ...)
 - Product Owner pflegt, erweitert und priorisiert
- **Sprint Backlog**
 - Liste mit Items (User Storys, Bugs, ...)
 - **Vom Entwicklungsteam für den kommenden Sprint selektiert**
 - Commitment vom Team (was schaffen wir im nächsten Sprint?)
- **User Story**
 - Eine kleine Aufgabe
 - "**Als [Rolle] möchte ich [was?], damit [warum?].**"
- **Akzeptanzkriterien**
 - Anforderungen an die Umsetzung
 - **Eindeutig** formuliert und **testbar**
 - Item gilt als abgeschlossen, wenn **alle Akzeptanzkriterien** erfüllt sind

Sprint-Elemente

- **Sprint**
 - Ein Entwicklungszyklus
 - Dauer: 1-4 Wochen
 - Beinhaltet Entwicklungszeit und feste Meetings
 - Geschützter Rahmen: **Keine Änderung des Sprint Backlogs, Kunde soll nicht eingreifen**
- **Sprint Planning**
 - Meeting zu Beginn des Sprints
 - Items werden für Sprint **gewählt**
 - Aufwand pro Item wird **geschätzt** (Komplexität)
 - **Arbeitsschritte werden diskutiert**
- **Daily Scrum**
 - Tägliches 15-minütiges Stand-up-Meeting
 - **Statusupdate und Plan**, was heute gemacht wird
- **Sprint Review**
 - Meeting zum Ende des Sprints
 - **Ergebnisse** des Sprints werden dem Product Owner und Stakeholdern **vorgestellt**
- **Retrospektive**
 - Meeting zum Ende des Sprints
 - Was lief gut/schlecht? Was wollen wir ändern?

Erster Tag



08:45 Uhr – Vorgehensmodelle (Wasserfall und Scrum)



09:15 Uhr - Projektvorstellung: Zeugnismanagement-Software & Arbeiten am Projekt



12:00 Uhr – Rundgang durch das Gebäude



12:45 Uhr - Mittagspause



13:05 Uhr – Vorstellung von Git u. Arbeiten am Projekt



15:15 Uhr - Feierabend



Pause



Erster Tag



08:45 Uhr – Vorgehensmodelle (Wasserfall und Scrum)



09:15 Uhr - Projektvorstellung: Zeugnismanagement-Software & Arbeiten am Projekt



12:00 Uhr – Rundgang durch das Gebäude



12:45 Uhr - Mittagspause



13:05 Uhr – Vorstellung von Git u. Arbeiten am Projekt

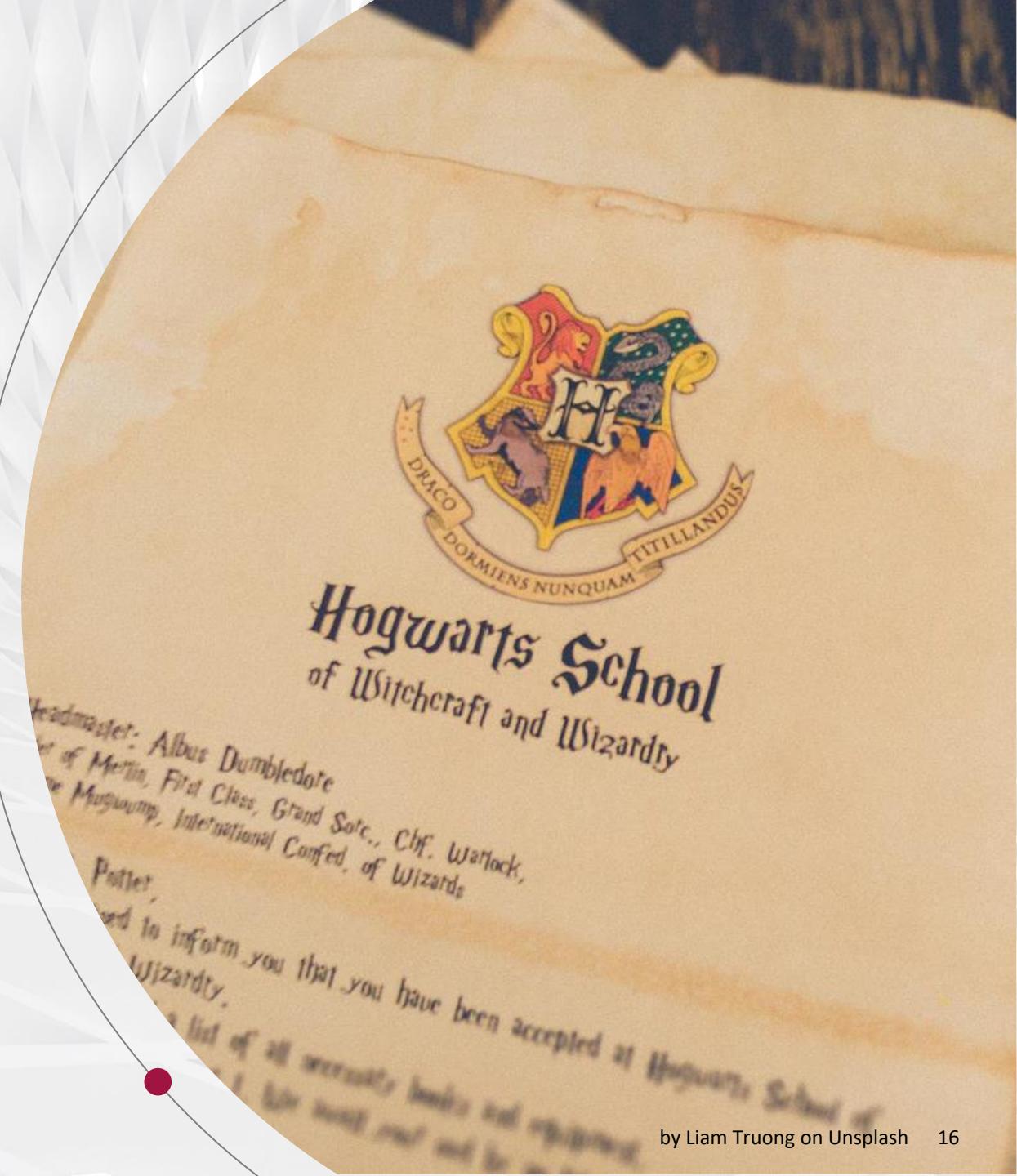


15:15 Uhr - Feierabend



Projektvorstellung

Fabian Goethert

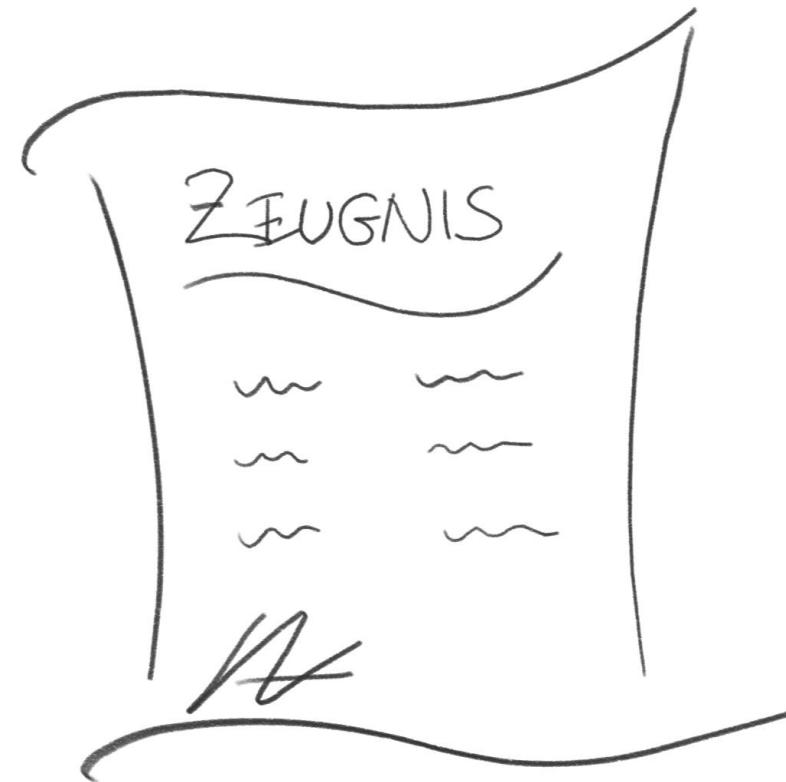


Zeugnismanagement-Software

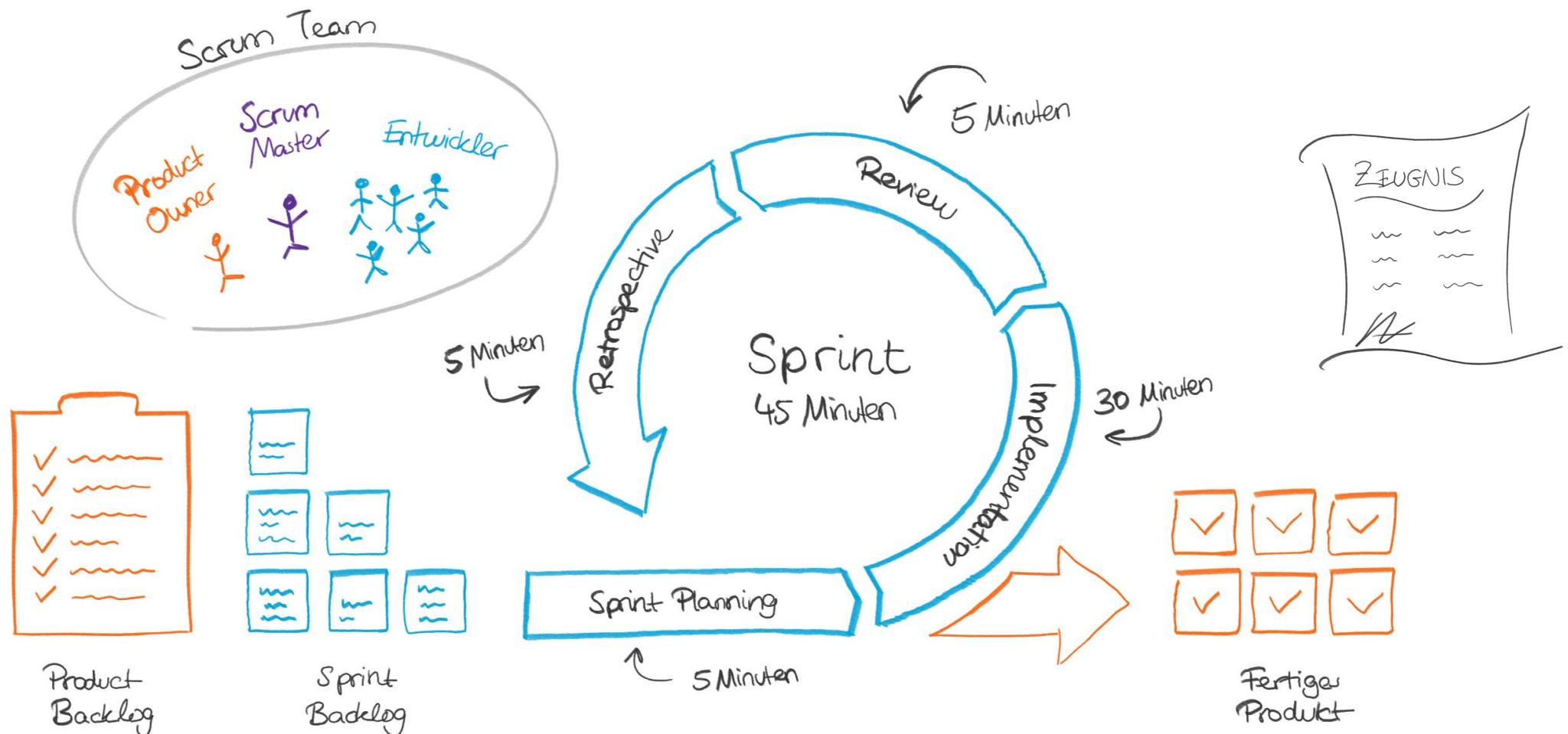
Aufgabenstellung

Das David-Gymnasium hat Sie beauftragt, eine Zeugnismanagement-Software zu entwickeln, mit deren Hilfe ein Zeugnis für einen Schüler erstellt werden kann.

