Vilei Vanessa
IMS2024A
28.05.2026
12.5h

# CodeSNP

CodeSNP ist eine Plattform zum Erstellen und Verwalten von Code-Snippets, entwickelt im Modul LB2 nach der Scrum-Methode. Das Projekt wurde im 4er-Team umgesetzt, wobei ich die Rolle des Scrum Masters übernommen habe.

## Ausgangslage/Aufgabenstellung

Ziel des Projekts war die praktische Anwendung der Scrum-Methode sowie die Entwicklung einer modernen TypeScript-Anwendung im Team. Das Team bestand aus vier Mitgliedern mit klar definierten Rollen (Scrum Master, Product Owner, zwei Developer). Die Anwendung sollte es Benutzer:innen ermöglichen, sich zu registrieren, anzumelden und eigene Code-Snippets zu erstellen, zu verwalten sowie über ein Dashboard einzusehen.

## Verwendete Technologien

Umgesetzt wurde die Anwendung mit TypeScript und React im Frontend sowie Node.js im Backend, mit MongoDB als Datenbank. Die Projektorganisation erfolgte über Azure DevOps (Backlog, Sprints, Boards), zum API-Testing wurde Bruno eingesetzt.

## Umsetzung/Herausforderungen

Das Team arbeitete in einem dreiwöchigen Sprint mit einer Kapazität von rund 50 Teamstunden (ca. 12.5h pro Person). Während des ersten Sprints zeigte sich, dass zu Sprintbeginn noch nicht alle nötigen Aufgaben erfasst waren, insbesondere Mockups, Unit- und End-to-End-Tests sowie zusätzliche Backend-/Datenbankaufgaben wurden erst im Sprintverlauf ergänzt, was sich im Burndown Chart als ansteigender Scope zeigte. Zudem hatte das Team anfangs wenig Erfahrung mit Azure Boards und Scrum, wodurch einige Arbeiten zuerst umgesetzt und erst später als Tasks erfasst wurden. Beim Testing (Unit-Tests mit Jest/Supertest sowie End-to-End-Tests) wurden zudem mehrere Fehler identifiziert, etwa bei der E-Mail-Validierung, der Token-Prüfung geschützter Routen und der Fehlerbehandlung bei Profil- und Snippet-Funktionen, die im Verlauf des Sprints behoben wurden.

## Ergebnisse und Erfolge

Das Ergebnis ist eine funktionierende Plattform mit Authentifizierung, Profilverwaltung, Dashboard und Snippet-Funktionen (Erstellen, Anzeigen, Aktualisieren, Löschen). Durch systematisches Unit- und End-to-End-Testing konnten mehrere Fehler in Authentifizierung, Profil- und Dashboard-Logik gefunden und erfolgreich behoben werden, was die Stabilität der Anwendung deutlich verbesserte.

## Erkenntnisse/Lernerfahrung

Aus dem ersten Sprint haben wir gelernt, dass bereits im Sprint Planning alle notwendigen Aufgabentypen möglichst vollständig erfasst werden sollten, insbesondere Planung/Mockups, Datenbank-, Backend- und Frontend-Arbeiten sowie Unit- und End-to-End-Tests. Für künftige Sprints planen wir die Tasks detaillierter, damit der Projektfortschritt im Burndown Chart genauer abgebildet wird. Als Scrum Master konnte ich zudem Erfahrung im Umgang mit Impediments und der Team-Organisation über Azure DevOps sammeln.

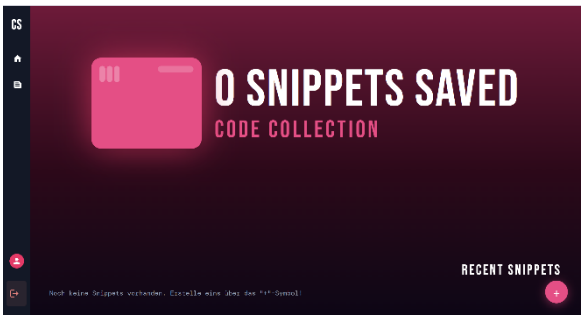


*Abb 1: CodeSNP*