JoinSports

Gruppenmitglieder:
Timo Rautenberg, Alexander Bierenstiel, Dominik Schmitt

Datum: 20. Juni 2017

Agenda

- Vision
- Rollenverteilung
- RUP (Rational Unified Process)
- Technology
- Development Tools
- Risk Management
- Kostenabschätzung - Function Points
- Softwarearchitektur
- Patterns
- Testing
- Automatisierung
- Metrics
- Continious Integration/Lifecycle Management
- Live Demo

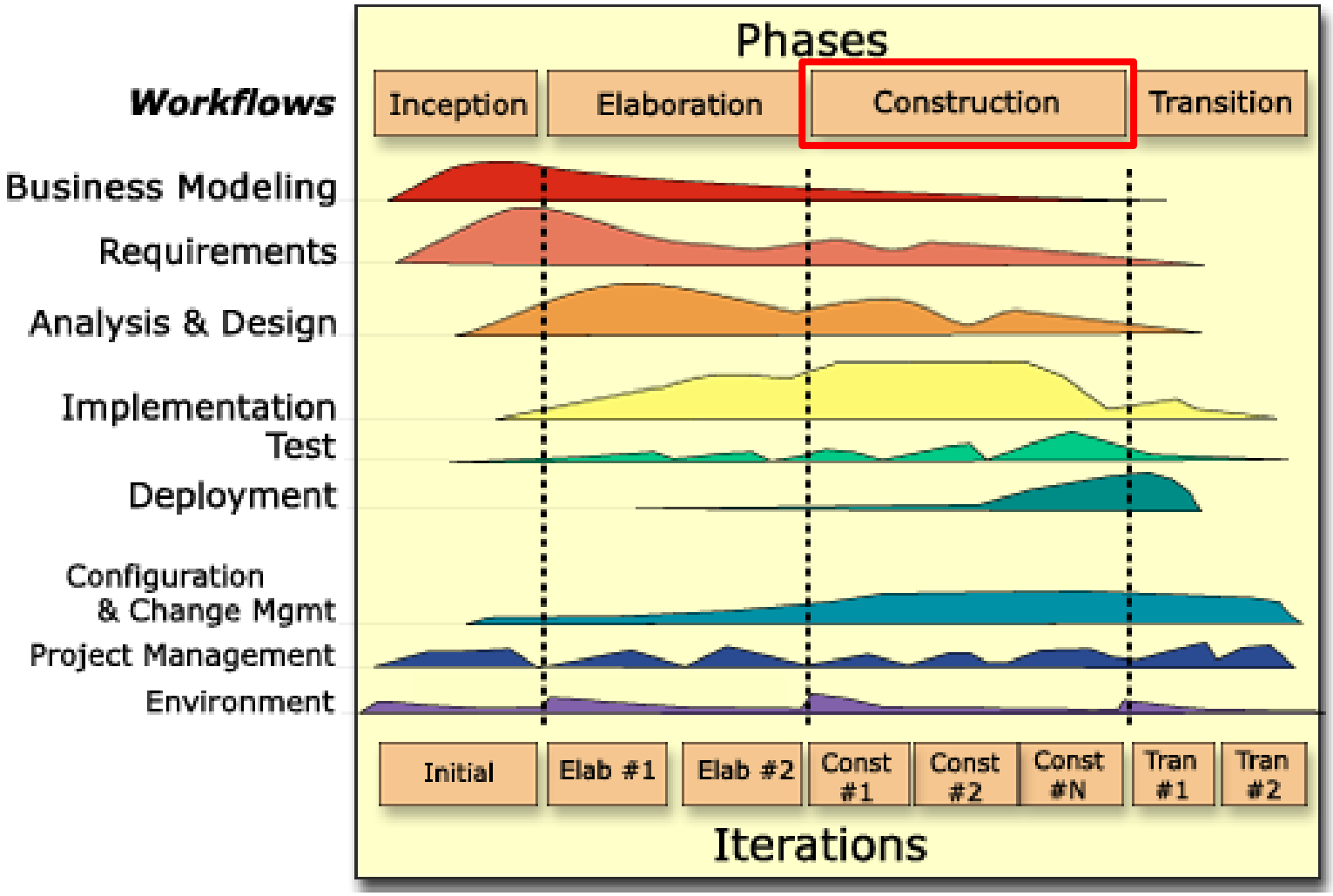
Vision

- App-Anwendung zur einfachen Vernetzung von Fußballfans
- Organisation von Matches auf öffentlichen Sportplätzen

Rollenverteilung

Timo Rautenberg	Alexander Bierenstiel	Dominik Schmitt
• Project Manager • Implementer • Tool Specialist • Graphic Artist	• Implementer • Test Designer/Tester • Tech Writer • Editor	• Implementer • Test Designer/Tester

RUP (Rational Unified Process)