Dazu sollen die erforderlichen Daten manuell über eine Konsole eingelesen werden, mit deren Hilfe die Software automatisch analysiert, ob der betreffende Schüler versetzt wird.



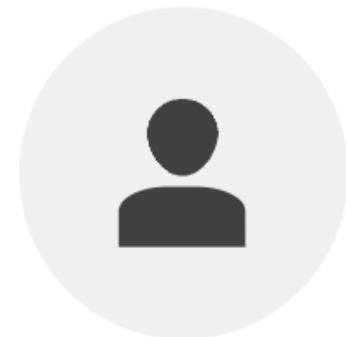
Unser Scrum-Vorgehen



Brainstorming über mögliche Aufgaben

Eure Ideen?

<https://bit.ly/praktikumstage>



Anmelden



Gastzugang

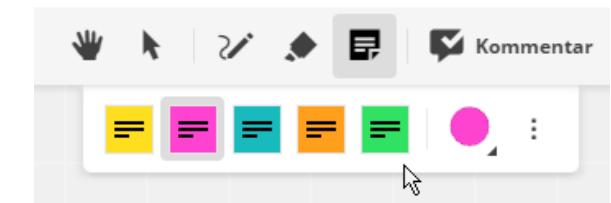
Als Guest beitreten

Shark

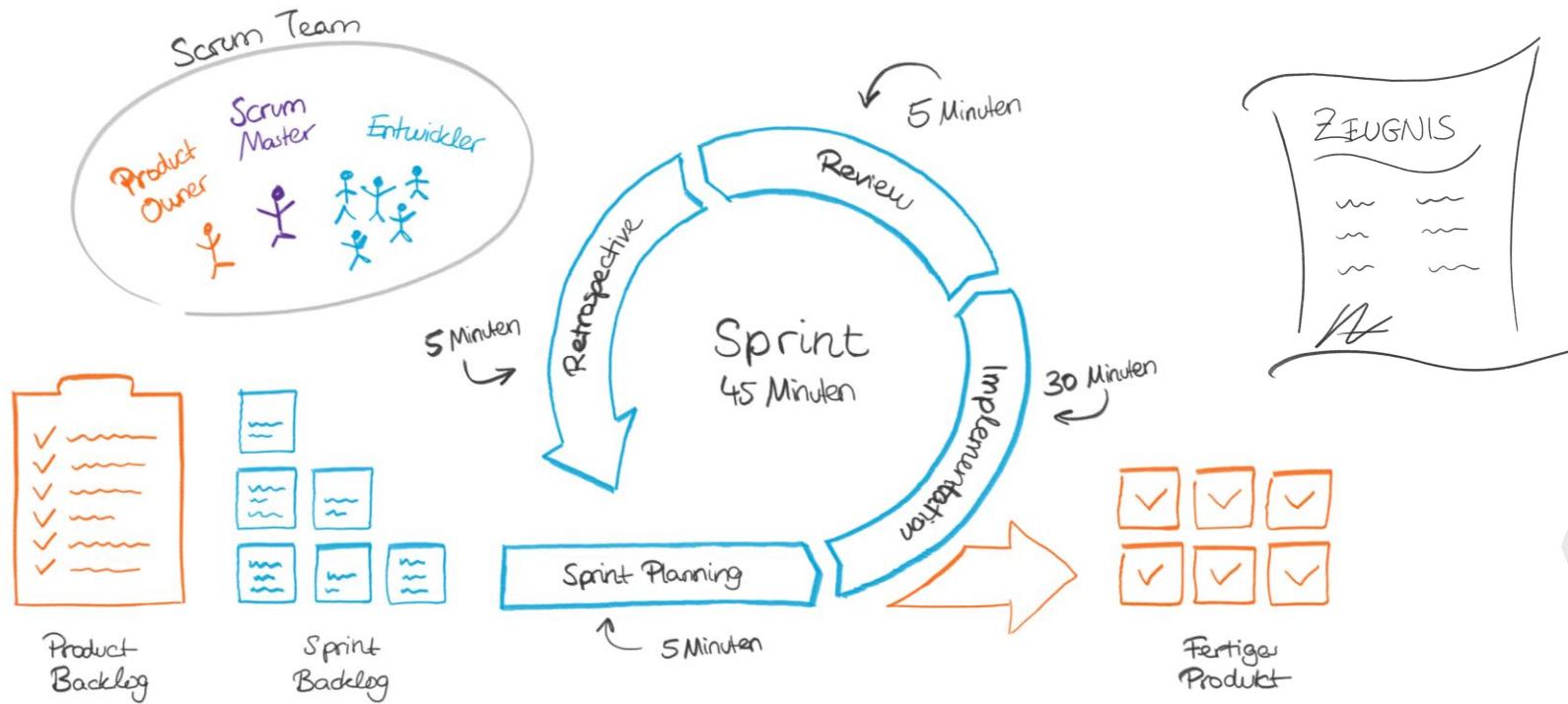
Ich habe die Datenschutzerklärung zur Kenntnis genommen.

Als Guest mitarbeiten

[zurück](#)



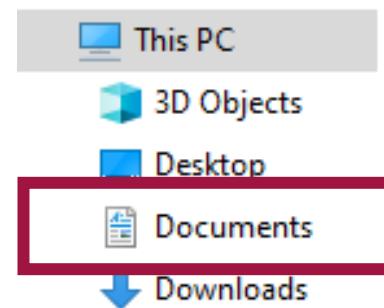
Projektdurchführung



User: Schulungsteilnehmer

Passwort: Schulung1

Projekt-Ort:



Aufgabenstellung

Das David-Gymnasium hat Sie beauftragt, eine Zeugnismanagement-Software zu entwickeln, mit deren Hilfe ein Zeugnis für einen Schüler erstellt werden kann.

Dazu sollen die erforderlichen Daten manuell über eine Konsole eingelesen werden, mit deren Hilfe die Software automatisch analysiert, ob der betreffende Schüler versetzt wird.

WLAN Verbindung



Gastzugang zum Internet innerhalb der Besprechungsräume der msg DAVID GmbH erhalten Sie über WLAN mit den folgenden Zugangsdaten:



SSID: **msgDAVID-internet**
PASSWORD: **Intrnt4Vstrs2018!**



msg DAVID WLAN-Zugang

Behilflich beim Einrichten der Verbindung ist nebenstehender QR-Code



Sollten Sie Unterstützung bei der Einrichtung benötigen, stehen wir Ihnen unter der Rufnummer 0531 24379111 gerne zur Verfügung.

Folgende Dienste werden unterstützt:

- http, https Webservices
- pop3s, imaps, smtpe E-Mail Dienste

Erster Tag



08:45 Uhr – Vorgehensmodelle (Wasserfall und Scrum)



09:15 Uhr - Projektvorstellung: Zeugnismanagement-Software & Arbeiten am Projekt



12:00 Uhr – Rundgang durch das Gebäude



12:45 Uhr - Mittagspause



13:05 Uhr – Vorstellung von Git u. Arbeiten am Projekt



15:15 Uhr - Feierabend



Mittagspause



Erster Tag



08:45 Uhr – Vorgehensmodelle (Wasserfall und Scrum)



09:15 Uhr - Projektvorstellung: Zeugnismanagement-Software & Arbeiten am Projekt



12:00 Uhr – Rundgang durch das Gebäude



12:45 Uhr - Mittagspause



13:05 Uhr – Vorstellung von Git u. Arbeiten am Projekt



15:15 Uhr - Feierabend



Versionskontrolle: Git

Eugen Stepanov



Was ist Git?

- **Version Control System**
- **Trackt und speichert Veränderungen an Dateien**
- **Vereinfacht Team-Arbeit**
- Veröffentlicht 2005
- Kommandozeilenbasiertes Tool
- Umfangreiches Set an Kommandos (über 100!)
- **Flexibel und leicht erlernbar**
- Für **jede Programmiersprache** einsetzbar
- Aber auch für mehr: Texte, (Bilder), ...
- Aktuell **State of the Art**
- Alternativen: Subversion (SVN), Mercurial, ...



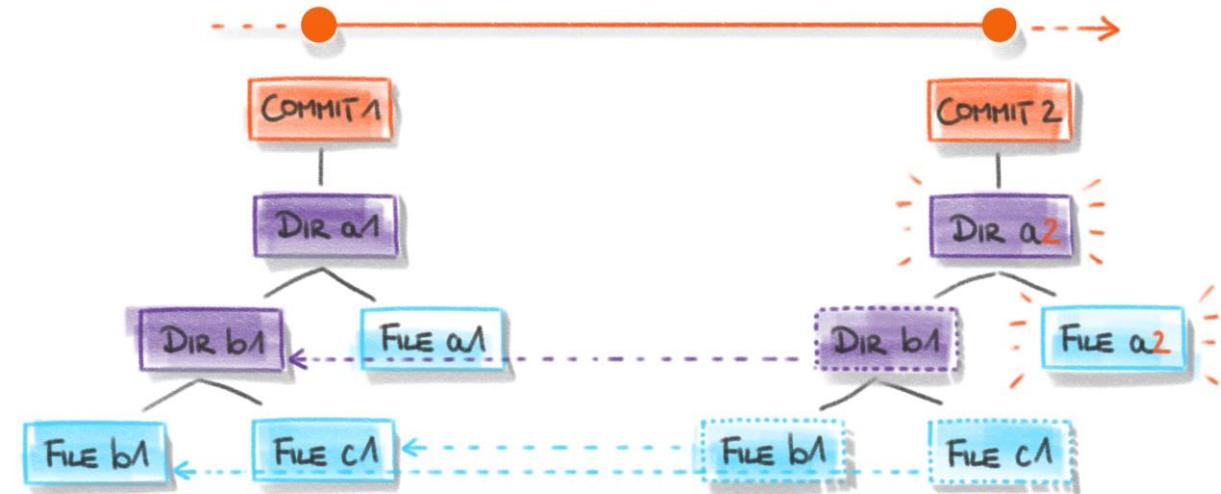
<https://xkcd.com/1597/>

Die wichtigsten Konzepte

- **Version Control System**
 - System zum Speichern von Veränderungen
 - Innerhalb eines bestimmten Ordners (**Repository**)
 - Erkennt automatisch Veränderungen

