



Technische
Universität
Braunschweig

Institut für
Flugführung



Softwareprojektmanagement

Einführung in das Projektmanagement bei Softwareprojekten

Prof. Dr.-Ing. Peter Hecker, Dipl.-Ing. Paul Frost, 11. April 2017

Agenda

- 04. April Kick-Off
- 11. April Projektmanagement**
- 18. April Vorgehensmodelle
- 25. April Versionsverwaltung und Entwicklungsumgebungen
- 02. Mai Einführung Arduino/Funduino
- 09. Mai Entwicklungsumgebungen und Debugging
- 16. Mai Dokumentation und Testing
- 23. Mai Dateieingabe und -ausgabe
- 30. Mai GUI-Erstellung mit Qt
- 06. Juni Exkursionswoche
- 13. Juni Bibliotheken
- 20. Juni Netzwerke
- 27. Juni Projektarbeit
- 04. Juli Projektarbeit
- 11. Juli Vorbereitung der Abgabe

Teil I

Wiederholung

Klausur

Bedingungen für die Teilnahme an der Klausur

- Ich war bereits für die Informatikprüfung angemeldet.

Bedingungen für die Bearbeitung der Projektmappe

- Ich war noch nicht für eine Prüfung angemeldet.
ODER
- Ich habe mich rechtzeitig von der Prüfung abgemeldet.
ODER
- Ich brauche nur einen Schein.

Prüfungsmodalitäten

alter Modus

Prüfungsart

Informatik/C++:	Klausur 180 Min	Gewichtung: 2/3
API:	Klausur 60 Min	Gewichtung: 1/3

Voraussetzungen für die Vergabe von Leistungspunkten

50% innerhalb des Gesamtmoduls

Inhalte

Inhalte aus dem Sommersemester 2016 (siehe StudIP)

MATLAB/SIMULINK

Es wird keine Übung für MATLAB/SIMULINK geben!

Übungsmöglichkeiten

- Rechner mit MATLAB/SIMULINK-Installationen werden vom GITZ bereitgestellt.
- Online-Tutorial:
<https://matlabacademy.mathworks.com/R2016b/portal.html?course=gettingstarted>
- Folien sind im StudIP verfügbar

Prüfungsmodalitäten

neuer Modus

Prüfungsart

Informatik/C++:	Klausur 180 Min	Gewichtung: 2/3
API:	Projektmappe	Gewichtung: 1/3

Voraussetzungen für die Vergabe von Leistungspunkten

50% in Informatik/C++ und
50% in API

Ausarbeitung von Projektideen

- Vorstellung von Projektideen zur Inspiration

Wie weit sollte ich heute sein?

- Ich sollte einer Gruppe angehören
- Die Gruppe sollte mindestens eine Projektidee ausgearbeitet haben
- Die Gruppe sollte Risiken formuliert haben
- Die Gruppe sollte erste Anforderungen formuliert haben

Projekt

RFID-Verwaltung der Arduino-Kits

Die Idee

Verwaltung der Arduino-Kits über eine PC-Datenbank mithilfe von RFID

Funktionale Anforderungen

- Auslesen der Kit-Nr. mittels eines RFID-Lesers
- Zuweisung von Informationen (Bauteile, ausleihende Gruppe) zur Kit-Nr.
- Speicherung der Informationen auf dem PC
- Bearbeitung der gespeicherten Daten, beispielsweise bei Defekt von Bauteilen

Projekt

RFID-Verwaltung der Arduino-Kits

Enthaltene Module

- RFID-Aufkleber und -Lesegerät
- Datenbank (PC)

Risiken

- Verlorene Arbeitszeit bei Misslingen des Projektes

Nutzen

- Einfache Übersicht über Zustand der Kits
- Zeitersparnis bei der Verwaltung

Teil II

Softwareprojektmanagement

Wie Softwareprojekte wirklich bearbeitet werden...



Abbildung 1: Wie der Kunde es erklärt hat¹

¹URL: <http://projectcartoon.com>.

Wie Softwareprojekte wirklich bearbeitet werden...

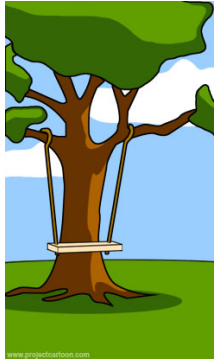


Abbildung 1: Wie der Projektleiter es verstanden hat

Wie Softwareprojekte wirklich bearbeitet werden...