Technology

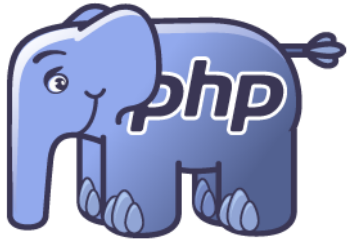
- Alle verwendeten Sprachen/Tools

Frontend

ANDROID

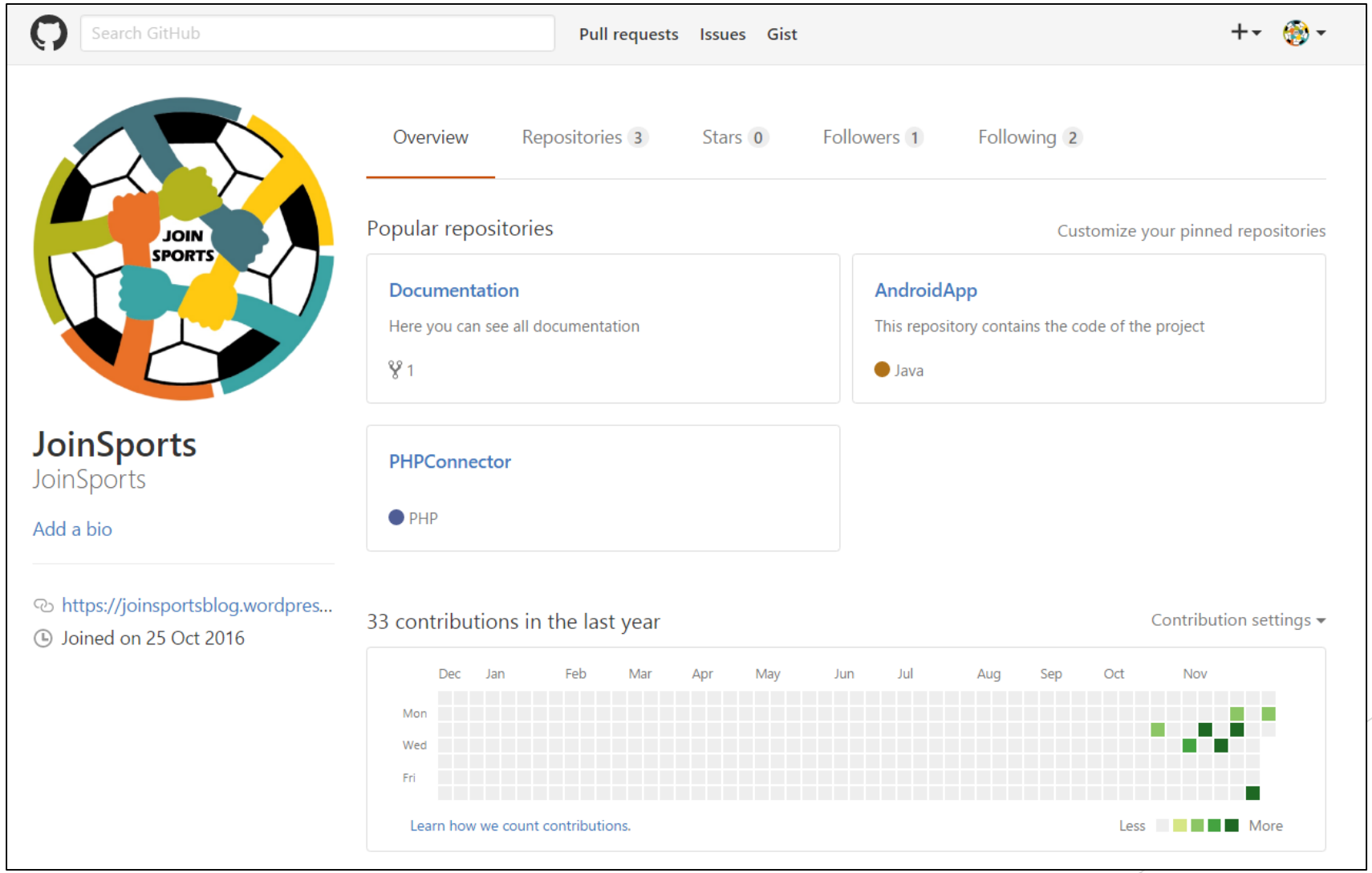
IDE

Backend



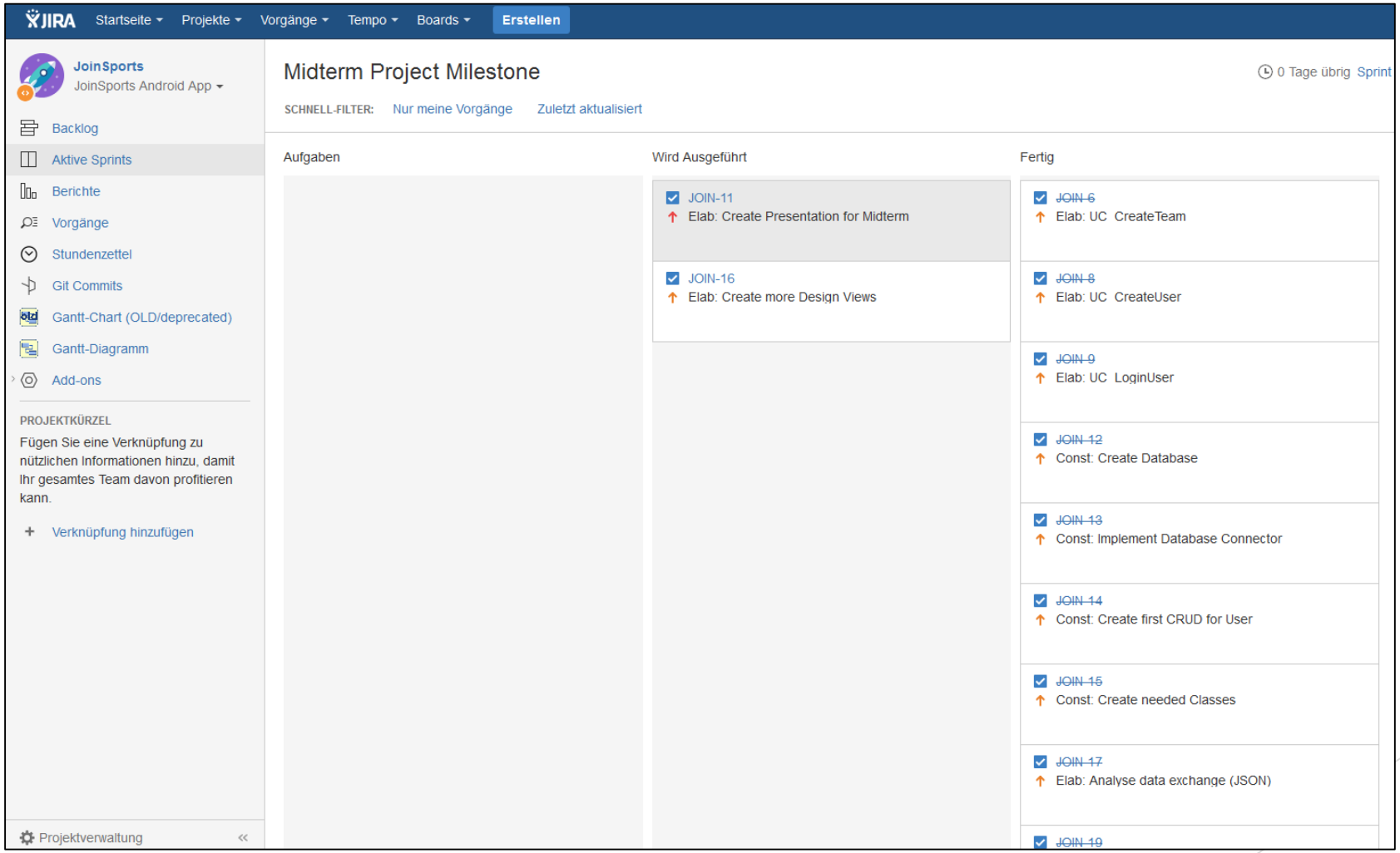
Development Tools

- GitHub als Versionsverwaltung



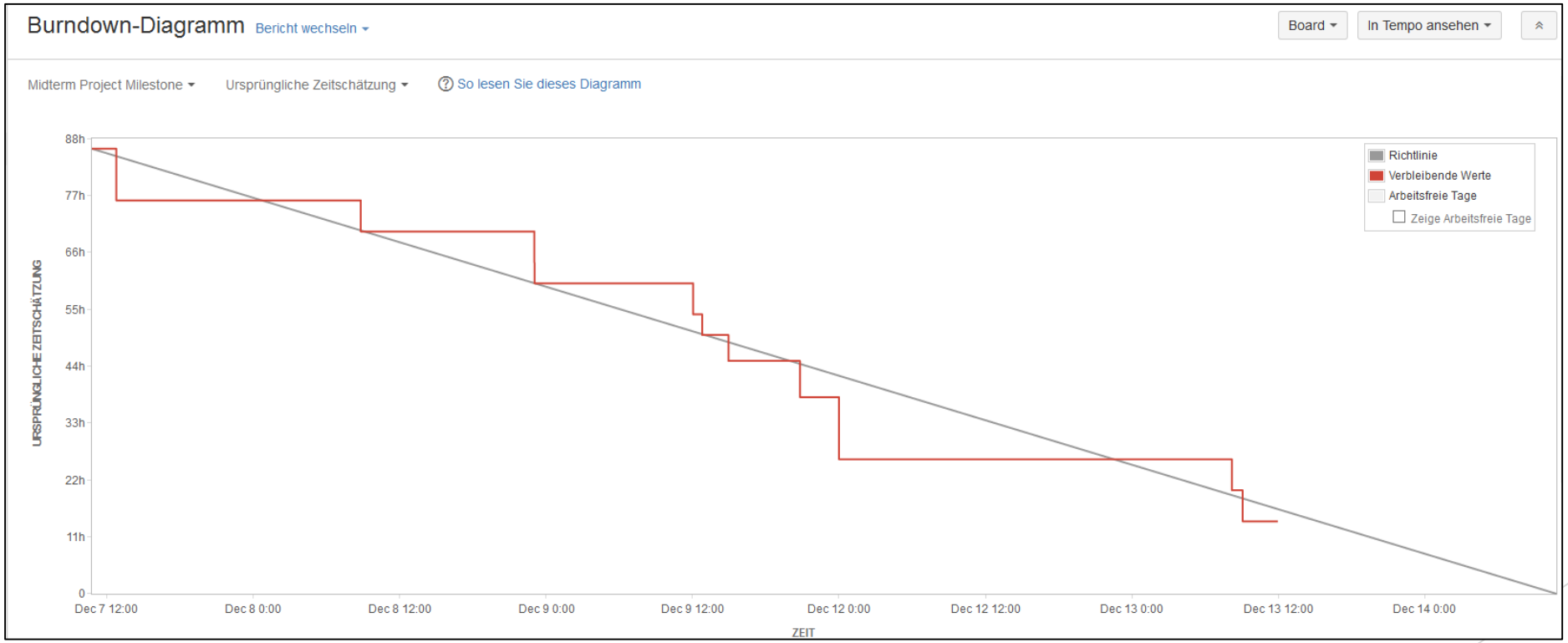
Development Tools

- Scrumming via Jira



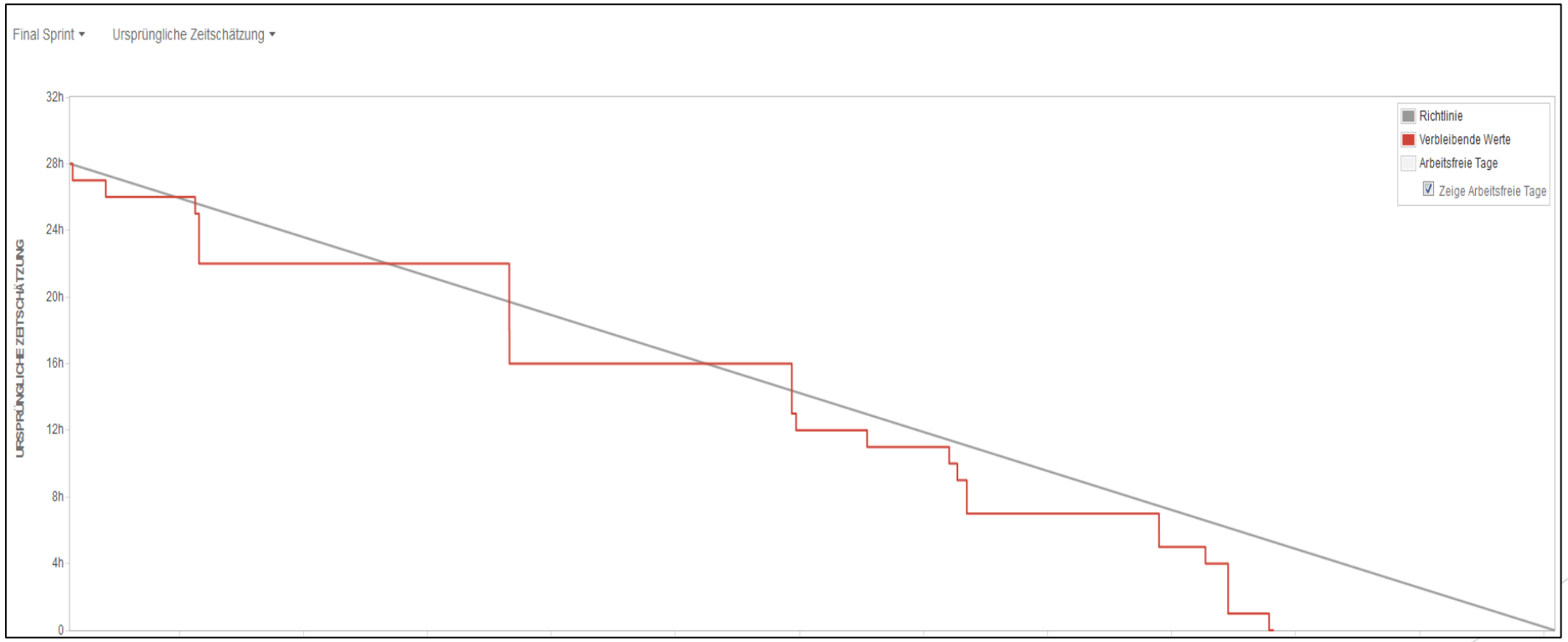
Development Tools

- Burndown Chart: Midterm Presentation



Development Tools

- Burndown Chart: Final Sprint

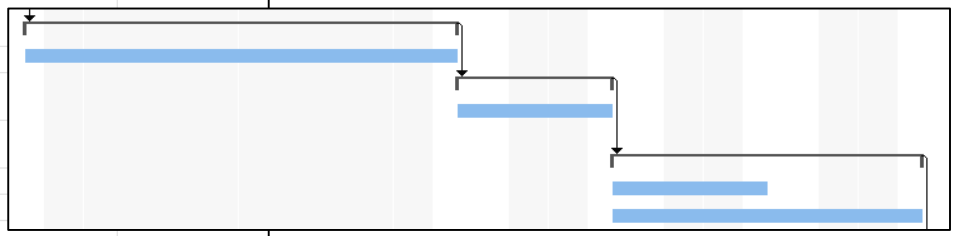


Development Tools

- Langzeitplanung via MS Project

	Vorgangsmodus	Vorgangsname	Dauer	Anfang	Ende	Vorgänger	Ressourcennamen
1		**Inception**	**5,75 Tage**	**Mit 05.10.16**	**Mit 12.10.16**		
2		**Project Managment**	**0,5 Tage**	**Mit 05.10.16**	**Mit 05.10.16**		
3		Projektidee überlegen	2 Std.				
4		Projektplan erstellen	4 Std.				
5		Rollenverteilung der Teammitglieder definieren	1 Std.				
6		**Buisness Modelling**	**0,38 Tage**	**Fre 07.10.16**	**Fre 07.10.16**		
7		Blog in Wordpress erstellen	3 Std.				
8		**Requirements**	**0,13 Tage**	**Mit 12.10.16**	**Mit 12.10.16**		
9		Entscheidung für Technology (Android App)	1 Std.				
10		**Analysis and Design**	**0,75 Tage**	**Mit 12.10.16**	**Mit 12.10.16**		
11		Diskussion über Features der App	6 Std.				
12		Diskussion über Aufbau & Design der App	4 Std.				
13		**Test**	**0 Tage**	**Mit 05.10.16**	**Mit 05.10.16**		
14		<Neuer Vorgang>	0 Std.				
15		**Environment**	**0,03 Tage**	**Mit 05.10.16**	**Mit 05.10.16**		
16		Einrichten Team-Email-Account	0,25 Std.				
17		Einrichten GoogleDocs Speicher	0,25 Std.				