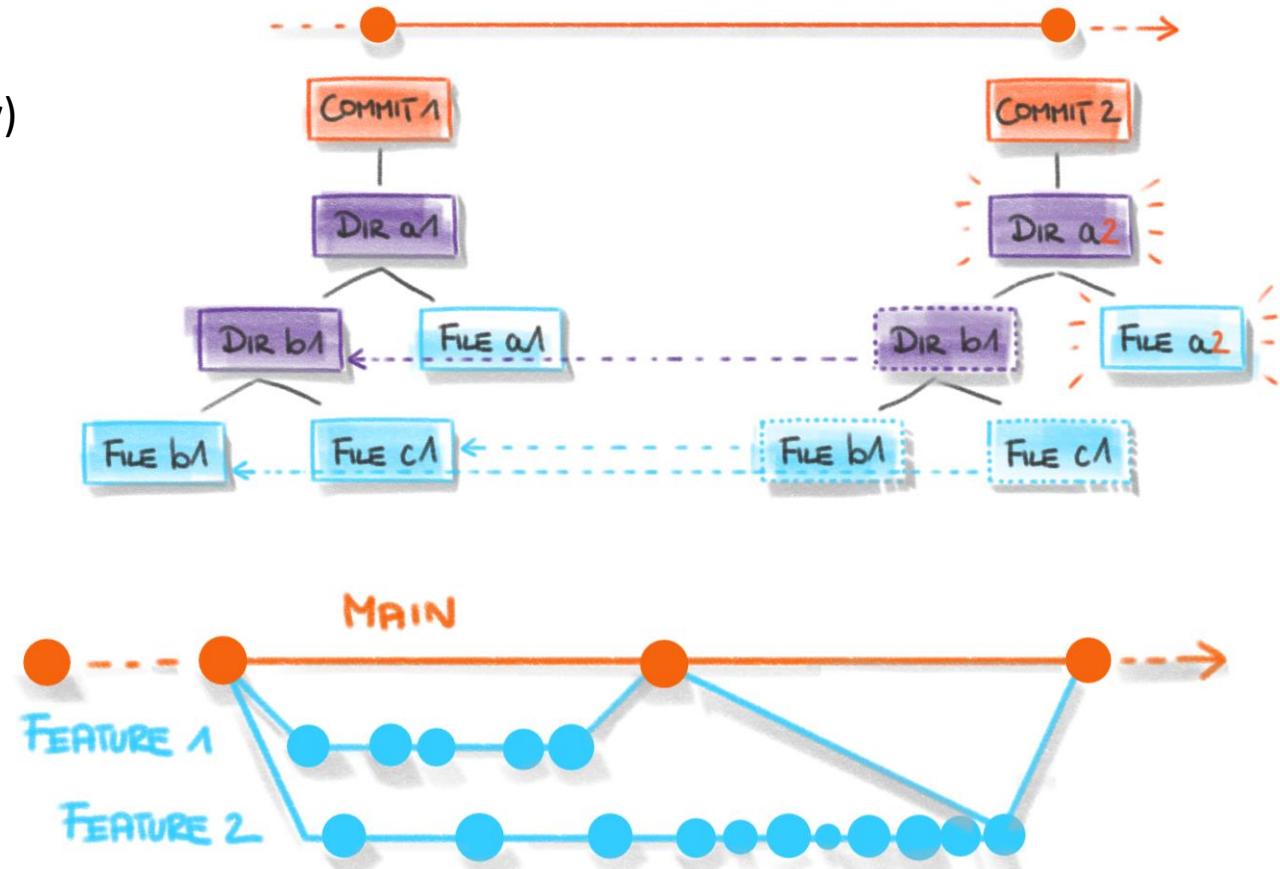
Die wichtigsten Konzepte

- **Version Control System**
 - System zum Speichern von Veränderungen
 - Innerhalb eines bestimmten Ordners (**Repository**)
 - Erkennt automatisch Veränderungen
- **Commit**
 - **Snapshot** mit hinzugefügten Änderungen (**Delta**)
 - Plus Metadaten wie z.B. Zeitstempel, Autor, eindeutige Hash-ID, Commit-Nachricht, ...

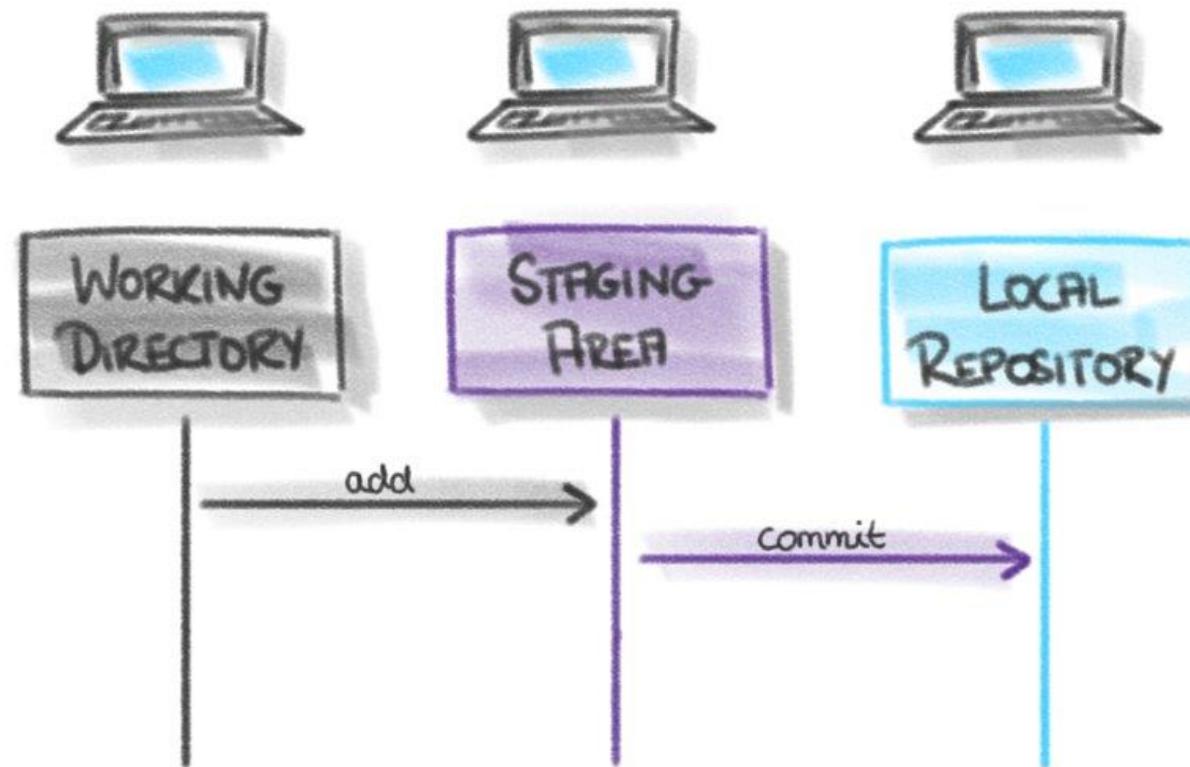


Die wichtigsten Konzepte

- **Version Control System**
 - System zum Speichern von Veränderungen
 - Innerhalb eines bestimmten Ordners (**Repository**)
 - Erkennt automatisch Veränderungen
- **Commit**
 - **Snapshot** mit hinzugefügten Änderungen (**Delta**)
 - Plus Metadaten wie z.B. Zeitstempel, Autor, eindeutige Hash-ID, Commit-Nachricht, ...
- **Branch**
 - Commits auf Branches chronologisch sortiert
 - Mehrere Branches möglich
 - Heute: Nur Haupt-Branch (**main**)



Die wichtigsten Kommandos für heute und morgen (Part 1)



Von Kommandozeilen, GUIs und Webanwendungen

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18362.1139]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\lisar\Projects\foo>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
      modified:   lorem.txt

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\lisar\Projects\foo>git add .

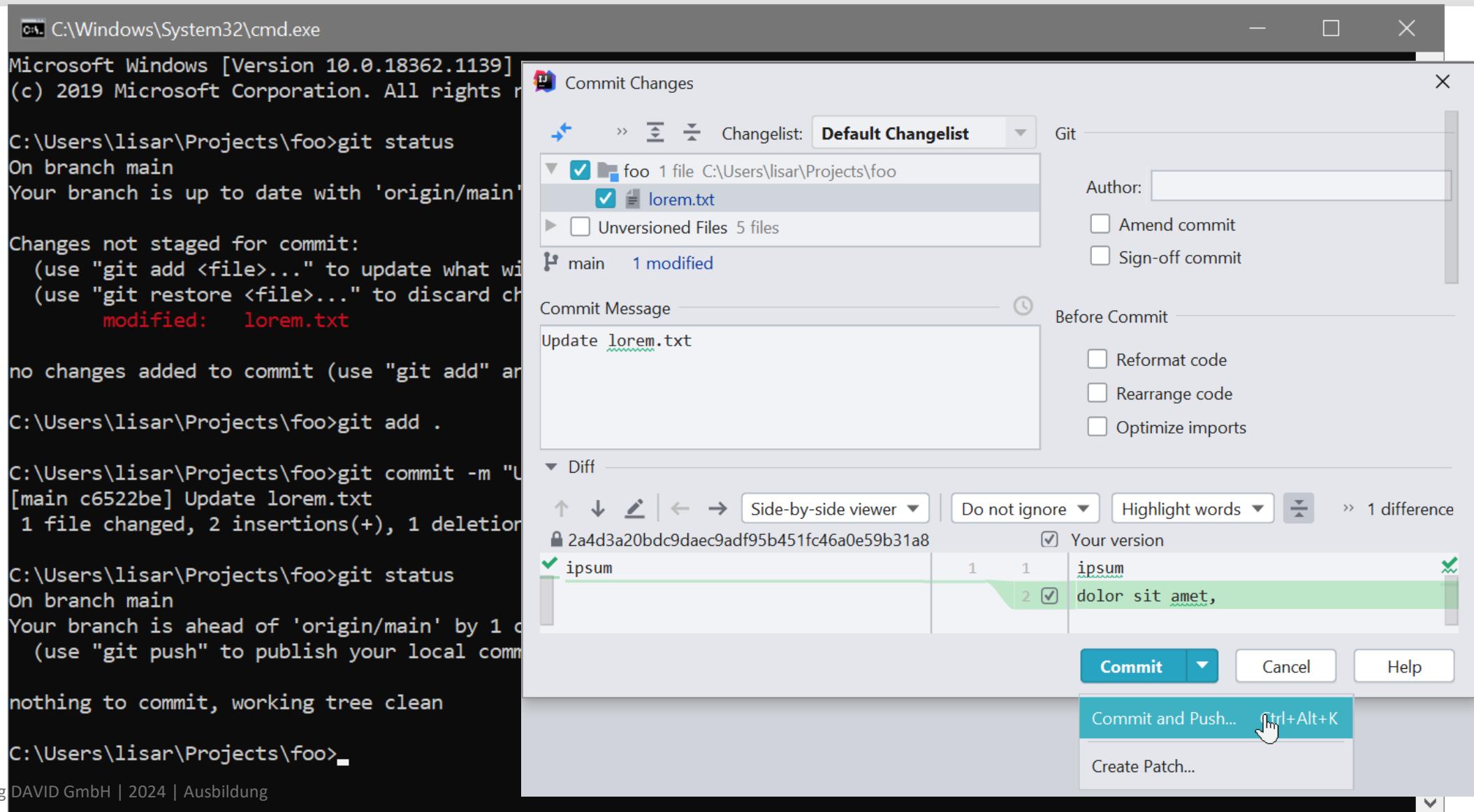
C:\Users\lisar\Projects\foo>git commit -m "Update lorem.txt"
[main c6522be] Update lorem.txt
 1 file changed, 2 insertions(+), 1 deletion(-)