Abbildung 1: Wie der Analyst es auffasst

Wie Softwareprojekte wirklich bearbeitet werden...

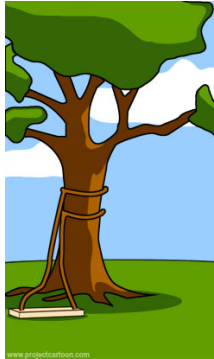


Abbildung 1: Wie der Programmierer es geschrieben hat

Wie Softwareprojekte wirklich bearbeitet werden...



Abbildung 1: Was die Beta-Tester erhalten

Wie Softwareprojekte wirklich bearbeitet werden...



Abbildung 1: Wie der Wirtschaftsberater es verkauft

Wie Softwareprojekte wirklich bearbeitet werden...

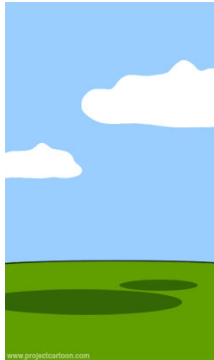


Abbildung 1: Wie das Projekt dokumentiert wurde

Wie Softwareprojekte wirklich bearbeitet werden...



Abbildung 1: Was der Kunde wirklich gebraucht hätte

Problemstellung

- Anforderungen der Kunden werden nicht erfüllt
- Kosten für Software sind zu hoch
- Methoden der Softwareentwicklung ändern sich schnell
- Spezialisierung von Entwicklern kurzlebig (Innovationsdruck)
- Weiterverwendung alter Technologien
- Fokus auf Technologien zur Realisierung, nicht bei der eigtl. Anwendung

Problemstellung

- Anforderungen der Kunden werden nicht erfüllt
- Kosten für Software sind zu hoch
- Methoden der Softwareentwicklung ändern sich schnell
- Spezialisierung von Entwicklern kurzlebig (Innovationsdruck)
- Weiterverwendung alter Technologien
- Fokus auf Technologien zur Realisierung, nicht bei der eigtl. Anwendung

Problemstellung Anforderungen¹

- Kundenspezifische Problematik
 - Es ist nicht genau bekannt was entwickelt wird.
 - Anforderungen werden ungenau/falsch formuliert
 - Fachsprache kann für Verständnisprobleme sorgen
 - Unterschiedliche Anforderungen der Projektbeteiligten (Stakeholder)
- Randbedingungen
 - Politische Einschränkungen
 - Änderungen der Anforderungen möglich

¹Thomas Thüm. "Vorlesung Software Engineering, Anforderungsanalyse". S. 5.

Problemstellung Anforderungen¹

- Kundenspezifische Problematik
 - Es ist nicht genau bekannt was entwickelt wird.
 - Anforderungen werden ungenau/falsch formuliert
 - Fachsprache kann für Verständnisprobleme sorgen
 - Unterschiedliche Anforderungen der Projektbeteiligten (Stakeholder)
- Randbedingungen
 - Politische Einschränkungen
 - Änderungen der Anforderungen möglich

Notwendigkeit, Anforderungen so genau wie möglich auszuformulieren
Auch zur rechtlichen Absicherung!

¹Thüm, "Vorlesung Software Engineering, Anforderungsanalyse", S. 5.

Anforderungen ermitteln²

Möglichkeiten, Kundenanforderungen zu ermitteln

- Interviews
- Dokumentenanalyse
- Focus Group
- Prototyping
- Beobachtungen
- Fragebögen

²Thüm, “Vorlesung Software Engineering, Anforderungsanalyse”, S. 19.

Sinnvolle Anforderungen

Die Qualität von Anforderungen wird über die folgenden Charakteristiken definiert³

- korrekt
- unzweideutig
- vollständig
- widerspruchsfrei
- priorisiert
- verifizierbar

³IEEE. IEEE Guide to Software Requirements Specification ANSI/IEEE Std 29148-2011. 2011.

Anforderungen verfassen⁴

Stilistisch sollten Anforderungen wie folgt formuliert werden:



- Kurze Sätze
- Eine Aussage pro Satz
- Keine Verschachtelungen
- Konsistente Terminologie
- Wenig Abkürzungen und nur bekannte Abk. verwenden
- Vermeidung von Generalität
- Verbindlichkeiten deutlich ausdrücken
- Aktiv formulieren

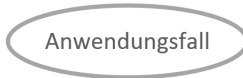
⁴Thüm, "Vorlesung Software Engineering, Anforderungsanalyse", S. 22.