18		**Configuration and Change Managment**	**0 Tage**	**Mit 05.10.16**	**Mit 05.10.16**		
20		**Elaboration**	**43 Tage**	**Don 13.10.16**	**Die 13.12.16**	**1**	
21		**Project Managment**	**0,5 Tage**	**Mit 19.10.16**	**Mit 19.10.16**		
22		Erstellung Jira Scrumboard	1 Std.				
23		Verwaltung der Projekt-Tasks in Jira	4 Std.				
24		Projektplan updaten	4 Std.				
25		Erstellen eines Longterm-Plans in MS Project	4 Std.				
26		**Requirements**	**1 Tag**	**Mit 19.10.16**	**Don 20.10.16**		
27		Software Requirement Specication Template erstellen					
28		Mockups entwerfen	2 Std.				

GANTT-DIAGRAMM



Risk Management

Risk	Description	Probability of occurence	Damage	Mitigation strategy	Person in charge	Risk factor
System outages	PHP-API is unavailble	1%	9	Redundant infrastructure	Dominik Schmitt	0.09
Bad time management	Project exceeds deadline	8%	3	Define exact time plan	Timo Rautenberg	0.24
Versioning troubles	Get trouble with Git-Repository	3%	2	Read Github tutorials	Alex Bierenstiel	0.06
Personnel deficits	Team member drops out	5%	4	Adjust time plan	Timo Rautenberg	0.2
Technical deadlock	Technology change required	20%	7	Searching for alternatives	Dominik Schmitt	1.4
Inexperience of team members	Most of the members use the technologies for the first time	60%	5	Read documentations and tutorials	ALL	3.0
Influence of the workload of other lectures	Parallel projects like PIC-Simulator, Web-Engineering, ...	30%	7	Strict Round Robin processing	Timo Rautenberg	2.1
Bugs	Throwback in time plan through bugs	50%	4	Clean coding	Alexander Bierstiel	2.0
Miscommunication	Inadequate communication among the members	5%	2	Weekly meetings	Timo Rautenberg	0.1