C:\Users\lisar\Projects\foo>git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\Users\lisar\Projects\foo>
```

Von Kommandozeilen, GUIs und Webanwendungen



The screenshot illustrates the workflow of committing changes in Git, showing both a command-line interface and a graphical user interface (GUI) side-by-side.

Windows Command Prompt (cmd.exe):

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18362.1139]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\lisar\Projects\foo>git status
On branch main
Your branch is up to date with 'origin/main'

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
    (use "git restore <file>..." to discard changes in working directory)
          modified:   lorem.txt

no changes added to commit (use "git add" and "git commit" as needed)

C:\Users\lisar\Projects\foo>git add .

C:\Users\lisar\Projects\foo>git commit -m "Update lorem.txt"
[main c6522be] Update lorem.txt
 1 file changed, 2 insertions(+), 1 deletion(-)

C:\Users\lisar\Projects\foo>git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\Users\lisar\Projects\foo>
```

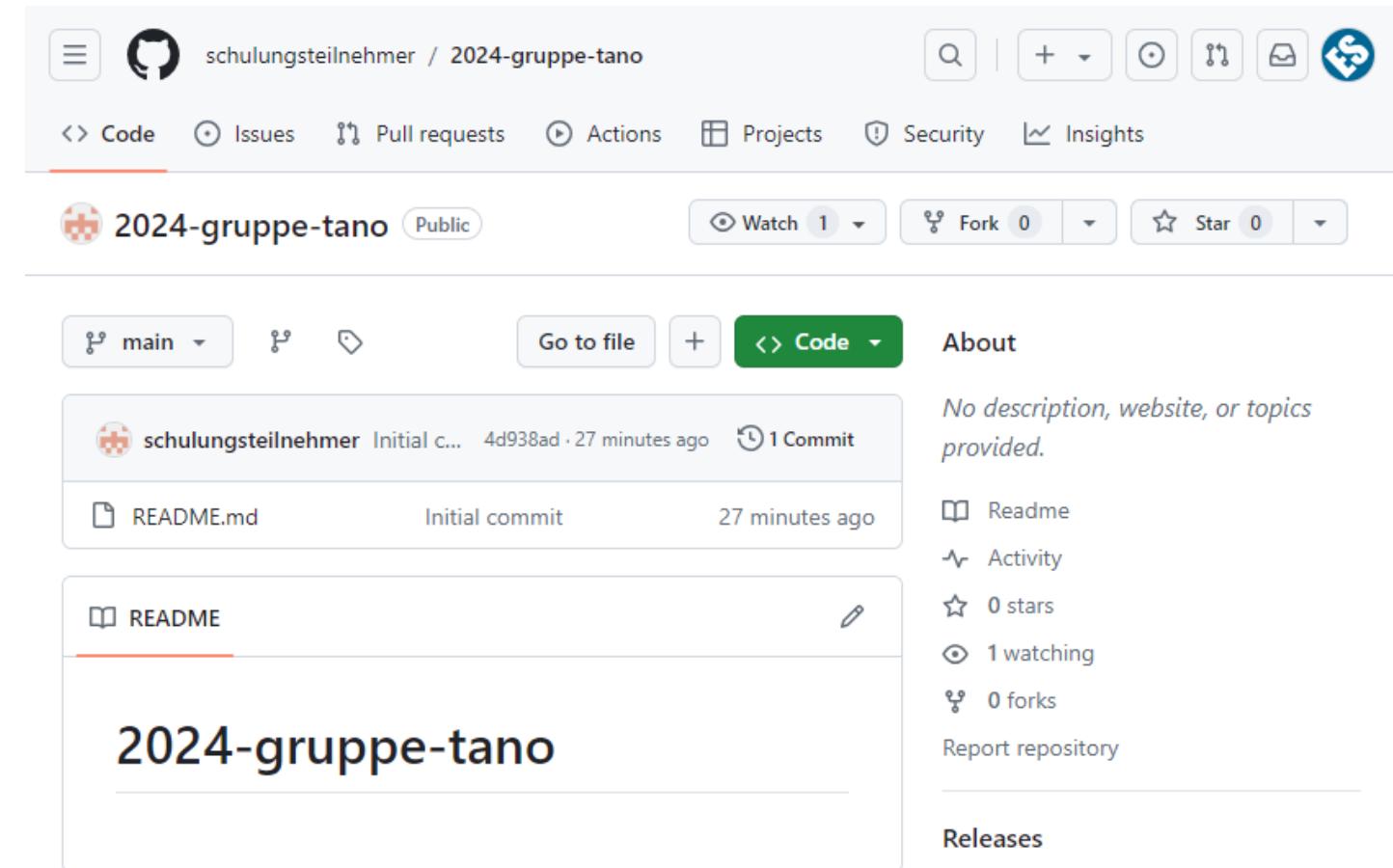
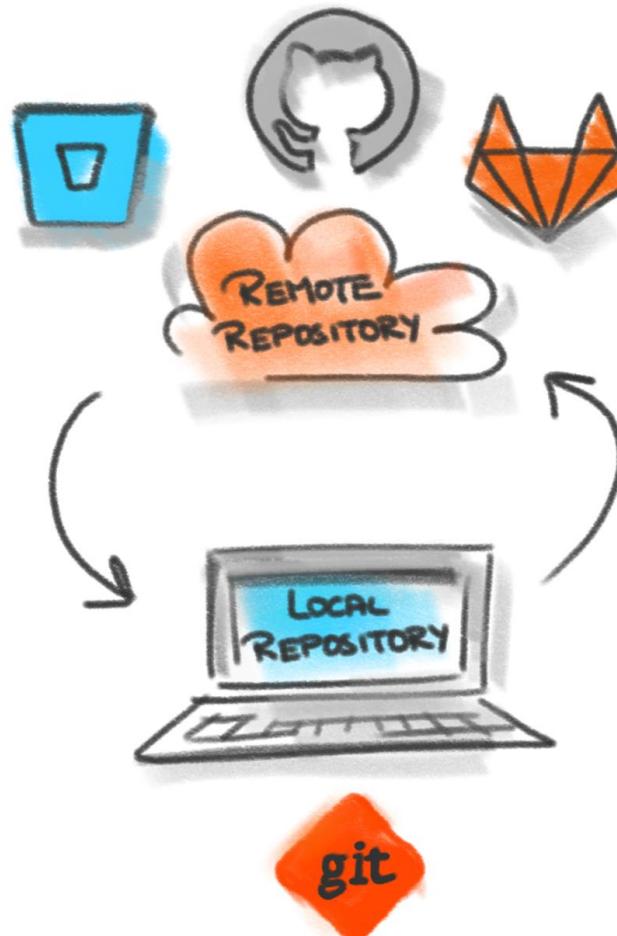
IDE Commit Changes Dialog:

The dialog shows the following details:

- Changelist:** Default Changelist
- Git:**
 - Author:** [Empty field]
 - Amend commit
 - Sign-off commit
- Before Commit:**
 - Reformat code
 - Rearrange code
 - Optimize imports
- Commit Message:** Update lorem.txt
- Diff:** Side-by-side viewer showing the change in the lorem.txt file. The original line "ipsum" is shown in the left pane, and the modified line "dolor sit amet," is shown in the right pane. A green highlight indicates the change. The commit hash 2a4d3a20bdc9daec9adf95b451fc46a0e59b31a8 is also visible.
- Buttons:** Commit, Cancel, Help, Commit and Push..., Create Patch...

A mouse cursor is hovering over the "Commit and Push..." button, which is highlighted in teal.

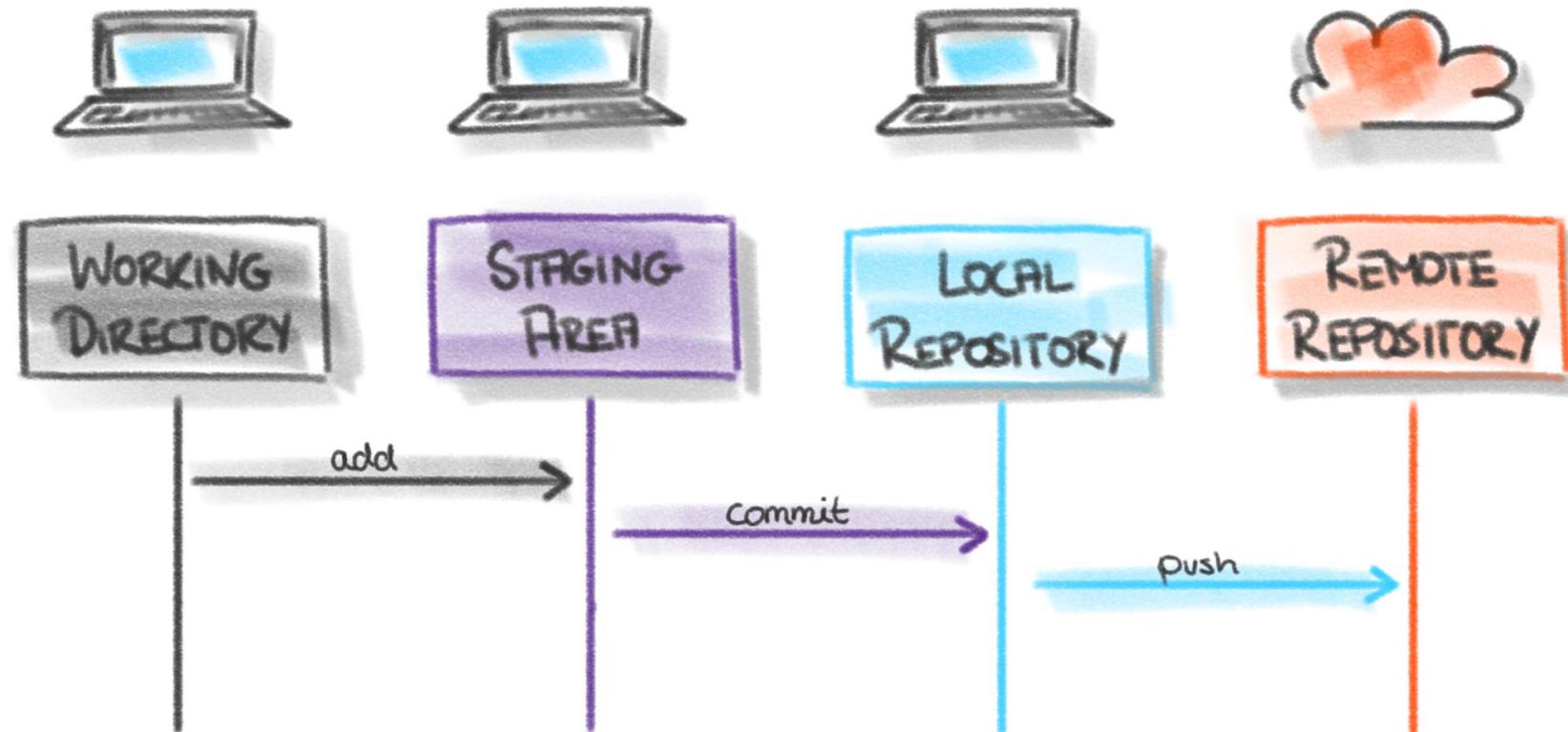
Von Kommandozeilen, GUIs und Webanwendungen



The screenshot shows a GitHub repository page. At the top, the repository name is `schulungsteilnehmer / 2024-gruppe-tano`. Below the header, there are tabs for Code, Issues, Pull requests, Actions, Projects, Security, and Insights. The Code tab is selected. The main content area shows a commit from `schulungsteilnehmer` with the message "Initial c..." and timestamp "4d938ad · 27 minutes ago". The commit has "1 Commit". Below the commit is a file named `README.md` with the status "Initial commit" and timestamp "27 minutes ago". To the right of the commit list is a sidebar titled "About" which states "No description, website, or topics provided." It also lists "Readme", "Activity", "0 stars", "1 watching", "0 forks", and a "Report repository" link. Below the sidebar is a section titled "Releases" with the message "No releases published".