Use-Case Diagramm

Akteur



Anwendungsfall



Assoziation



Ableitung



Erweiterung



Einschließen



Use-Case Diagramm

Beispiel

- Ein Akteur kann Aktion A ausführen
- Wenn Aktion A ausgeführt wird ist Aktion B darin bereits enthalten
- Wenn Aktion A ausgeführt wird kann Aktion C im Anschluss ausgeführt werden

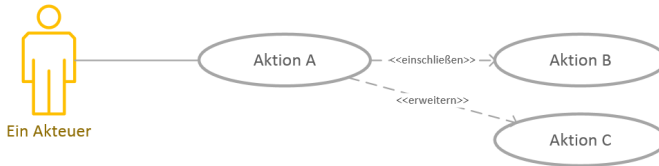


Abbildung 2: Beispiel eines Use-Case Diagramms

Lastenheft

- Definition
 - Vom Auftraggeber formuliert
 - Legt die Anforderungen fest
 - Englische Bezeichnung: Statement of Work
- Abgrenzung
 - Legt fest was, aber nicht wie es gemacht wird
 - Ist so präzise wie möglich, ohne einen Lösungsweg vorzugeben

Beispielanforderungen Lastenheft

- Festlegen des Funktionsumfangs
 - Der Benutzer soll die Möglichkeit haben, Messaufzeichnungen zu editieren
 - Die gemessene Entfernung soll auf einer grafischen Benutzeroberfläche angezeigt werden
- Festlegen der Zielplattform
 - Die grafischen Benutzeroberfläche soll auf mobilen Endgeräten bedienbar sein
 - Vorgabe von Betriebssystemen (iOS 8, Android 5.1, Windows 10)

Ungültige Anforderungen

- Nutzung einer bestimmten Webtechnologie (HTML 5, Java)
- Codierungsvorschriften für Daten

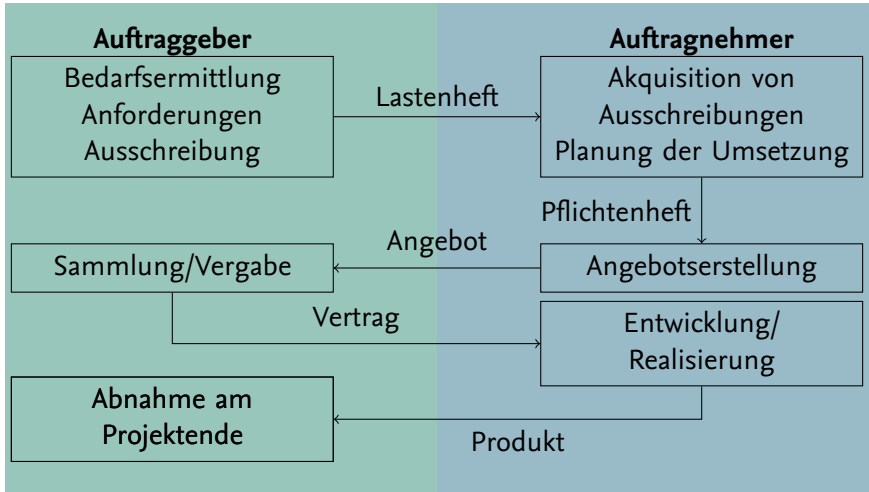
Pflichtenheft

- Definition
 - Vom Auftragnehmer formuliert
 - Legt die Vorgaben für die Umsetzung fest
 - Englische Bezeichnung: Proposal
- Abgrenzung
 - Wie möchte der Auftragnehmer die im Lastenheft definierten Anforderungen erfüllen
 - Bereitet die mögliche Umsetzung im ersten Schritt vor
 - Ermöglicht eine Abschätzung des Gesamtaufwands
 - Erleichtert die Planung eines Angebots für den Auftrag

Beispielanforderungen Pflichtenheft

- Definition der genutzten Technologien
 - Implementierung als Webapplikation unter Einsatz von verbreiteten Technologien (Javascript, HTML, CSS)
 - Festlegung einer Datenbankstruktur (MySQL, SQLite)
 - Nutzung von Onlinekartendiensten zur Datenvisualisierung
- Beschreibung der Umsetzung von Funktionen
 - Anzeige mittels PDF-Plugin im Browser
 - Nutzung von Simulationsmodellen für die Berechnung von Daten
 - Bereitstellen von Daten via Cloud-Dienst
- Festlegen von Implementierungsstandards
 - Datenübertragung in verschlüsselter Form
 - Codierung der Daten in XML
 - Implementierungsform und Dokumentationsstandards