Kostenabschätzung – Function Points

Vergleich:

- wirklich benötigte Zeit
- geschätzte Zeit

UC	Time	FP
CreateUser	11	27,59
LoginUser	9	22,25
UpdateUser	6	31,15
DeleteUser	4	16,91
CreateTeam	6	22,25
ReadTeam	6,873568314	20,47
UpdateTeam	10,81947831	25,81
DeleteTeam	7	16,91
CreateEvent	18,05364664	35,6
ReadEvent	6,873568314	20,47
UpdateEvent	20,68425331	39,16
DeleteEvent	4,242961649	16,91
CreateMatch	16,08069164	32,93
ReadMatch	6,873568314	20,47
UpdateMatch	18,71129831	36,49
DeleteMatch	4,242961649	16,91

Kostenabschätzung – Function Points

Complexity Adjustment Table

ITEM	COMPLEXITY ADJUSTMENT QUESTIONS	SCALE No Influence 0	1	2	3	4	Essential 5
1	Does the system require reliable backup and recovery?	○	◉	○	○	○	○
2	Are data communications required?	○	○	○	○	○	◉
3	Are there distributed processing functions?	○	○	◉	○	○	○
4	Is performance critical?	○	◉	○	○	○	○
5	Will the system run in an existing, heavily utilized operational environment?	◉	○	○	○	○	○
6	Does the system require on-line data entry?	○	○	○	◉	○	○
7	Does the on-line data entry require the input transaction to be built over multiple screens or operations?	◉	○	○	○	○	○
8	Are the master files updated on-line?	◉	○	○	○	○	○
9	Are the inputs, outputs, files or inquiries complex?	○	○	○	○	◉	○
10	Is the internal processing complex?	○	○	◉	○	○	○
11	Is the code to be designed reusable?	○	◉	○	○	○	○
12	Are conversion and installation included in the design?	◉	○	○	○	○	○
13	Is the system designed for multiple installations in different organizations?	◉	○	○	○	○	○
14	Is the application designed to facilitate change and ease of use by the user?	○	○	○	○	○	◉

