

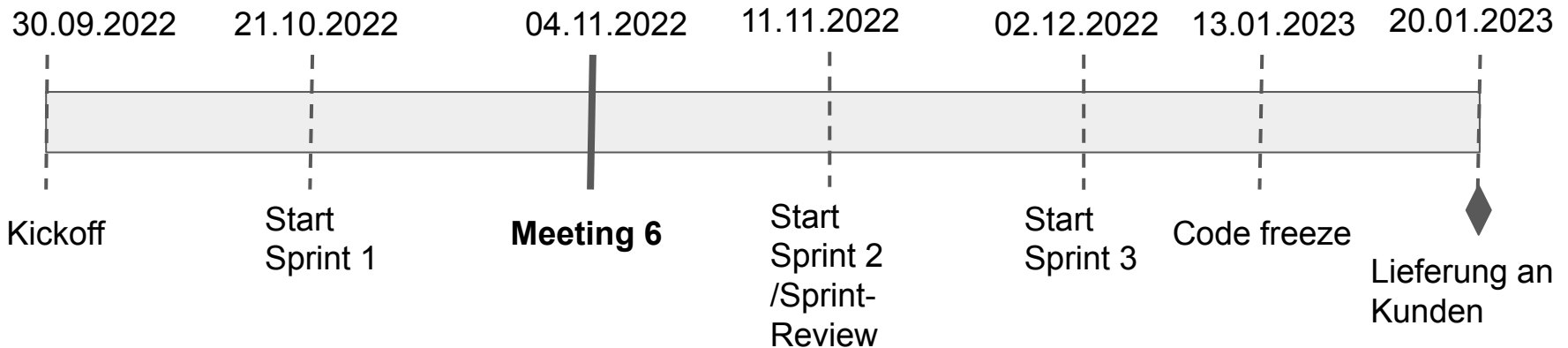
Meeting 6



Fabio Bender, Tom Keppeler, Salah Khalosi, Lukas Lesch, Sophia Nowicki, Kilian Villa, Leon Wagner

- Projektziel: Entwicklung einer Webanwendung, um Entwickler in einem geeigneten Kundenprojekt schnellstmöglich einzusetzen.

- Projekt Strategieplan:



Sprint Backlog




Aufgaben

Wird Ausgeführt

Fertig

-  **WIP-8**
↑ Manager soll Kundenprojekten erstellen 
13
-  **WIP-10**
↑ Manager soll Kundenprojekte veröffentlichen 
3
-  **WIP-9**
↑ Entwickler Profil erstellen (für Entwickler) 
13
-  **WIP-12**
↑ Anzeige aller öffentlichen Kundenprojekte (als Reviewer) 
13

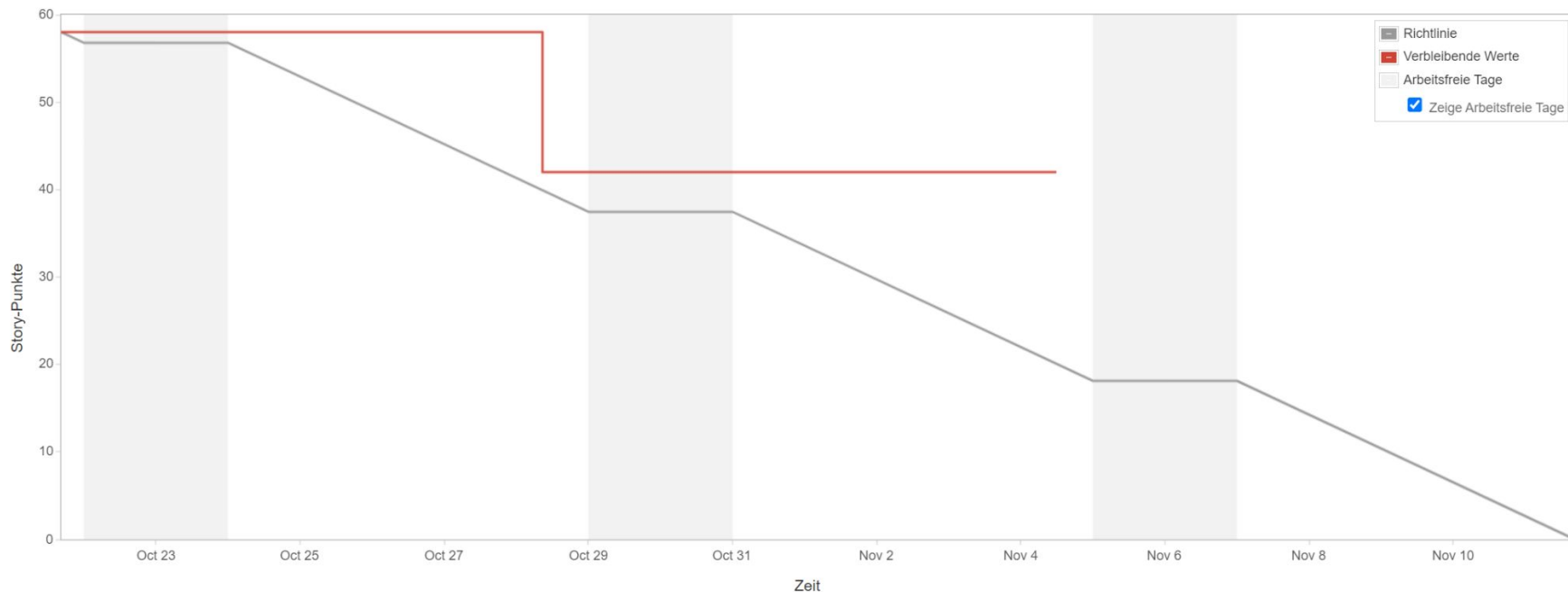
-  **WIP-5**
↑ Registrierung 
8
-  **WIP-6**
↑ Login 
5
-  **WIP-7**
↑ Logout 
3