<https://t.ly/BNZb7>

Die wichtigsten Kommandos für heute und morgen (Part 2)



Erster Tag



08:45 Uhr – Vorgehensmodelle (Wasserfall und Scrum)



09:15 Uhr - Projektvorstellung: Zeugnismanagement-Software & Arbeiten am Projekt



12:00 Uhr – Rundgang durch das Gebäude



12:45 Uhr - Mittagspause



13:05 Uhr – Vorstellung von Git u. Arbeiten am Projekt



15:15 Uhr - Feierabend



Pause



Erster Tag



08:45 Uhr – Vorgehensmodelle (Wasserfall und Scrum)



09:15 Uhr - Projektvorstellung: Zeugnismanagement-Software & Arbeiten am Projekt



12:00 Uhr – Rundgang durch das Gebäude



12:45 Uhr - Mittagspause



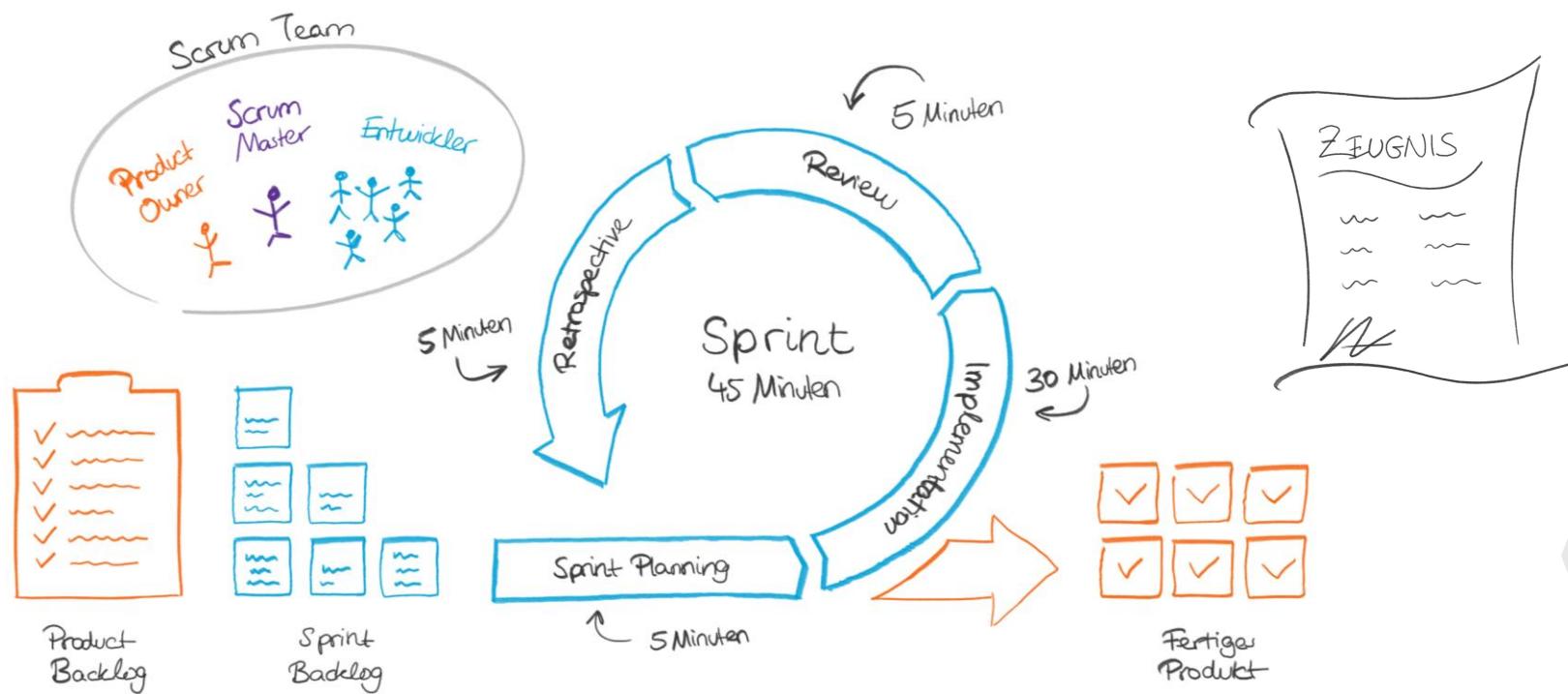
13:05 Uhr – Vorstellung von Git u. Arbeiten am Projekt



15:15 Uhr - Feierabend



Projektdurchführung



User: Schulungsteilnehmer

Passwort: Schulung1

WLAN: msg-DAVID-internet

Passwort: Intrnt4Vstrs2018!

Repository-URL: <https://github.com/schulungsteilnehmer>

Git-Kommandos:
`git add .`
`git commit -m "Message"`
`git push`

Aufgabenstellung

Das David-Gymnasium hat Sie beauftragt, eine Zeugnismanagement-Software zu entwickeln, mit deren Hilfe ein Zeugnis für Schüler in eine Datei gespeichert werden kann.

Dazu sollen die erforderlichen Daten manuell über eine Konsole eingelesen werden, mit deren Hilfe die Software automatisch analysiert, ob der betreffende Schüler versetzt wird.

User-Story mit absoluter Priorität

Berücksichtigung von Fehltagen

Als Benutzer möchte ich, dass die Fehltagen als weiteres Versetzungskriterium berücksichtigt werden.

Akzeptanzkriterien

- I. Fehltagen und entschuldigte Fehltagen werden abgefragt
- II. Bei 30 oder mehr unentschuldigten Fehltagen wird die Bemerkung „Der Schüler wird nicht versetzt“ auf dem Zeugnis vermerkt
- III. Bei weniger unentschuldigten Fehltagen wird die Bemerkung „Der Schüler wird versetzt“ auf dem Zeugnis vermerkt
- IV. Die Fehltagen und davon unentschuldigten Fehltagen werden auf dem Zeugnis vermerkt



Aufgabenstellung

Das David-Gymnasium hat Sie beauftragt, eine Zeugnismanagement-Software zu entwickeln, mit deren Hilfe ein Zeugnis für Schüler in eine Datei gespeichert werden kann. Dazu sollen die erforderlichen Daten manuell über eine Konsole eingelesen werden, mit deren Hilfe die Software automatisch analysiert, ob der betreffende Schüler versetzt wird.

Git-Kommandos:

`git add .`

`git commit -m "Message"`

`git push`

User-Story mit absoluter Priorität, ist laut SCRUM nicht vorgesehen

Berücksichtigung von Fehltagen

Als Benutzer möchte ich, dass die Fehltage als weiteres Versetzungskriterium berücksichtigt werden.

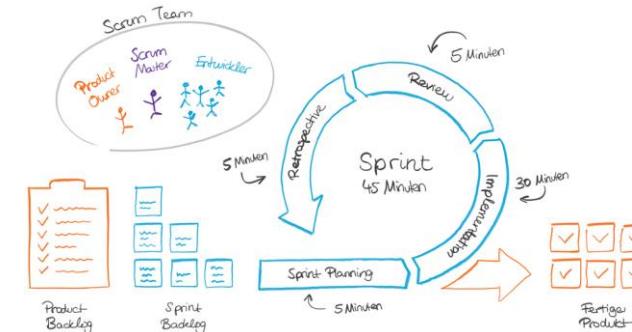
Akzeptanzkriterien

- I. Fehltage und entschuldigte Fehltage werden abgefragt
- II. Bei 30 oder mehr unentschuldigten Fehltagen wird die Bemerkung „Der Schüler wird nicht versetzt“ auf dem Zeugnis vermerkt
- III. Bei weniger unentschuldigten Fehltagen wird die Bemerkung „Der Schüler wird versetzt“ auf dem Zeugnis vermerkt
- IV. Die Fehltage und davon unentschuldigten Fehltage werden auf dem Zeugnis vermerkt

EIGENTLICH ABSOLUTES NO-GO!!

Laut SCRUM ist es **absolut verboten, laufende Sprints zu erweitern bzw. zu verändern.**

Dies bedeutet, dass derlei Einwürfe grundsätzlich abgelehnt werden und stattdessen in einem **künftigen Sprint-Planning** berücksichtigt werden!



Aufgabenstellung

Das David-Gymnasium hat Sie beauftragt, eine Zeugnismanagement-Software zu entwickeln, mit deren Hilfe ein Zeugnis für Schüler in eine Datei gespeichert werden kann. Dazu sollen die erforderlichen Daten manuell über eine Konsole eingelesen werden, mit deren Hilfe die Software automatisch analysiert, ob der betreffende Schüler versetzt wird.