Software Engineering
Tiny Tools

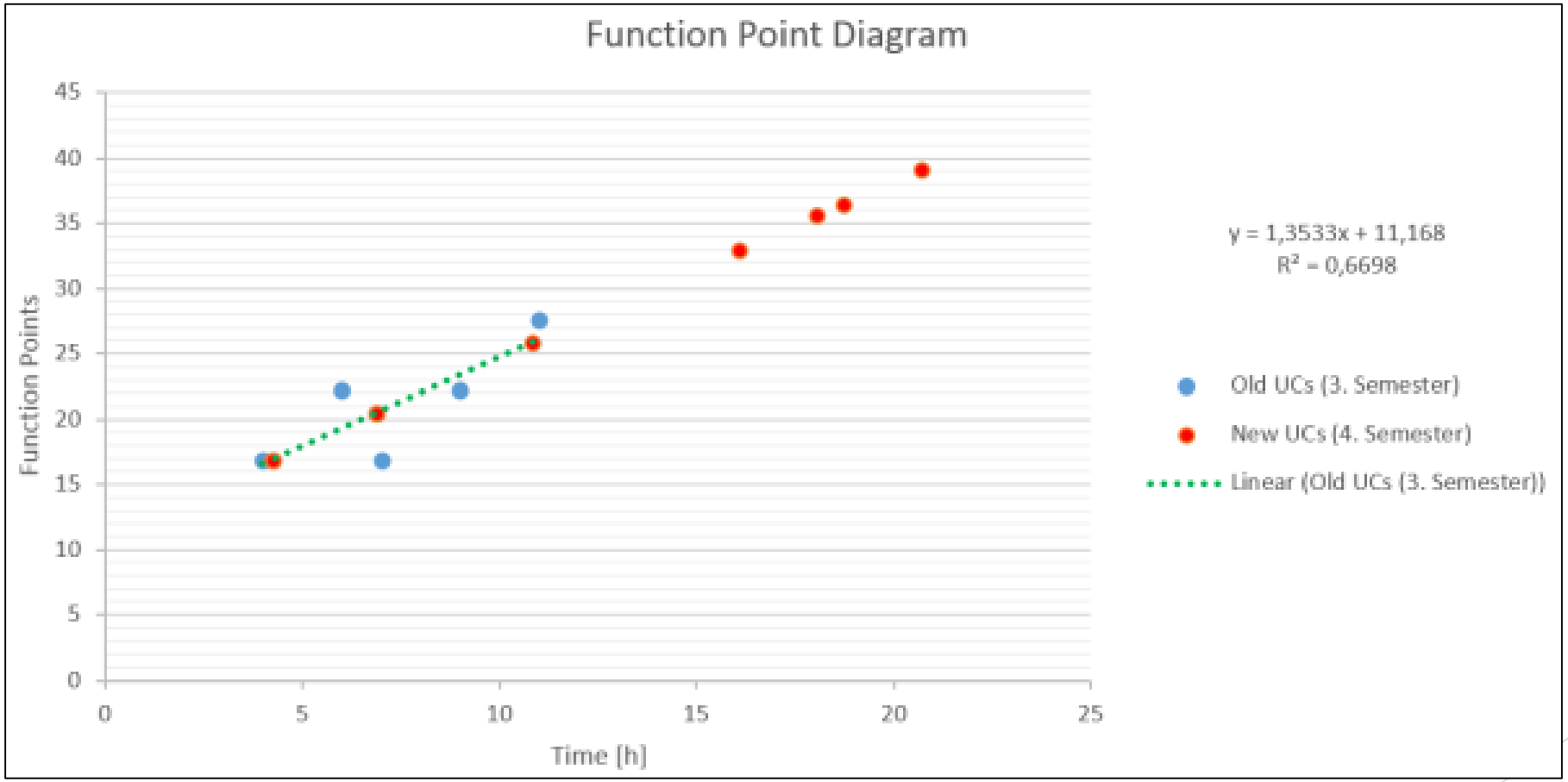
Kostenabschätzung – Function Points

Domain Characteristic Table

MEASUREMENT PARAMETER	COUNT (value >= 0)	WEIGHTING FACTOR Simple	Average	Complex
Number of User Input	6	◉	○	○
Number of User Outputs	1	◉	○	○
Number of User Inquiries	2	◉	○	○
Number of Files	1	◉	○	○
Number of External Interfaces	1	◉	○	○

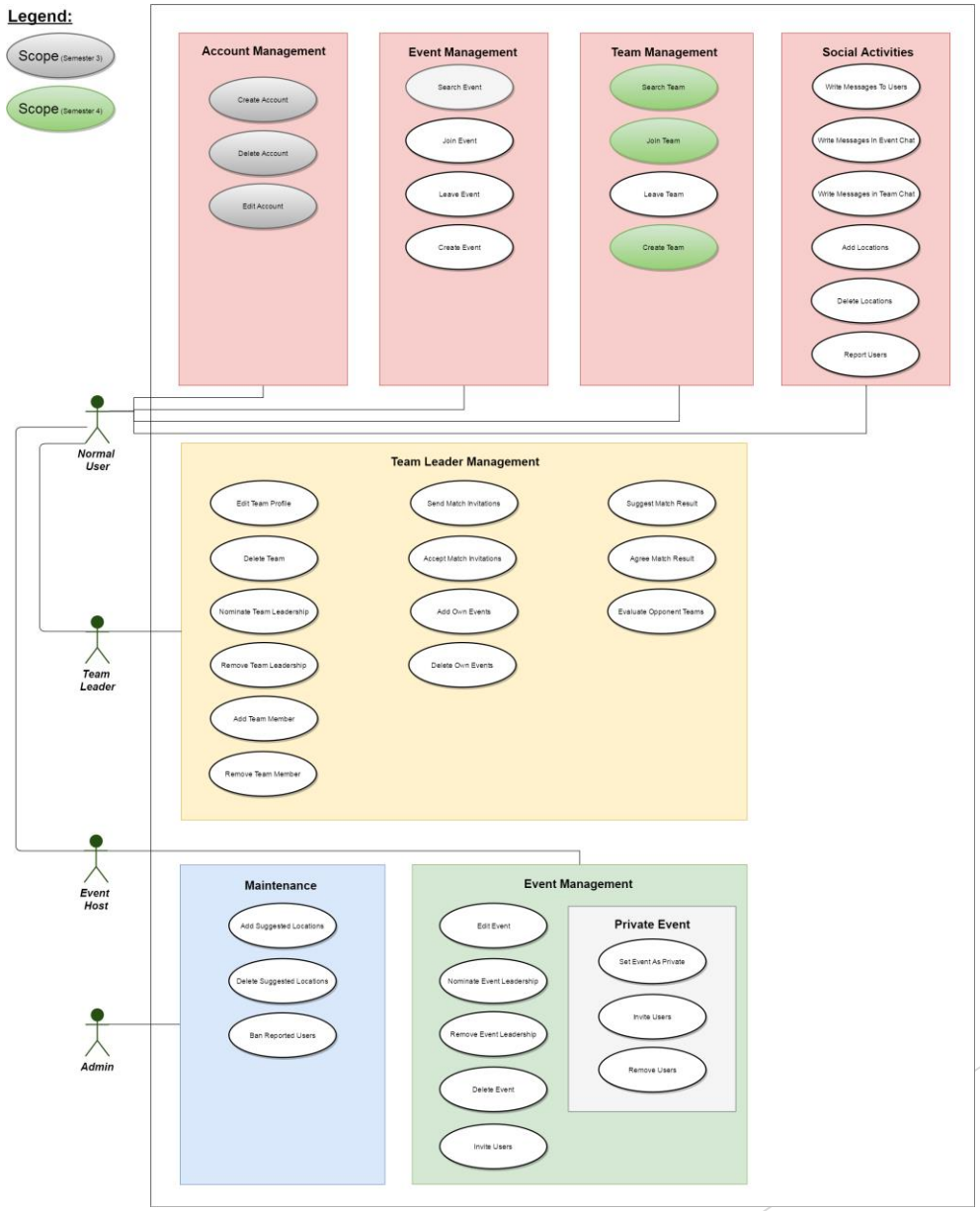
Complexity Adjustment Table | FP Calculation

