

Web Engineering

Dr. Michael Lesniak
mlesniak@micromata.de

17. Juni 2019

Heute...

- › Mini-Review Übung (primär: Debug-Mode)
- › Vorstellung der Projektaufgabe
 - › Eure Ziele
 - › Timeline
 - › Meetings
 - › Technologien
 - › Scheinkritieren
 - › Die eigentliche Aufgabe: Features, ...
- › Heute starten (Ideen sammeln, Feedback einholen, ...)

Unklarheiten beseitigen, erste Absprachen treffen, Projekt im GitHub aufsetzen, ...

Review Übung

› Commit für Commit auschecken und verstehen (ggf. fragen!)

› Refactoring Backend

- › Services
- › Kleinere Methoden, Kommentare, ...
- › **Debug-Mode für Authentifizierung („ohne Token“)**

› Refactoring Frontend

- › authentication.jsx als „sauberes Objekt“
- › ...

} Optional

› Bis 08.06.2019 23:59 per Slack-Nachricht an @Michael Lesniak

- › eine konkrete Idee für ein Projekt
- › einen Wunschpartner (Vor- und Nachname) für das Projekt
- › ggf. Stärken und Vorlieben (aber: Full-Stack!)

} Pflicht
Voraussetzung für
Projekt-Teilnahme

Eure (potentiellen) Ziele

- › Schein mit sehr guter Note und entsprechende ECTS-Credits
- › Ein großes Projekt (optional: mit Quellcode) online stellen
- › Fähigkeit zur Team-Zusammenarbeit zeigen
- › Show-Case für eure Zukunft zu besitzen
- › Full-Stack-Erfahrung sammeln

Timeline

➤ 17.06 Projektvorstellung und -start

➤ 24.06 Weekly (30 Minuten vor Ort pro Team, dazwischen gerne #slack)

➤ 01.07 - Bitte Nachricht mit zeitlichen Verfügbarkeiten schreiben

➤ 08.07 - ggf. Reflektionsfragen im Channel und Live

➤ 15.07

... Selbstständige Arbeit, per #slack / Google Hangout / ... erreichbar (sporadischer)

➤ 06.09 Abgabe
- Optional: **alle** Gruppenteilnehmer müssen mir Teile des Codes erklären

➤ xx.yy Grillparty :-)

Tipps für das initiale Vorgehen

- MVP – Ansatz
 - iterativ
 - ausprobieren
 - schnelle Durchstiche
- Nutzt die **initialen vier Wochen** in der Projektphase (mit intensivem Support)
- Nutzt Slack und ein Ticket-System für Absprachen
(ggf. kann ich dann zwischendurch reinschauen und Feedback geben)

Schein-Kriterien

- Umsetzung der geforderten Features
 - Ausgleich (und Bonuspunkte) für zusätzliche Features
 - ...vielleicht auch einfach mal aus Neugier...?
- Online erreichbar über Cloud-Dienstleister
- Deployment automatisiert
 - über einen einzigen Befehl
 - Alternativ / Ergänzend: GitHub-Hooks, ...
- Lesbarer Code
 - Konsistenz
 - Kommentare („Why, not How“)
 - Saubere Schichten (siehe letzte Übung)
 - Kleine Methoden und Klassen
- Saubere git - Kommentare (... , Branches)
- Ticket-System benutzt (z. B. auf GitHub)
- README.md mit Hinweisen (optional)

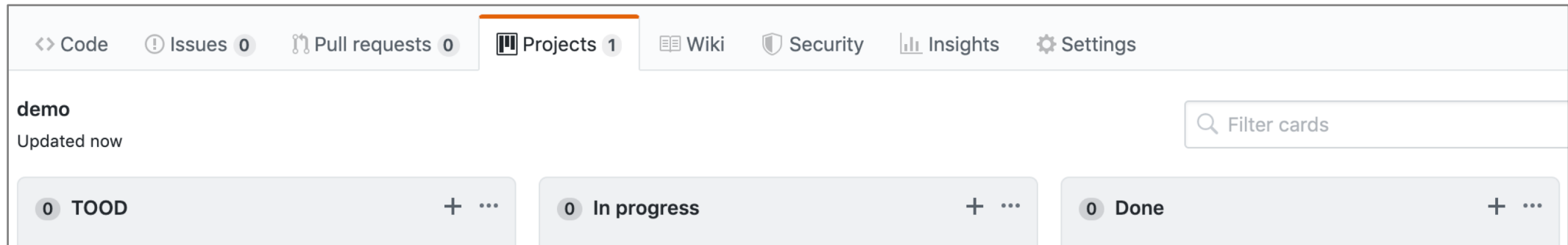
Je besser dokumentiert, desto eher kann ich wohlwollend bewerten.

