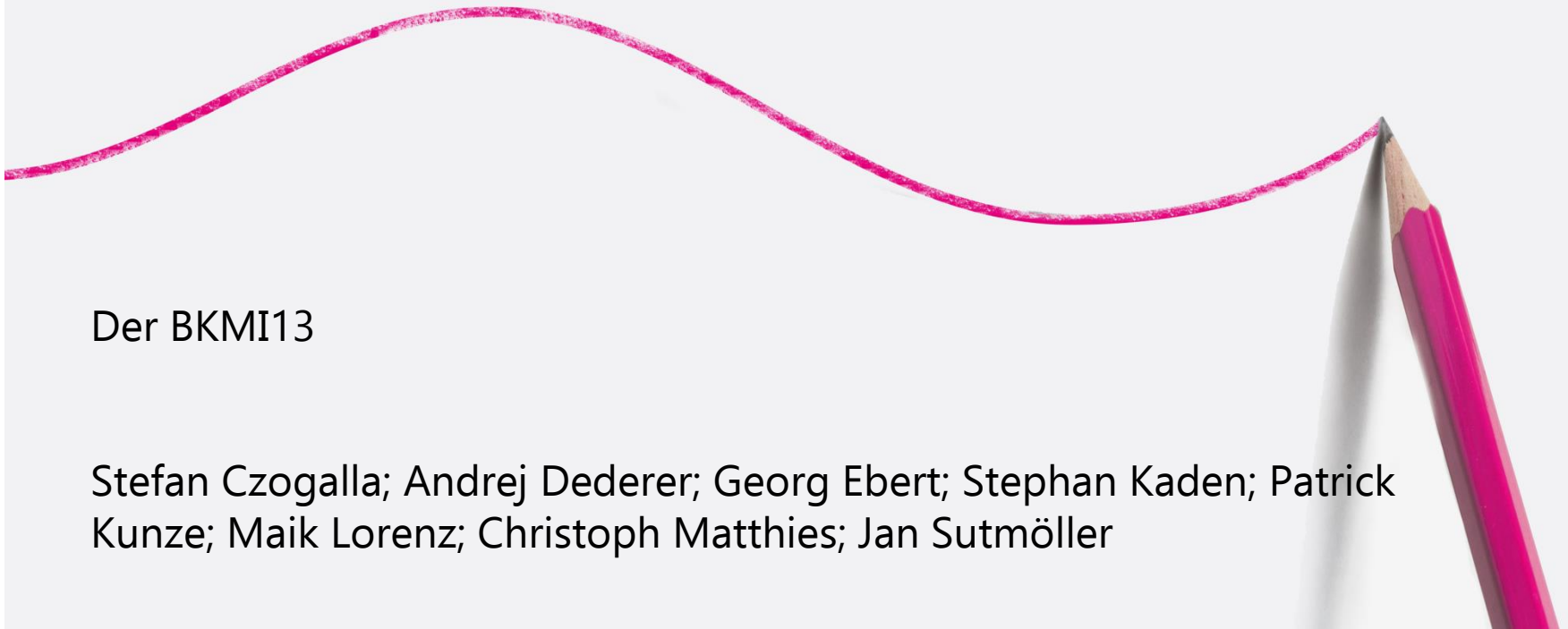


# ENTWICKLUNG EINER HFTL-APP



Der BKMI13

Stefan Czogalla; Andrej Dederer; Georg Ebert; Stephan Kaden; Patrick Kunze; Maik Lorenz; Christoph Matthies; Jan Suttmöller