Kostenabschätzung – Function Points



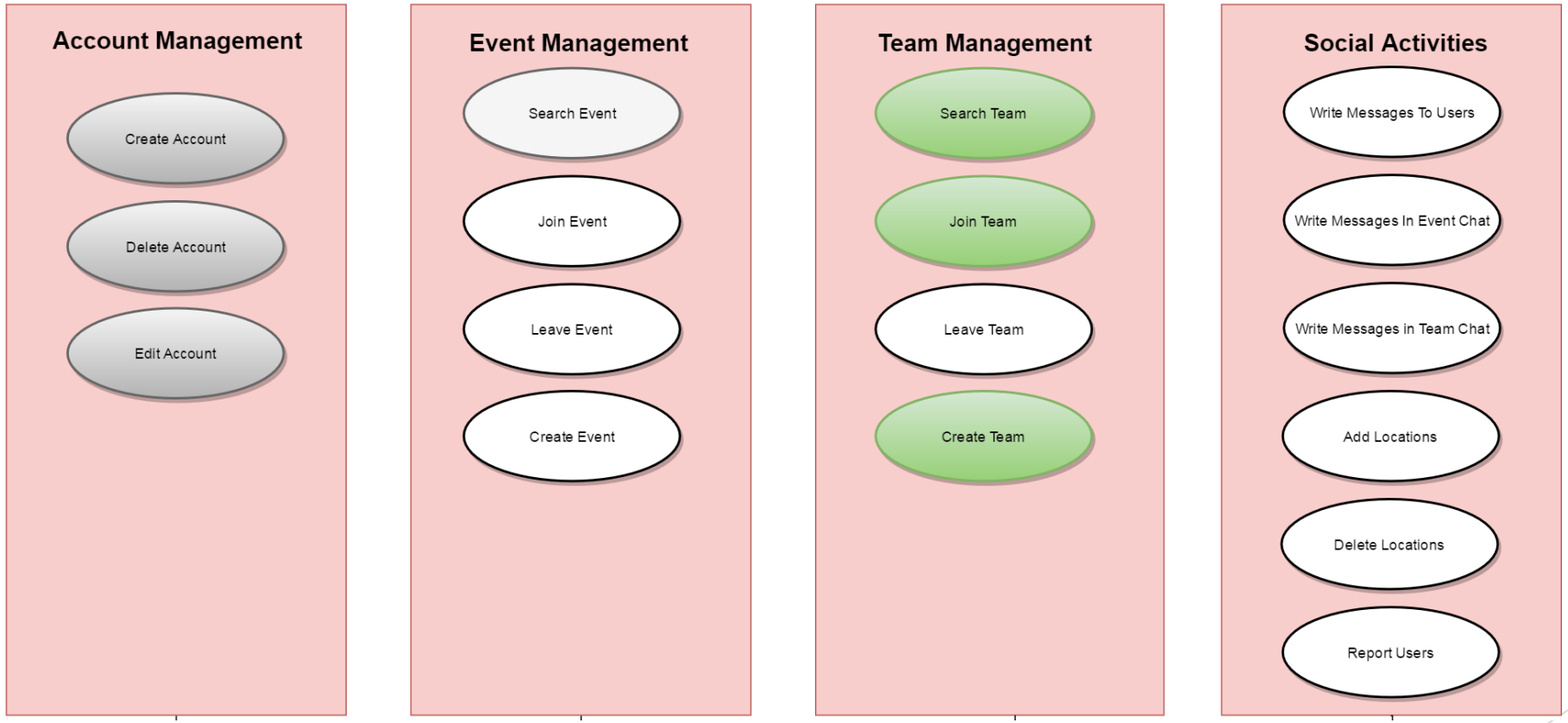
Softwarearchitektur

- UC Diagramm



Softwarearchitektur

- UC Diagramm