Technologien

- › Minimum
 - › Backend: Java, Spring-Boot
 - › Frontend: React, (optional: Typescript)
- › Projekt muss auf meinem Rechner baubar sein
 - › ggf. Anleitung in der README.md
 - › alternativ: Docker Multi-Stage-Builds bei exotischem Setup
- › Tipp: Rad nicht neu erfinden
 - › Bibliotheken verwenden
 - › falls ihr Snippets verwendet (Stack Overflow, ...), URL im Kommentar angeben
 - › auf Lizenzen achten (mind. GPL-basierte im Paketmanager kennzeichnen)

Quellcode

- Privates Repository auf GitHub
- @mlesniak (mind.) lesenden Zugriff geben
- Kopiert nicht einfach gefundenen Quellcode – wenn so etwas rauskommt (und das tut es viel häufiger als ihr glaubt), **fällt das ganze Team durch**. Snippets sind ok (mit URL!). Im Zweifel nachfragen.
- Low-Hanging-Fruit: GitHub Project für Tickets



Kommunikation

Ein #slack-Kanal pro Team (Teamname änderbar)

Je mehr ihr über den Kanal kommuniziert, desto mehr habe ich überhaupt die Chance, euch zu helfen ;-)

Bei allen anderen Fragen, @Michael Lesniak direkt anschreiben

Kanal > Direktnachricht

Projektvorschläge

- › How long does it take to beat a game?
- › Pokemon-Datenbank
- › Webshop
- › Reddit-Klon
- › Whatsapp-Klon
- › Bucket-List
- › Kinoplatz—Reservierung
- › Yu-Gi-Oh Deck-Tester (Erweiterung)
- › GitHub-Klon
- › Chefkoch-Nachbau
- › Brot-back Support
- › Twitter-Klon
- › UI-Tool für Security Framework
- › Stundenplan Online zusammenstellen
- › Kollaborativer Kalender
- › Fußball-Manager Browser-Game
- › Online-Shop
- › Kollaboratives Wörterbuch
- › Online-Compiler
- › Maps + Yelp für Restaurants
- › Support-Tool für Lernen von WebDev

A **bucket list** is, by definition, a list of things that you want to do before you die.

Meine eigene Bucket List...

- › kann ich...
 - › erstellen
 - › löschen
 - › ansehen
 - › ändern
- › ist sichtbar für...
 - › die Öffentlichkeit
 - › mich (und niemanden sonst)
 - › ausgewählte andere User

Einträge in meiner Liste haben...

- › Titel
- › Beschreibung
- › Checkbox, falls erledigt (und wann)
- › ggf. eine Deadline
- › Beliebigen Meta-Daten (optional)
 - › Bilder
 - › Maps-Einträge („Yelp für Bucket List Einträge“)
- › Kommentare (je nach Sichtbarkeit)

Seid kreativ...

Bucket Lists anderer User kann ich...

- › anzeigen (ggf. sortieren)
- › suchen
- › kommentieren
- › in eine bestehende importieren (komplett & teilweise)
- › up- und downvoten

Authentifizierung über...

- GitHub (Pflicht)
- weitere Provider (über Bibliothek sehr einfach möglich)

Teams

Team 1

Eric Wagner
Billy Duong
Marc-Tell Lindner
Dominik Waas
Viktor Schmidt

Team 2

Jürgen Fast
Niclas Wolff
Melina Althoff
Lukas Reitz
Steve Hildebrandt

Team 3

Clemens Emme
Philipp Copei
Jonas Martin
Kevin Lichtenberg
Alexander Weidt

Team 4

Marcel Viehmeier
Tobias Schellien
Malte Fax
Nils Schönewolf
Steffen Krapf

Team 5

Lukas Hagenhauer
Omar Sood
Maximilian Dewald
Paul Plock
Michael Prasil

Los geht's ...