Erster Tag



08:45 Uhr – Vorgehensmodelle (Wasserfall und Scrum)



09:15 Uhr - Projektvorstellung: Zeugnismanagement-Software & Arbeiten am Projekt



12:00 Uhr – Rundgang durch das Gebäude



12:45 Uhr - Mittagspause



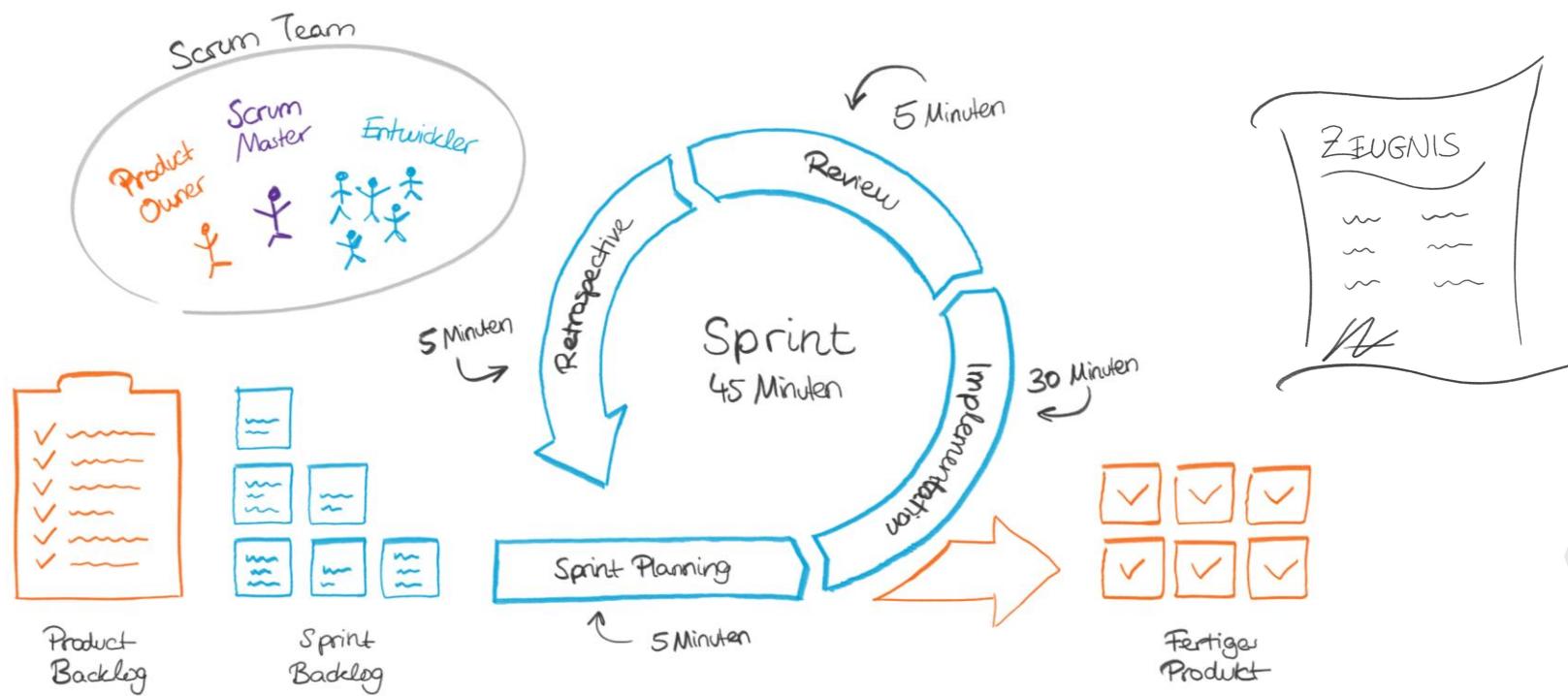
13:05 Uhr – Vorstellung von Git u. Arbeiten am Projekt



15:15 Uhr - Feierabend



Projektdurchführung



User: Schulungsteilnehmer

Passwort: Schulung1

WLAN: msg-DAVID-internet

Passwort: Intrnt4Vstrs2018!

Repository-URL: <https://github.com/schulungsteilnehmer>

Git-Kommandos:
`git add .`
`git commit -m "Message"`
`git push`

Aufgabenstellung

Das David-Gymnasium hat Sie beauftragt, eine Zeugnismanagement-Software zu entwickeln, mit deren Hilfe ein Zeugnis für Schüler in eine Datei gespeichert werden kann.

Dazu sollen die erforderlichen Daten manuell über eine Konsole eingelesen werden, mit deren Hilfe die Software automatisch analysiert, ob der betreffende Schüler versetzt wird.

Feierabend



Praktikumstage bei der msg DAVID GmbH

Mai 2024

Zweiter Tag



08:45 Uhr – Projektsprint u. Einblick hinter die Kulissen



10:30 Uhr – Entwicklungsprinzipien und Arbeiten am Projekt



12:00 Uhr - Mittagspause



12:30 Uhr - Vorstellung eurer und unserer Ergebnisse



14:15 Uhr - Infos zum Unternehmen & Ausbildung



14:45 Uhr - Feedback & Abschluss

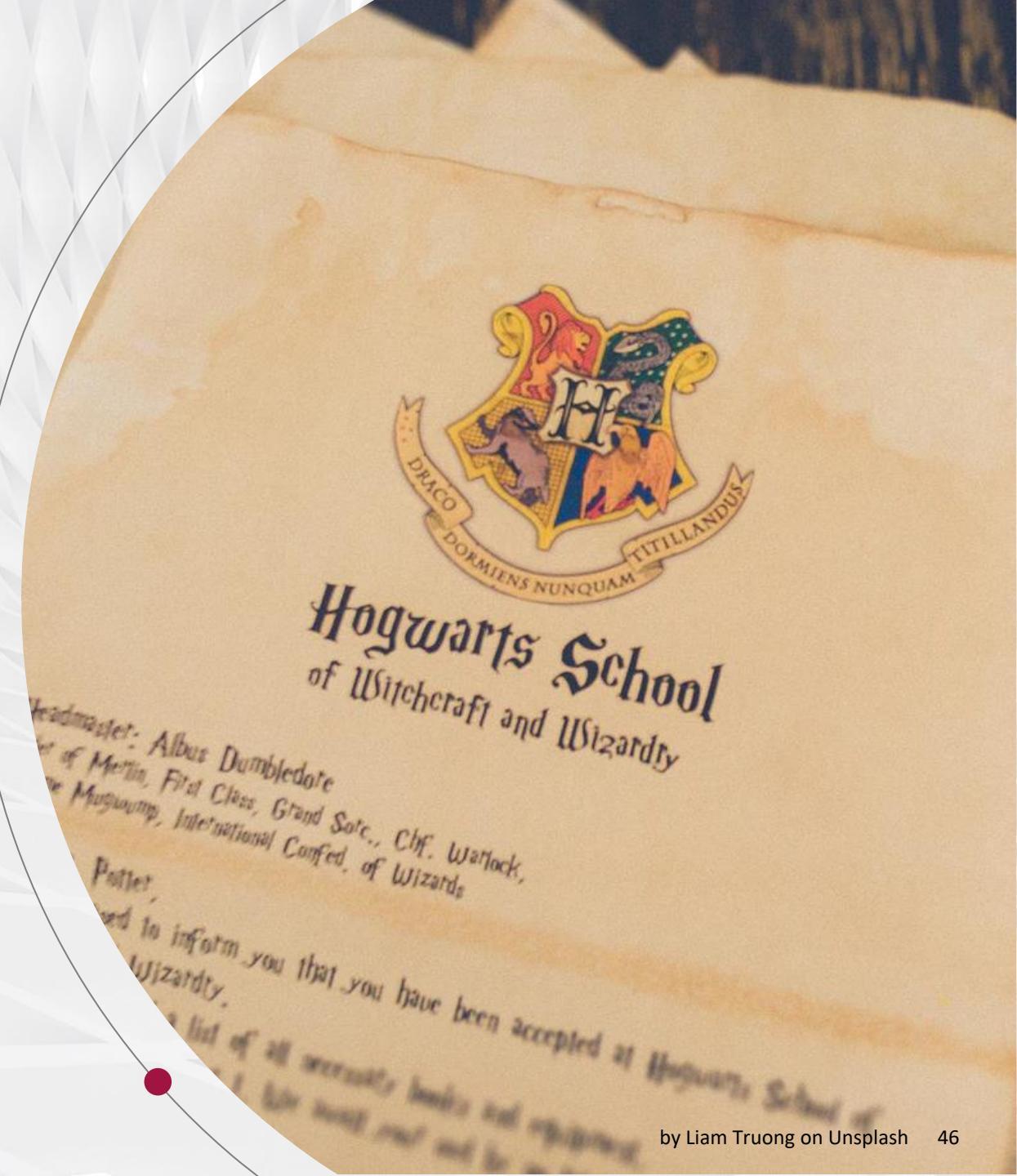


15:15 Uhr - Feierabend

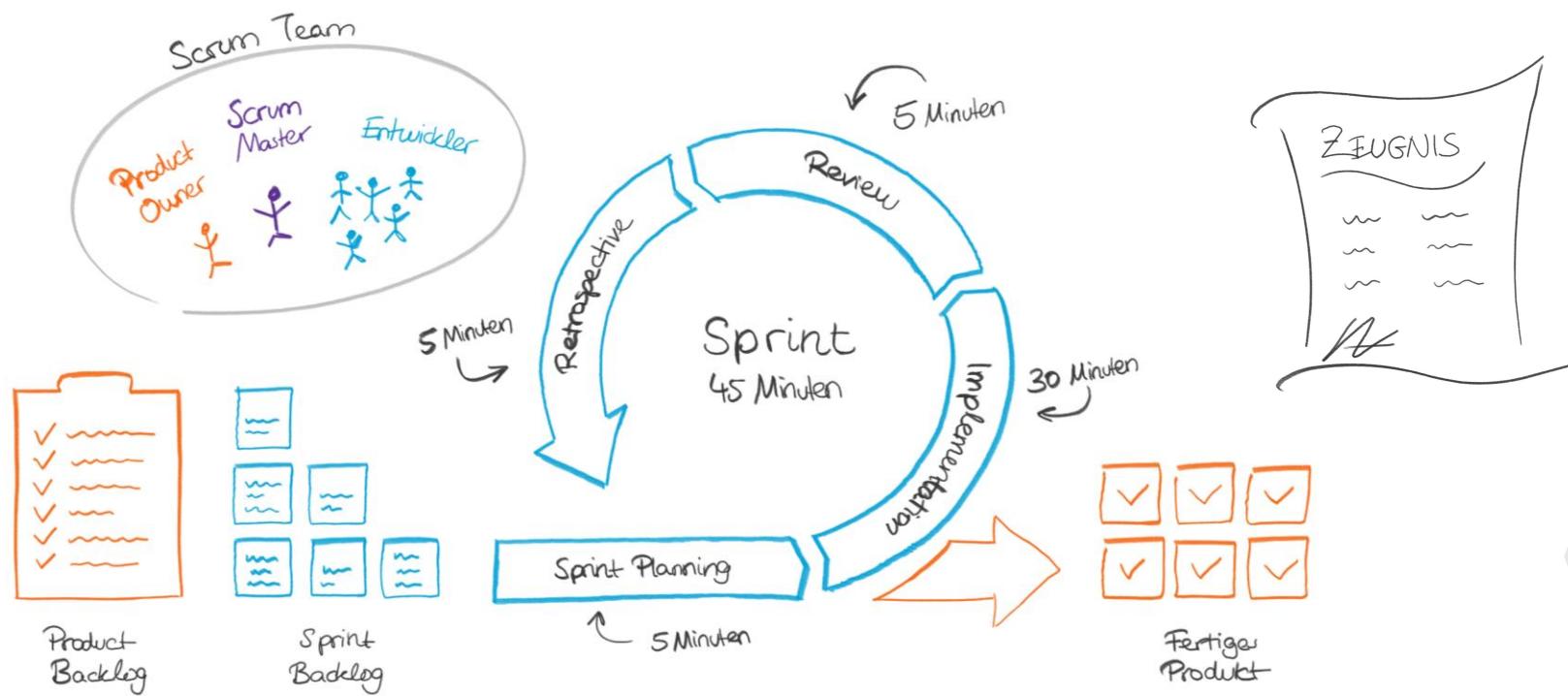


Projektarbeit

Leon Alexander Widerspan



Projektdurchführung



User: Schulungsteilnehmer

Passwort: Schulung1

WLAN: msg-DAVID-internet

Passwort: Intrnt4Vstrs2018!