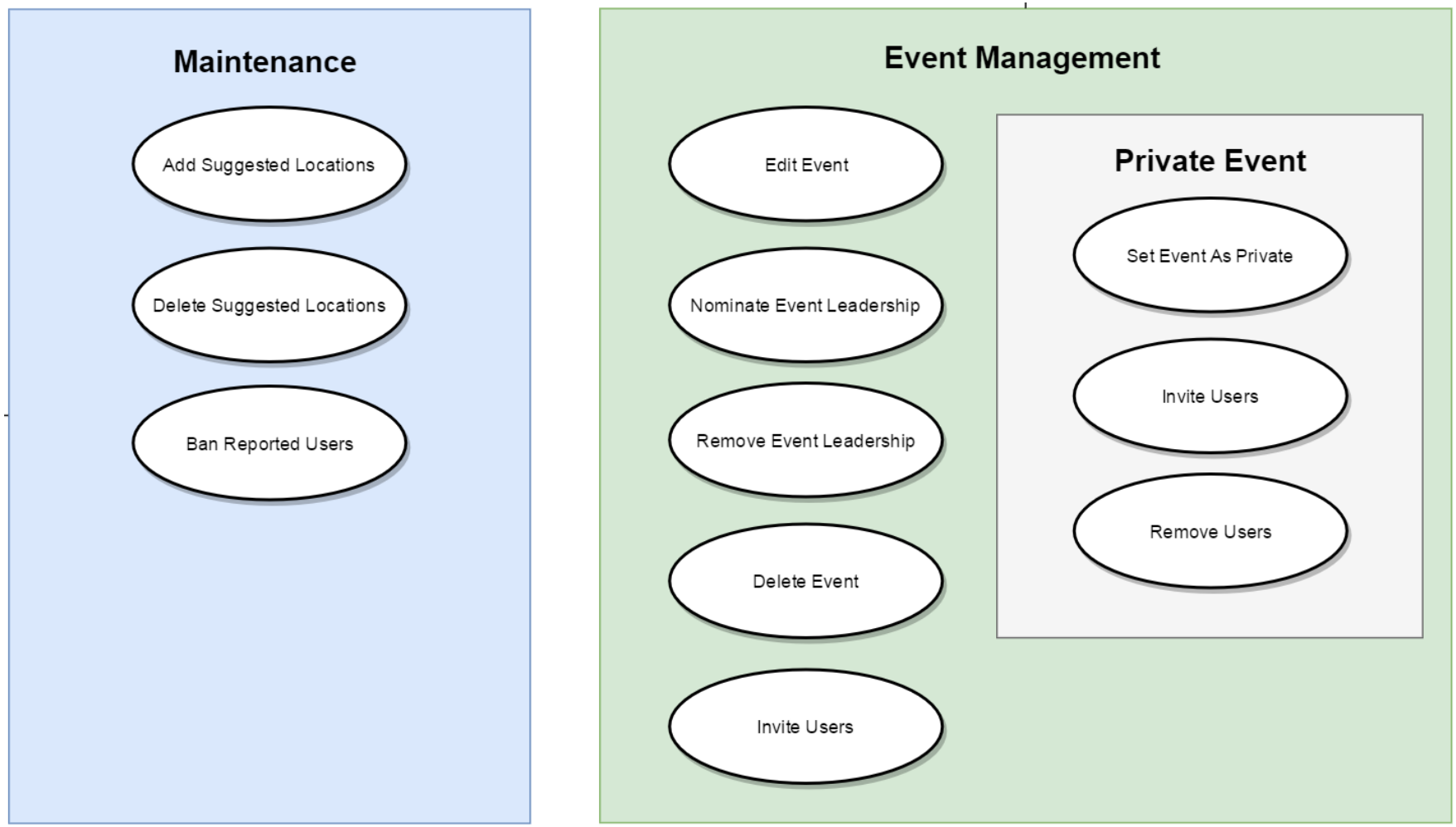


Softwarearchitektur

- UC Diagramm

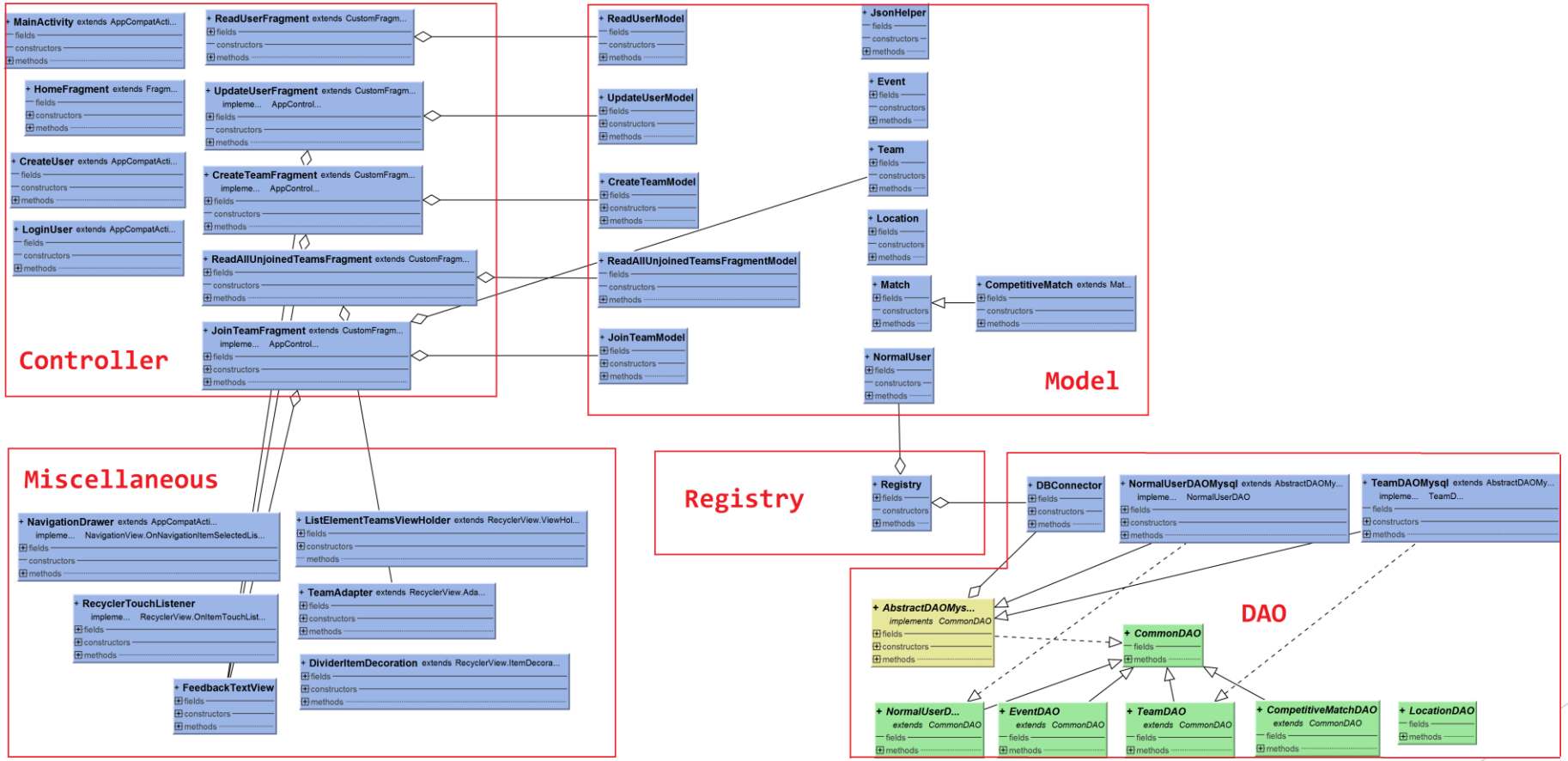
Softwarearchitektur

- UC Diagramm