Sprint Backlog



Burndown-Diagramm Sprint 1

Velocity: 16



Statusbericht



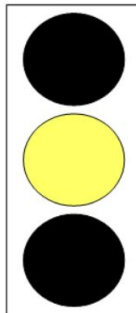
Budget:

insg. gegebenes Budget:

5.460.000 EUR

verbrauchtes Budget:

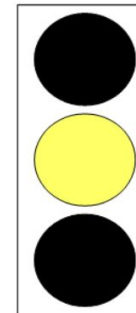
4.900.000 EUR bei 7h / Woche



Quality:

Deployment Jenkins:

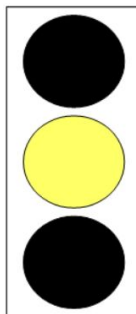
Quality-Gate SonarQube:



Scope:

Abgearbeitete SP:

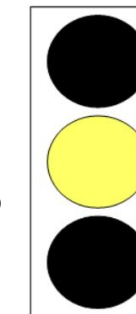
16 SP von 55



Time:

KW 44: 16/55 SP geschafft

KW 45 voraussichtlich: 55/55 SP



- Besser Koordination der Aufgaben, um Zeit zu sparen
- Code Verbesserung
 - Code umstrukturieren
 - Code kommentieren
- Testdokumentation hinzufügen

- Stories in intensiver Bearbeitung: “Manager soll Kundenprojekt erstellen” & “Manager soll Kundenprojekt veröffentlichen”
- Überarbeitetes prototypischen Mockup
- Einigung auf Architektur-Design (Layer-Pattern) und Modellierung als Baustein-Diagramm
- Software Tests

Heutige Agenda



- Artefakte vorstellen
 - aktualisiertes Mockup
 - Software Architektur
- Vorstellen eines ersten Ansatzes für die Verifikation der E-Mail
- Code Walkthrough und Demo der Use Case
 - Entwicklerprofil erstellen
 - Kundenprojekt erstellen

Mockup



<https://git.fslab.de/tkeppe2s/bsi-projekt-ws22-tbc/-/blob/main/docs/usability/mockup.pdf>

https://git.fslab.de/tkeppe2s/bsi-projekt-ws22-tbc/-/blob/main/docs/diagram/ArchitekturDiagramm_Bausteinsicht.drawio.pdf

- Verifikation mit **Spring Security Framework**
- Nutzer erhält im Registrierungsprozess ein einzigartiges, zufällig generiertes Verifikations-Token
- zusätzliches Attribut der Entität Nutzer 'enabled' ist nach erfolgreicher Verifikation true
- Einbinden der Bibliothek `javax.mail`
- ausführliches Tutorial:
<https://www.baeldung.com/registration-verify-user-by-email>
- **Konsens** des Teams: **zusätzlicher Aufwand** durch **Spike Story** nötig

To Dos



- E-Mail Verifikation implementieren
- Restliche Story Points von Sprint 1 abarbeiten
- Test-Coverage erhöhen
- Qualität verbessern (Dokumentation auch von Architekturentscheidungen, SonarQube und Jenkins)
- Sprint Retrospektive
- Sprint Planung

**Danke für Ihre
Aufmerksamkeit**