# Inhalt

1. Vorgehensmodell
2. Versionsverwaltung
3. Architektur
4. Inhalte der HfTL- APP
5. Vorführung des SW Systems

# 1. Vorgehensmodell

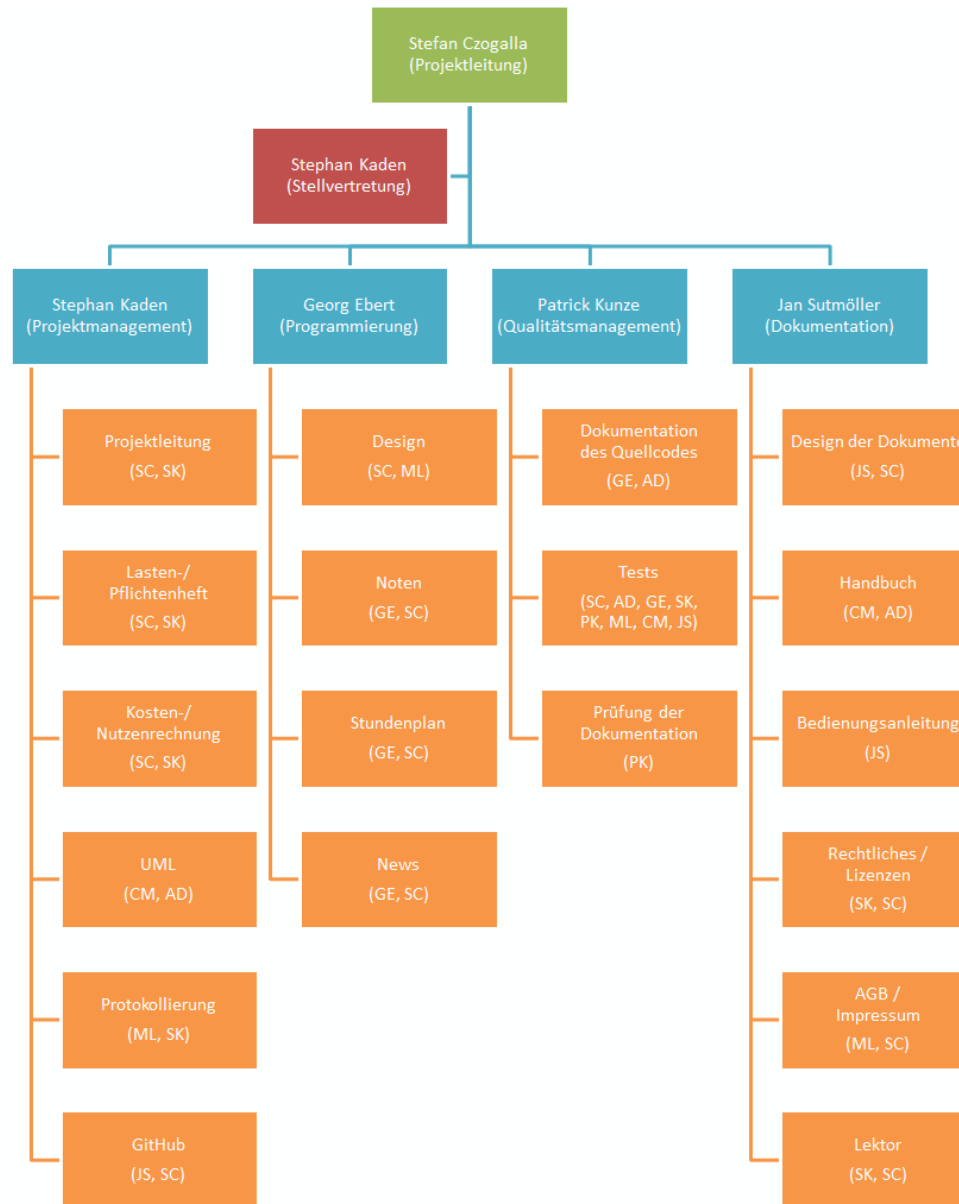
## Spiralmodell

- Geeignet für verteilte Teams
- Spezifizierter Ablauf der 4 Phasen
- Regelmäßige Prototypen
  - Durchgehende Überprüfung und Verbesserung
- Minimierung der Risiken
  - Frühes einfließen von Änderung der Spezifikationen und zusätzlichen Anforderungen
- Dokumentation durchgehend geprüft

# Umsetzung

- Festlegung der Organisation
  - Einteilung der Mitarbeiter auf die Teilbereiche
- Anforderungen und Spezifikationen festlegen
  - Lastenheft u. Pflichtenheft
- Ablauf der Zyklen
  - Prototyp erstellen
  - Testen des Prototypen
  - Überprüfung und Anpassung der Anforderungen
  - Festlegen der Spezifikationen für den nächsten Prototypen
- Durchgehende Dokumentation

# Umsetzung



## 2. Versionsverwaltung

### GitHub

- Eine Hauptversion im zentralen Repository
- Lokal sind nur Arbeitskopien davon gespeichert
- Abfrage auf Versionskonflikte bei Änderungen.
  - Schutz vor ungewolltem verändern
- Planbarkeit durch setzen von Meilensteinen
- Zuweisung von Aufgaben an bestimmte Mitarbeiter
- Protokollierung der Arbeiten
  - Was wurde wann von wem geändert.
- Rückfalloption auf eine frühere Versionen

# Umsetzung

- Repository HfTL-APP
- Ablage aller Projektrelevanten Dokumente im Repository
- Festlegen von Meilensteinen und den zuständigen Mitarbeitern
- Überführen von Änderungen mit einem erklärenden Kommentar
- Erstellen eines Releases bei Erreichen eines neuen Prototypen

# 3. Architektur

## Android

- Derzeit keine App für Android OS vorhanden
- Erreichen einer möglichst großen Abdeckung bei den Studenten
- Objektorientiert für einfache Wartung und Erweiterbarkeit.
- Standardisierungen vorhanden
- Programmierung in Java und XML
- App „Grades“ für IOS bereits verfügbar
- Andere OS sind nicht so verbreitet



# Umsetzung

- Android Studio 1.x als IDE
- Nachvollziehbare Ordnerstrukturen für bessere Übersicht
- Tests der Prototypen an „realer Hardware“ und im Emulator
- ListView für dynamische Inhalte
- Parsen der HTML-Inhalte da keine API vorhanden

# Umsetzung

- Verwendung externer Bibliotheken
  - Jsoup 1.8.1 (HTML-Parser)
  - biweekly 0.4.2 (Ical Parser)
- Trennung von Design und Funktion
- Verschlüsselung der Benutzerdaten

# 4. Inhalt der HfTL-App

## Nachrichten

- Anzeige aktueller Nachrichten der HfTL

## Noten

- Regelmäßige Abfrage der Noten je nach Einstellung
- Anzeige der Noten sortiert nach Semester
- Benachrichtigung bei Änderungen optional

## Stundenplan

- Anzeige des Stundenplanes passend zum Nutzer
- Regelmäßige Abfrage
- Synchronisierung mit dem Kalender des mobilen Gerätes

Vielen Dank für Ihre Aufmerksamkeit.