Auftragsvergabe



Gewichtung von Projektzielen

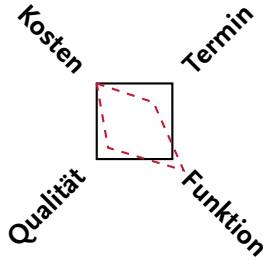


Abbildung 4: Projektziele

- Vier entgegengesetzte Ziele in der Produktentwicklung
- Kompromisse zwischen den Zielen
- Bei Änderungen sind manche Ziele schwerer zu erreichen
- Aufgabe eines Projektmanagers
 - Erreichen der definierten Ziele
 - Einhaltung eines passenden Gleichgewichts
- Maßnahmen eines Projektmanagers
 - Planung vor dem Projektstart
 - Kontrolle des Projektverlaufs

Risiko und Risikomanagement

Risiko

- Jedes Projekt birgt Risiken
- Ergebnis der Unsicherheiten eines Projekts
- Jede Unternehmung wird durch unerkannte Risiken gefährdet
- Ein Risiko ist ein Problem in der Zukunft

Risikomanagement

- Umgang mit den Risiken eines Projekts
- Generelle Unternehmensziele sind Aufgabe der Führungsebene
- Risikomanagement ist der Teil der sich mit den Abweichungen vom Ziel beschäftigt

Risiken in Softwareprojekten

- Klassisch organisierte Projekte schaffen nicht die Budgetvorgaben
 - Entwicklung wird teurer als geplant
 - Geplante Entwicklungszeiten werden überschritten
 - Die Qualität des Ergebnisses ist geringer als geplant
 - Abbruch des Projekts als letzte Konsequenz
- Beispiele aus dem öffentlichen Entwicklungssektor
 - Toll-Collect (Mautsystem), +6,9 Mrd. Euro⁵
 - FISCUS (geplante Software für Finanzämter), +4,6 Mrd. Euro¹
 - DoSV, Abwicklungssystem für Numerus-Clausus-Bewerbungen

Woher kommen diese Probleme in der Realisierung?

⁵[http:](http://www.spiegel.de/wirtschaft/soziales/grossprojekte-in-deutschland-die-top-und-flop-ten-a-1033977.html)

[//www.spiegel.de/wirtschaft/soziales/grossprojekte-in-deutschland-die-top-und-flop-ten-a-1033977.html](http://www.spiegel.de/wirtschaft/soziales/grossprojekte-in-deutschland-die-top-und-flop-ten-a-1033977.html)

Spezifische Probleme der Softwareentwicklung

- Software ist schwer messbar/beurteilbar
 - (Zwischen-) Ergebnisse sind für IT-Laien oft schwer beurteilbar
 - Es besteht ein hoher Verifizierungs-Aufwand
- Zusammenhang zwischen Anforderung und Kosten ist nicht intuitiv
 - Kleine Anforderungsänderungen können große Auswirkungen haben
 - Oftmals viele Änderungen der Anforderungen während des Projektverlaufs
- Es besteht starke Personalabhängigkeit
 - Programmierer sind nicht einfach austauschbar
 - Es bestehen erhebliche Produktivitätsunterschiede zwischen Programmierern

Ansätze zur Problemvermeidung

- Man muss das Rad nicht jedes mal neu erfinden
 - Viele Probleme wurden bereits gelöst und Lösungen sind frei verfügbar
 - Kein Zeitverlust bei der Implementierung standardisierter Lösungen
- Anforderungsbasierte Entwicklungsmodelle
 - Nur festgeschriebene (wenn auch flexible) Anforderungen umsetzen
- Weniger generische Ansätze, gezielteres Vorgehen erspart Zeit
- Dokumentation des gesamten Entwicklungsprozesses
 - Vorteil bei Personalwechsel und Weiterentwicklung
 - Vorgeschrieben bei der Entwicklung sicherheitskritischer Software
- Einbindung aller Beteiligten (Kunden, Management, Entwickler, etc.)
- Regelmäßige Updates der Anforderungen