Repository-URL: <https://github.com/schulungsteilnehmer>

Git-Kommandos:
`git add .`
`git commit -m "Message"`
`git push`

Aufgabenstellung

Das David-Gymnasium hat Sie beauftragt, eine Zeugnismanagement-Software zu entwickeln, mit deren Hilfe ein Zeugnis für Schüler in eine Datei gespeichert werden kann.

Dazu sollen die erforderlichen Daten manuell über eine Konsole eingelesen werden, mit deren Hilfe die Software automatisch analysiert, ob der betreffende Schüler versetzt wird.

Zweiter Tag



08:45 Uhr – Projektsprint u. Einblick hinter die Kulissen



10:30 Uhr – Entwicklungsprinzipien und Arbeiten am Projekt



12:00 Uhr - Mittagspause



12:30 Uhr - Vorstellung eurer und unserer Ergebnisse



14:15 Uhr - Infos zum Unternehmen & Ausbildung



14:45 Uhr - Feedback & Abschluss



15:15 Uhr - Feierabend



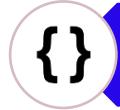
Pause



Zweiter Tag



08:45 Uhr – Projektsprint u. Einblick hinter die Kulissen



10:30 Uhr – Entwicklungsprinzipien und Arbeiten am Projekt



12:00 Uhr - Mittagspause



12:30 Uhr - Vorstellung eurer und unserer Ergebnisse



14:15 Uhr - Infos zum Unternehmen & Ausbildung



14:45 Uhr - Feedback & Abschluss



15:15 Uhr - Feierabend



Entwicklungsprinzipien: DRY & KISS

Jack Künstler



D R Y

- **Don't Repeat Yourself**
Wiederhole dich nicht
- Prinzip aus dem Clean Code
- Vermeidung von Redundanzen



Aufgabe

- Fünfmal eine Eingabe über die Konsole auslesen und Eingabe direkt ausgeben

```
using System;

public class InefficientSolution
{
    public static void Main()
    {
        Console.WriteLine("Enter five values");
        Console.WriteLine("Value 1");
        String inputOne = Console.ReadLine();
        Console.WriteLine(inputOne);
        Console.WriteLine("Value 2");
        String inputTwo = Console.ReadLine();
        Console.WriteLine(inputTwo);
        Console.WriteLine("Value 3");
        String inputThree = Console.ReadLine();
        Console.WriteLine(inputThree);
        Console.WriteLine("Value 4");
        String inputFour = Console.ReadLine();
        Console.WriteLine(inputFour);
        Console.WriteLine("Value 5");
        String inputFive = Console.ReadLine();
        Console.WriteLine(inputFive);
    }
}
```

```
using System;

public class MoreEfficientSolution
{
    public static void Main()
    {
        Console.WriteLine("Enter five values");
        handleInput(1);
        handleInput(2);
        handleInput(3);
        handleInput(4);
        handleInput(5);
    }

    private static void handleInput(int sequenceNumber)
    {
        Console.WriteLine("Value " + sequenceNumber);
        String inputOne = Console.ReadLine();
        Console.WriteLine(inputOne);
    }
}
```

K I S S

- **Keep It Simple, Stupid!**
- Nicht nur für Softwareentwicklung geeignet
- Strebe eine möglichst einfache Lösung an (die wenigsten Annahmen und Variablen)
- Das Internet auf Basis der TCP/IP-Protokollfamilie wurde nach diesem Prinzip entwickelt
- Gerne zusammen mit PEE (**P**oint, **E**vidence, **E**xplain: Behaupte, Belege, Erkläre) angewendet um die einfachste Lösung zu verargumentieren



```
using System;

public class ImprovedSolution
{
    public static int sequence = 1;
    public static void Main()
    {
        Console.WriteLine("Enter five values");
        handleInput(sequence);
        incrementSequence();
        handleInput(sequence);
        incrementSequence();
        handleInput(sequence);
        incrementSequence();
        handleInput(sequence);
        incrementSequence();
        handleInput(sequence);
        incrementSequence();
        handleInput(sequence);
    }

    private static void handleInput(int sequenceNumber)
    {
        Console.WriteLine(formatString(sequenceNumber));
        String inputOne = Console.ReadLine();
        Console.WriteLine(inputOne);
    }

    private static string formatString(int sequenceNumber)
    {
        return "Value " + sequenceNumber;
    }

    private static void incrementSequence()
    {
        sequence = sequence +1;
    }
}

// 36 lines of code
```

```
using System;

public class EvenMoreEfficientSolution
{
    public static void Main()
    {
        Console.WriteLine("Enter five values");
        int sequence = 1;
        while(sequence < 6)
        {
            handleInput(sequence++);
        }
    }

    private static void handleInput(int sequenceNumber)
    {
        Console.WriteLine("Value " + sequenceNumber);
        Console.WriteLine(Console.ReadLine());
    }
}

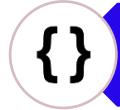
// 20 lines of code
```

[schulungsteilnehmer \(github.com\)](https://schulungsteilnehmer.github.com)

Zweiter Tag



08:45 Uhr – Projektsprint u. Einblick hinter die Kulissen



10:30 Uhr – Entwicklungsprinzipien und Arbeiten am Projekt



12:00 Uhr - Mittagspause



12:30 Uhr - Vorstellung eurer und unserer Ergebnisse



14:15 Uhr - Infos zum Unternehmen & Ausbildung



14:45 Uhr - Feedback & Abschluss

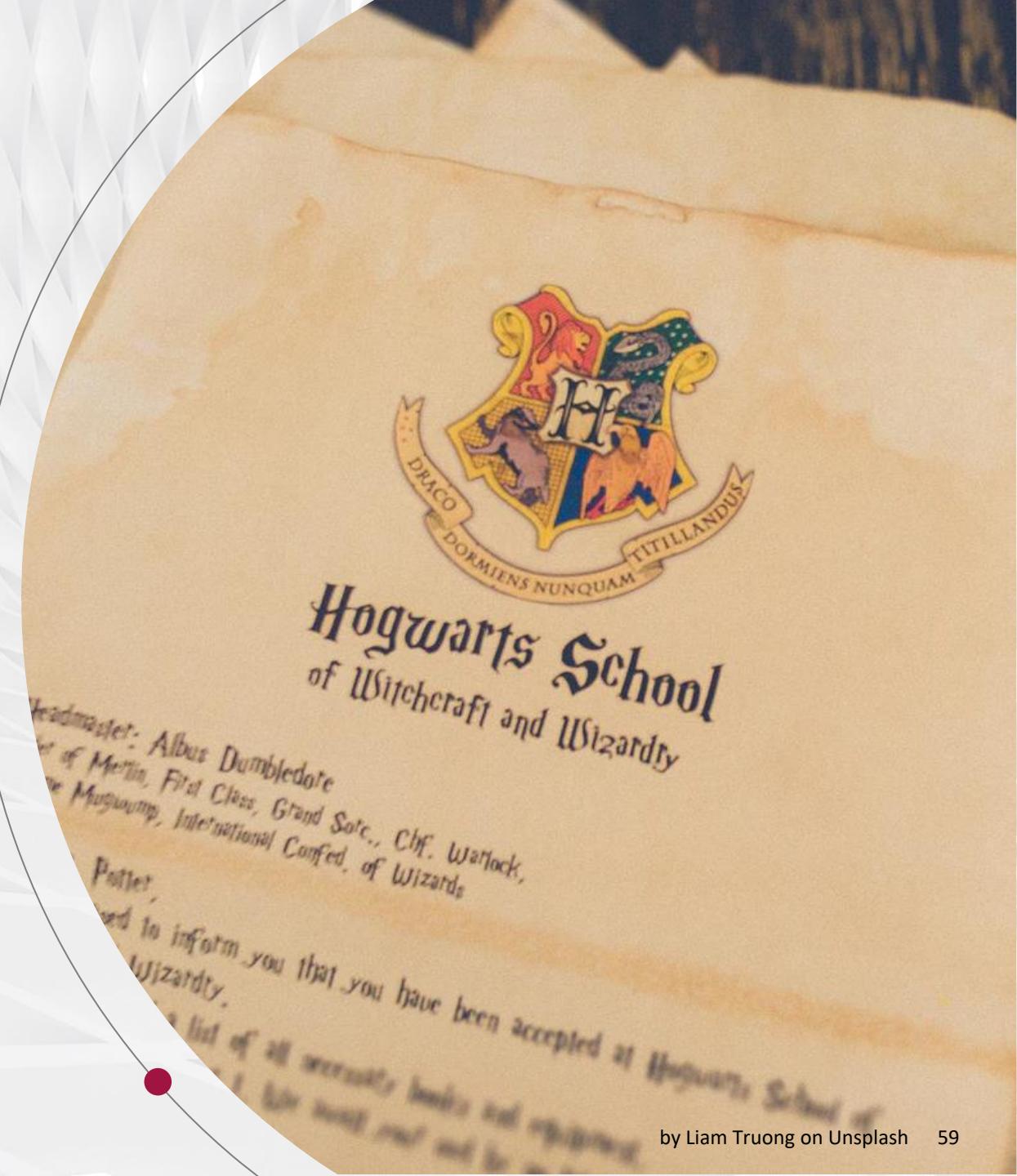


15:15 Uhr - Feierabend



Projektarbeit

Eugen und Fabian



Projektdurchführung – Beispiel für Endresultat nach Kundenwunsch

In etwa sollte eure Ausgabe folgendermaßen ausschauen:

```
=====Zeugnis=====
Name: Max Muster
Datum: 27.04.2022
=====
```

Mathe(LK)	:	13
Deutsch	:	7
Englisch	:	5
Biologie	:	9
Sport	:	12
Kunst	:	5
Geschichte	:	4
Informatik(LK)	:	14

Durchschnittsnote: 3,1

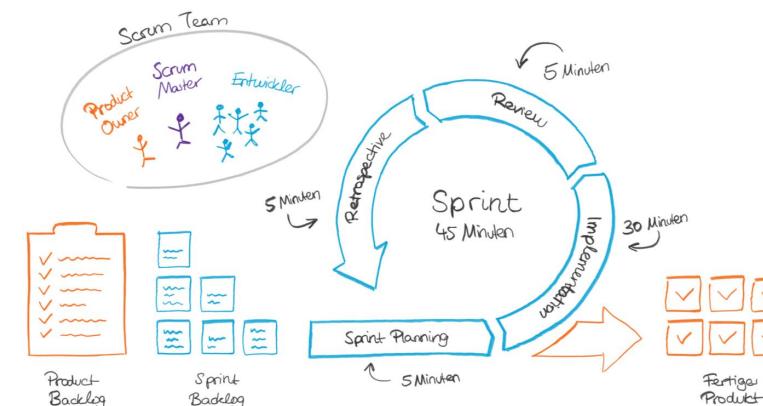
Fehltage: 25

Davon Unentschuldigt: 25

Der Schüler wird versetzt.



