

Programmierung von Web-Anwendungen - PM

04.07.2025

Agenda

1. Einleitung & Zielsetzung
2. Unsere Rolle als Betreuer
3. Projektmanagement – Vorgehen
4. Kommunikation & Zusammenarbeit
5. Qualitätssicherung & Code Reviews
6. Herausforderungen & Learnings
7. Verbesserungspotenzial
8. Tools & Zahlen
9. Fazit

Einleitung & Zielsetzung

Einleitung & Zielsetzung

Unsere Rolle als Betreuer

Projektmanagement – Vorgehen

Initialphase (Scoping)

Projektmanagement – Vorgehen

Laufende Betreuung

Protocol for the weekly meeting

Tasks: KW15

Mo

finished

- defined the Tools which will be used (Frontend and Backend)
- installation of project dependencies
- started implemented demo app
- started with project uml

planned

- try to finish the demo app

problems

- demo server is not running property

Daniel

finished

- installation of project dependencies
- implemented a demo app

planned

- started with init implementation

problems

- none

Kommunikation & Zusammenarbeit

- Offene Atmosphäre & regelmäßige Kommunikation
- Stets erreichbar für Fragen
- Gruppe hatte viel Eigenverantwortung
- Unsere Rolle: *Hürden früh erkennen, Diskussionen anstoßen, bei Blockaden helfen*

Qualitätssicherung & Code Reviews

Herausforderungen & Learnings

Was gut funktioniert hat

- Gute Eigenmotivation der Gruppe
- Frühzeitiges Scoping → keine Überlastung
- Gute Zusammenarbeit & Kommunikation

Herausforderungen & Learnings

Was besser hätte laufen können

- Issues & Zuständigkeiten zu spät eingeführt
- Verantwortlichkeiten nicht konsequent zugewiesen (z. B. Backend-Verantwortlicher etc.)
- Weniger strukturierte Aufgabenverteilung → teils unklare Verantwortlichkeiten
- Code Reviews nicht regelmäßig genug


Verbesserungspotenzial

- Frühzeitig GitLab Issues anlegen & strukturieren
- Eindeutige Rollenverteilung (Frontend, Backend, etc.)
- Mehr und frühere Code Reviews
- Zwischenergebnisse klarer einfordern
- Regelmäßige Reflexion des Projektstands gemeinsam mit der Gruppe

Tools & Zahlen

Tools

git-quick-stats/git-quick-stats

 Git quick statistics is a simple and efficient way to access various statistics in git repository.

 Shell  6.6k  265

boyter/scc

Sloc, Cloc and Code: scc is a very fast accurate code counter with complexity calculations and COCOMO estimates written in pure Go

 Go  7.4k  287

Tools & Zahlen

git-quick-stats

```
PS C:\Users\rajbi\repos\web-dev-tha> git quick-stats
```

```
Branching tree view:
```

```
* --+ Commit: d297c13
|   | Date:    Fri, 13 Jun 2025 15:10:24 +0200 (2 days ago)
|   | Message: add back favorite functionality (origin/29-favorite-functionality)
|   + Author:  man
|
* --+ Commit: c6c6333
|   | Date:    Fri, 6 Jun 2025 20:03:26 +0200 (9 days ago)
|   | Message: Fixing the edit page (HEAD → main, origin/main, origin/HEAD, origin/29-favorite-functionality-2)
|   + Author:  osal7613
|
* --+ Commit: 2c0fabbb
|   | Date:    Fri, 6 Jun 2025 19:54:36 +0200 (9 days ago)
|   | Message: Adding instructions for each recipe
|   + Author:  osal7613
|
| * --+ Commit: d0563c4
|/   | Date:    Fri, 6 Jun 2025 19:25:10 +0200 (9 days ago)
```

Tools & Zahlen

SCC

```
PS C:\Users\rajbi\repos\web-dev-tha> scc
```

Language	Files	Lines	Blanks	Comments	Code	Complexity
JSX	7	534	50	2	482	13
CSS	4	159	23	0	136	0
JavaScript	4	217	38	12	167	10
JSON	3	99	0	0	99	0
Markdown	2	183	71	0	112	0
HTML	1	43	0	23	20	0
Plain Text	1	3	0	0	3	0
Total	22	1238	182	37	1019	23

Estimated Cost to Develop (organic) \$27,553

Estimated Schedule Effort (organic) 3.51 months

Estimated People Required (organic) 0.70

Processed 44170 bytes, 0.044 megabytes (SI)

Fazit

- Betreuung war eine **wertvolle Erfahrung** in Führung, Kommunikation und Projektstruktur
- Ziel erreicht: Gruppe hat ein funktionierendes, gut durchdachtes Produkt geliefert
- Wichtigste Erkenntnis:
 - Coaching & Struktur bieten, aber nicht zu viel vorgeben
 - Lernprozess begleiten, nicht kontrollieren

Fragen?