Ständige Steuerung durch Projektmanager nötig
⇒ Risikomanagement

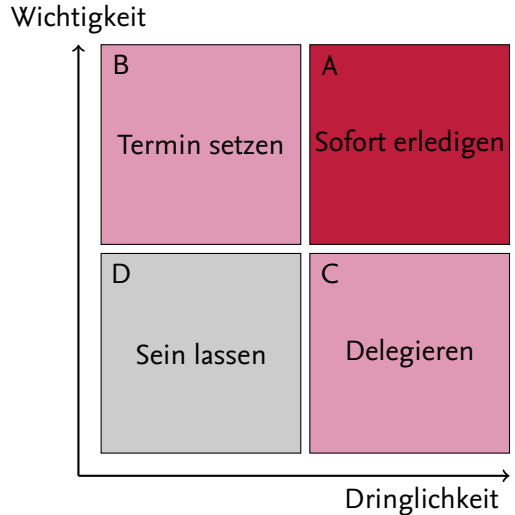
Risikomanagement

- Überwachungsprozess
Vom Entwicklungsprozess separiert aber integriert
- **frühzeitige** Erkennung und Bewertung möglicher Probleme
- Rechtzeitige Auswahl und Anwendung von Maßnahmen
- Projektbeginn: Mangel an Informationen zur Risikobewertung
- Im Projektverlauf: bessere Informationsbasis

Das Eisenhower-Prinzip

Prioritäten:

- A Aufgabe ist wichtig und dringend
- B Aufgabe ist wichtig, aber nicht dringend
- C Aufgabe ist dringend, aber nicht wichtig
- D Aufgabe ist weder wichtig noch dringend



Von den Anforderungen zur Zeitplanung

- Anforderungen generieren
- Anforderungen feiner abstufen
- TODO-Liste erstellen
- Abhängigkeiten erstellen

Zeitplanung über ein Gantt-Diagramm

- Ein Vorgang oder eine Gruppierung pro Zeile
- Ein Vorgang enthält
 - Fortlaufende Nr.
 - Bezeichnung
 - Dauer
 - Vorangegangene Vorgänge als Abhängigkeiten

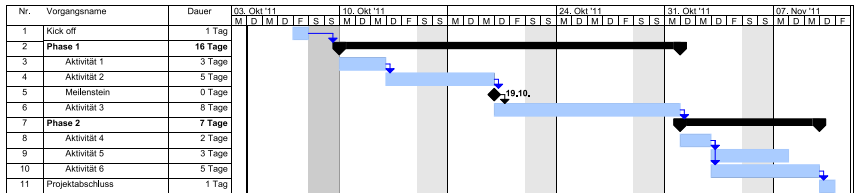


Abbildung 5: Gantt-Diagramm⁶

⁶URL:

http://de.wikipedia.org/w/index.php?title=Datei:Gantt_diagramm.svg&filetimestamp=20111004155955&.

Meilensteine und der kritische Pfad

Meilensteine sind Termine

- bei denen Liefergegenstände vorliegen können
- bei denen Prüfungen anliegen können
- bei denen weitere Entscheidungen getroffen werden

Sie werden in der Regel als Rauten abgebildet

Kritischer Pfad

- Weg durch ein Netzwerk aller Arbeitsgänge
- Verlängerung eines Vorgangs verlängert die Projektlaufzeit

Praxisdemonstration

Praxisdemonstration: MS Project und Visio

Software zur Projektplanung

- Microsoft Project
 - Für Studenten im Rahmen von DreamSpark verfügbar:
<https://www.tu-braunschweig.de/it/service-interaktiv/software/doku/msdn-aa/elms/login>
- OpenProj (<http://sourceforge.net/projects/openproj/>)
 - Import von Microsoft Project Dateien
 - OpenSource, Programmiersprache Java, Crossplattform
- TaskJuggler: (<http://taskjuggler.org/>)
 - Basiert auf einer Projektbeschreibungssprache
 - OpenSource, Programmiersprache Ruby, Linux
- GanttProject (<http://www.ganttproject.biz/>)
- Open Workbench
(<http://sourceforge.net/projects/openworkbench/>)
- Planner (<http://live.gnome.org/Planner>)

Software zur Softwaremodellierung

Die Erstellung von UML-Diagrammen kann über die folgenden Produkte erfolgen:

- Microsoft Visio
 - Für Studenten im Rahmen von DreamSpark verfügbar:
<https://www.tu-braunschweig.de/it/service-interaktiv/software/doku/msdn-aa/elms/login>
- ArgoUML (<http://argouml.tigris.org/>)
- StarUML (<http://staruml.io>)

Teil III

Projektarbeit

Bewertung der Projektmappe

Bewertung von vier Teilbereichen

1. Projektmanagement
2. Softwareentwicklung
3. Qualitätssicherung
4. Projektdarstellung

- Konzeption
 - min 2 Anforderungen wurden nach den vorgestellten Kriterien formuliert
 - Anforderungen werden im Quellcode umgesetzt
 - Ein Use-Case-Diagramm erstellt
 - **Lastenheft** erstellt
- Entwicklung nach einem Vorgehensmodell
- Zeitplanung
 - Zeitplan erstellt
 - Projektfortschritt fortlaufend dokumentiert
Wöchentlicher Turnus
- Meilensteine
 - Min 2 Meilensteine gesetzt
 - Meilensteine dokumentiert

Punkte, die von allen Gruppenmitgliedern zu bearbeiten sind, werden in rot dargestellt.

- Feinentwurf
 - **Pflichtenheft** erstellt
 - Mindestens ein Diagramm erstellt (Sequenz-, Zustands- oder Aktivitätsdiagramm)
- Versionsverwaltung
 - min 10 Commits
 - Kleinteilige Commits
Kleine Änderungen, bzw. nicht den gesamten Quellcode auf einmal in die Versionsverwaltung einpflegen
- Dokumentation des Quellcodes
 - Funktionen, Klassen und Variablen wurden sinnvoll benannt
 - Dokumentation zu Funktionen, Klassen und Variablen angefertigt

Punkte, die von allen Gruppenmitgliedern zu bearbeiten sind, werden in rot dargestellt.

- Testing
 - Zu jeder eigenen Anforderung sollte eine Testfunktion implementiert werden
 - Dokumentation der Tests
- Code-Reviews
 - Protokollierung eines internen Code-Reviews
 - Protokollierung eines externen Code-Reviews
 - Min. 2 Bug/Featurereports generiert
 - Min. 2 Bug/Featurereports bearbeitet und dokumentiert
- Programm/Skript ist im master-Branch ausführbar

Punkte, die von allen Gruppenmitgliedern zu bearbeiten sind, werden in rot dargestellt.

- Projektverlauf dokumentiert und/oder
- Darstellung des Endprodukts über ein Video, Präsentation, Screenshots...

Anmeldung der Gruppe

Wichtig:

Jede Anmeldung über GITHUB CLASSROOM kann nur einmal erfolgen.

Aufgabe 1

- **Alle Gruppenmitglieder** melden sich bei `https://github.com` an.
- **Einer der Gruppe** legt ein Team bei

`classroom.github.com/group-assignment-invitations/7f87857a5f8c192cabf8aec2b7209dc8` an.

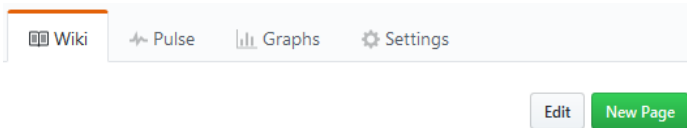
Eine Autorisierung von GITHUB CLASSROOM ist erforderlich.

- **Die restlichen Gruppenmitglieder** schließen sich dem erstellten Team über die gleiche URL an.

Lastenheft erstellen

Aufgabe 2

- Die bereits erstellten Anforderungen sollen überarbeitet werden, dass...
 - ... sie den Kriterien aus Folie 16 und 17 entsprechen.
 - ... sie fortlaufend nummeriert sind.
- Ein Use-Case-Diagramm erstellen
- Die überarbeiteten Anforderungen sollen im Lastenheft dokumentiert werden. Das Lastenheft wird in dem Wiki der Projektarbeitsgruppe auf der Seite „Lastenheft“ erstellt.



Zeitplan und Meilensteine ausarbeiten

Aufgabe 3

- Die Punkte aus dem Lastenheft sollen in den Zeitplan übernommen werden.
- Erstellung von zwei Meilensteinen pro Gruppenmitglied
- Der Zeitplan soll erstmal auf dem eigenen Rechner gespeichert werden.

Praxisdemonstration: Projektarbeit in GITHUB

Fragen?

Gibt es Fragen?

Vielen Dank für eure Aufmerksamkeit