Git-Kommandos:
git add .
git commit -m "Message"
git push



Aufgabenstellung

Das David-Gymnasium hat Sie beauftragt, eine Zeugnismanagement-Software zu entwickeln, mit deren Hilfe ein Zeugnis für Schüler in eine Datei gespeichert werden kann.

Dazu sollen die erforderlichen Daten manuell über eine Konsole eingelesen werden, mit deren Hilfe die Software automatisch analysiert, ob der betreffende Schüler versetzt wird.

Zweiter Tag



08:45 Uhr – Projektsprint u. Einblick hinter die Kulissen



10:30 Uhr – Entwicklungsprinzipien und Arbeiten am Projekt



12:00 Uhr - Mittagspause



12:30 Uhr - Vorstellung eurer und unserer Ergebnisse



14:15 Uhr - Infos zum Unternehmen & Ausbildung



14:45 Uhr - Feedback & Abschluss



15:15 Uhr - Feierabend



Mittag



Zweiter Tag



08:45 Uhr – Projektsprint u. Einblick hinter die Kulissen



10:30 Uhr – Entwicklungsprinzipien und Arbeiten am Projekt



12:00 Uhr - Mittagspause



12:30 Uhr - Vorstellung eurer und unserer Ergebnisse



14:15 Uhr - Infos zum Unternehmen & Ausbildung



14:45 Uhr - Feedback & Abschluss



15:15 Uhr - Feierabend



Vorstellung der Ergebnisse

Ihr seid dran



Pause



Zweiter Tag



08:45 Uhr – Projektsprint u. Einblick hinter die Kulissen



10:30 Uhr – Entwicklungsprinzipien und Arbeiten am Projekt



12:00 Uhr - Mittagspause



12:30 Uhr - Vorstellung eurer und unserer Ergebnisse



14:15 Uhr - Infos zum Unternehmen & Ausbildung



14:45 Uhr - Feedback & Abschluss



15:15 Uhr - Feierabend



Infos zum Unternehmen und der Ausbildung

Leon Alexander Widerspan



Gegründet
1991

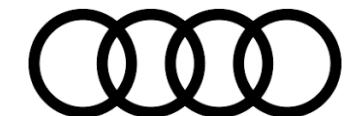
Daten & Fakten

Mitarbeiter
250

Standort
Mittelweg 7
38106 Braunschweig



Branche Automotive &
Manufacturing



Wir sind Teil einer international agierenden Unternehmensgruppe

- Gründungsjahr: 1980
- Unternehmenssitz: Ismaning/München
- Umsatz: 1.4 Mrd. € (2023)
- Über 10.000 Mitarbeiter
- Standorte in 28 Ländern
- Branchen:
 - Automotive
 - Banking
 - Food
 - Insurance
 - Life Science & Healthcare
 - Public Sector
 - Telecommunications
 - Travel & Logistics
 - Utilities
- Ausgezeichnet als „Bester Arbeitgeber Deutschlands“ von Great Place to Work in 2020

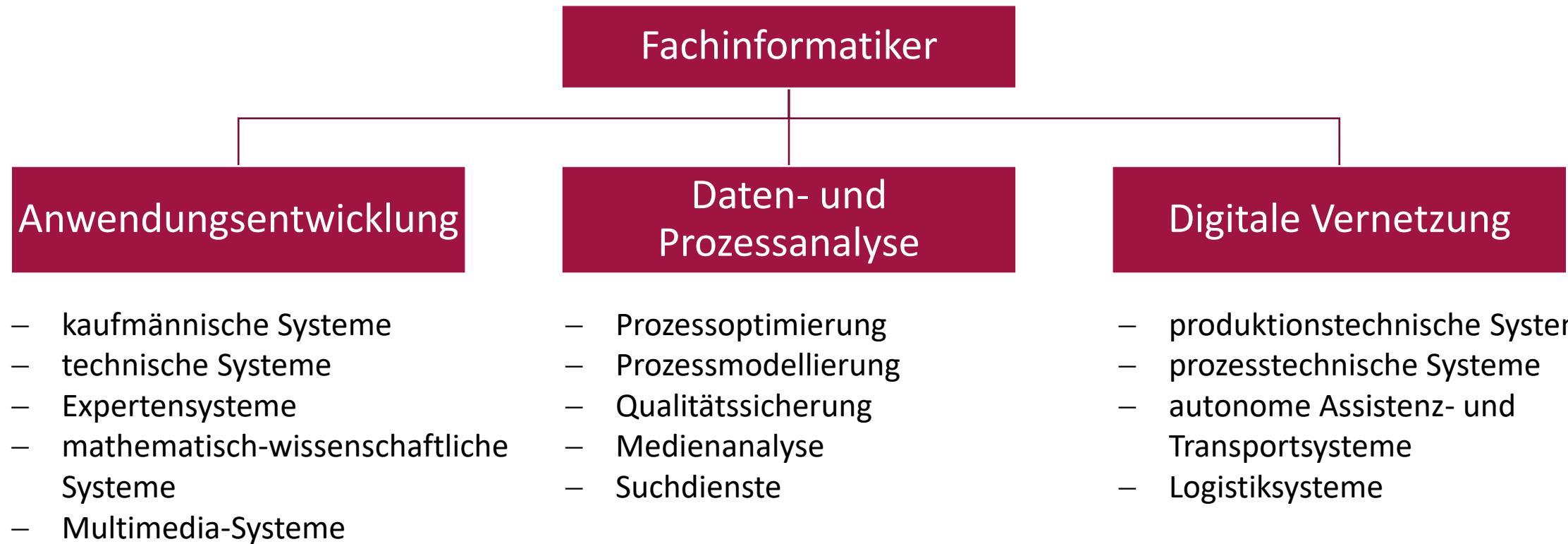


Neuordnung der IT-Berufe zum 01.08.2020

Die Verordnung über die Berufsausbildung im Bereich der Informations- und Telekommunikationstechnik vom 10. Juli 1997 wurde überarbeitet.

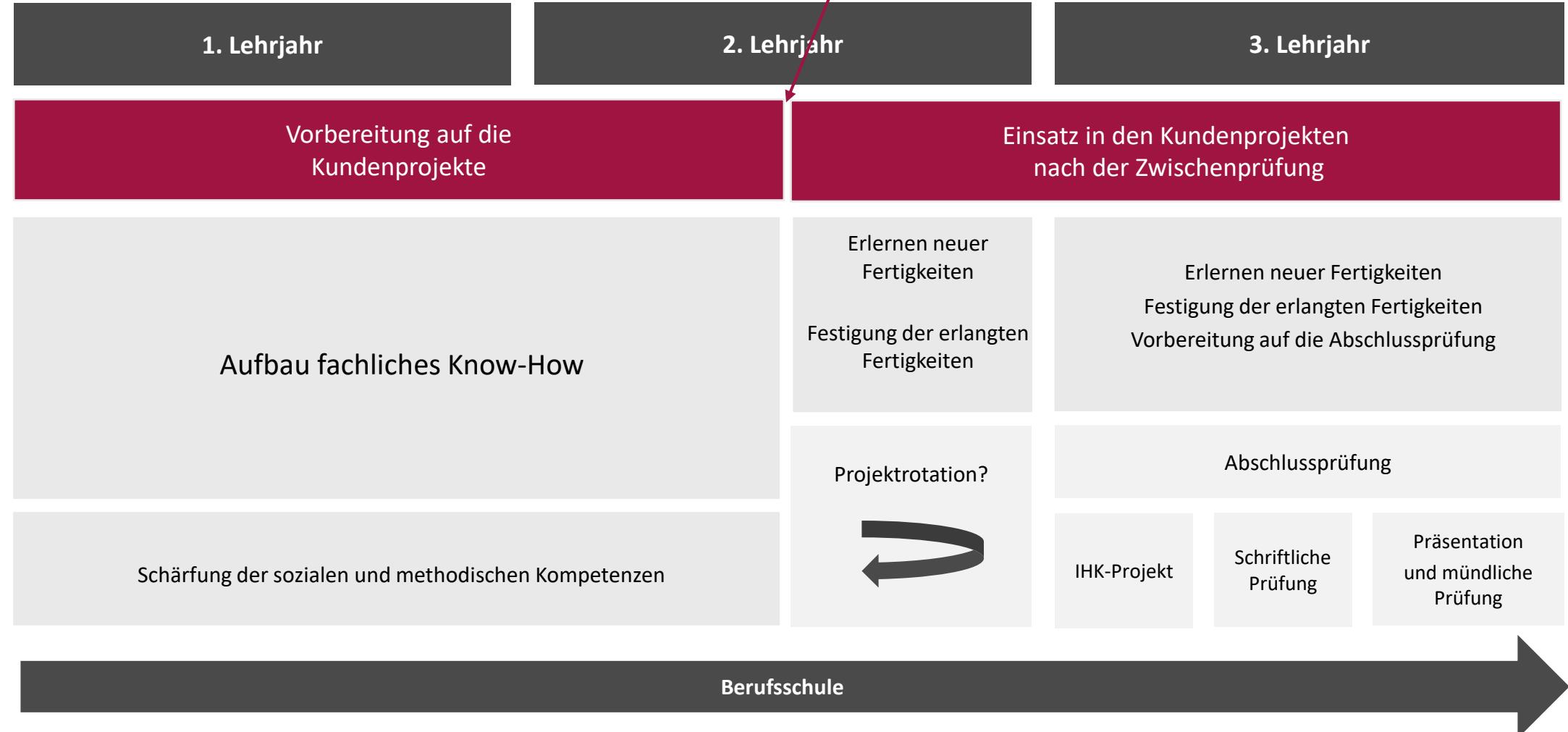
Seit dem 01.08.2020 wird die Ausbildung zum Fachinformatiker in vier Fachrichtungen angeboten:

- **Anwendungsentwicklung**
Softwareentwicklung und Programmierung
- **Digitale Vernetzung (neu)**
Vernetzung, Automatisierung, Optimierung digitaler Prozesse und smarter Produkte
- **Daten- und Prozessanalyse (neu)**
Datenbasierte Lösungen für digitale Produktions- und Geschäftsprozesse



<https://www.ausbildung.de/berufe/fachinformatiker>

Ausbildung Fachinformatiker



Zweiter Tag



08:45 Uhr – Projektsprint u. Einblick hinter die Kulissen



10:30 Uhr – Entwicklungsprinzipien und Arbeiten am Projekt



12:00 Uhr - Mittagspause



12:30 Uhr - Vorstellung eurer und unserer Ergebnisse



14:15 Uhr - Infos zum Unternehmen & Ausbildung



14:45 Uhr - Feedback & Abschluss



15:15 Uhr - Feierabend



War schön mit euch ❤
Wie fandet ihrs?
Schreibt uns gern!

hallo@msg-david.de

michael.kohrs@msg-david.de

oliver.wiesbrock@msg-david.de

lisa.rosenberg@msg-david.de

Eure Ergebnisse und die Musterlösung findet ihr
hier:

<https://github.com/schulungsteilnehmer>