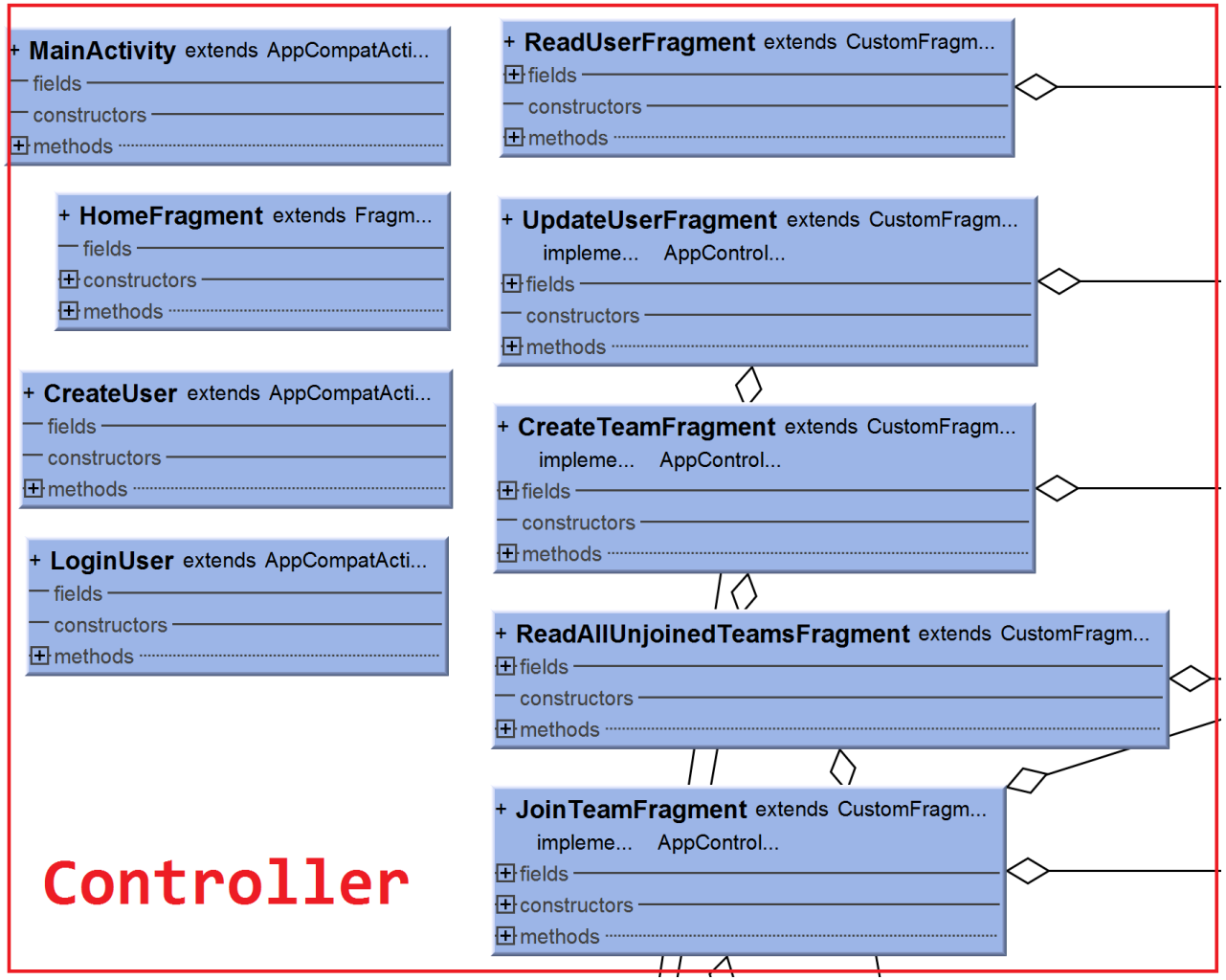
Softwarearchitektur

- Klassen-Diagramm (UML)



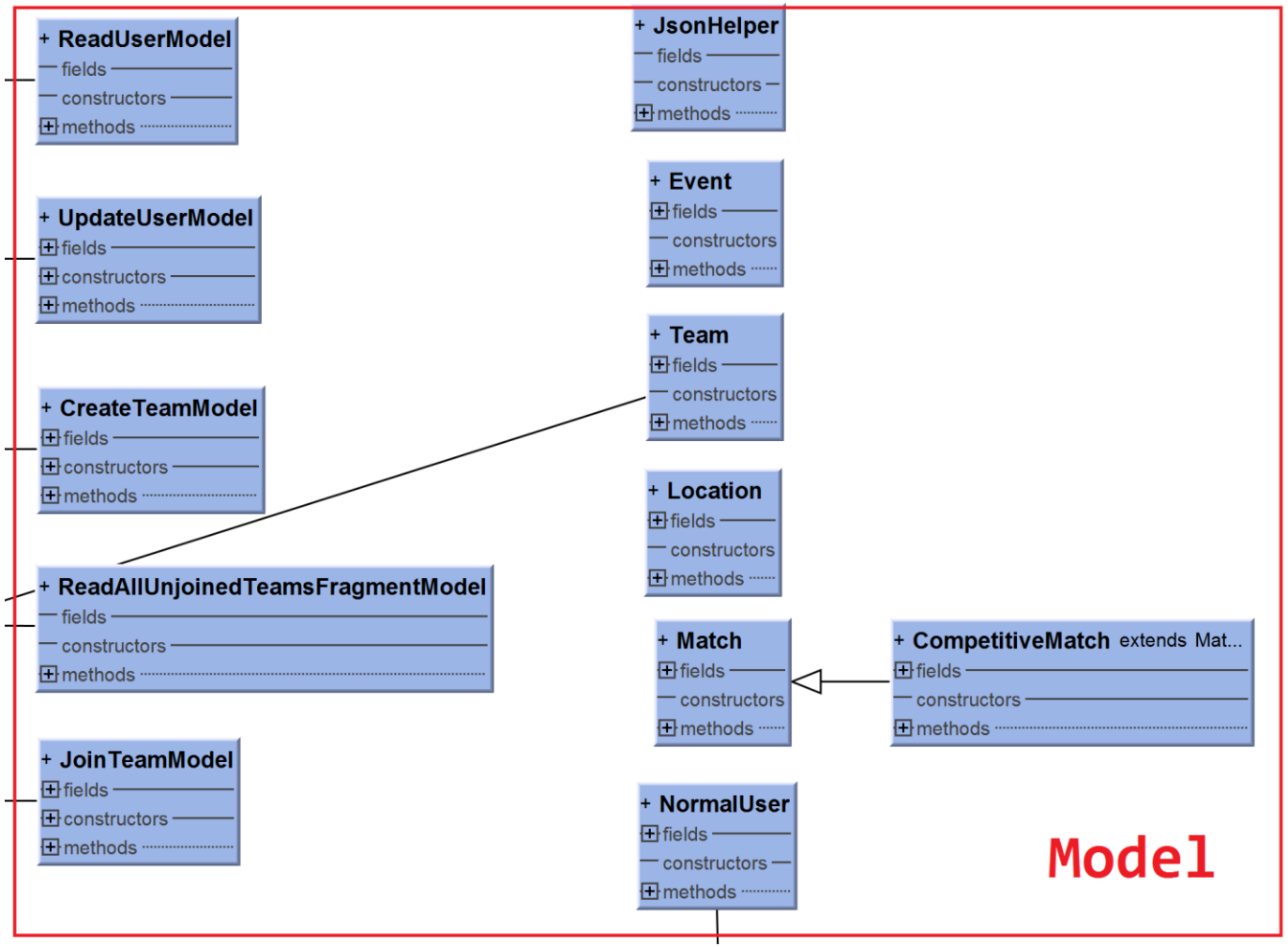
Softwarearchitektur

- Klassen-Diagramm (UML)



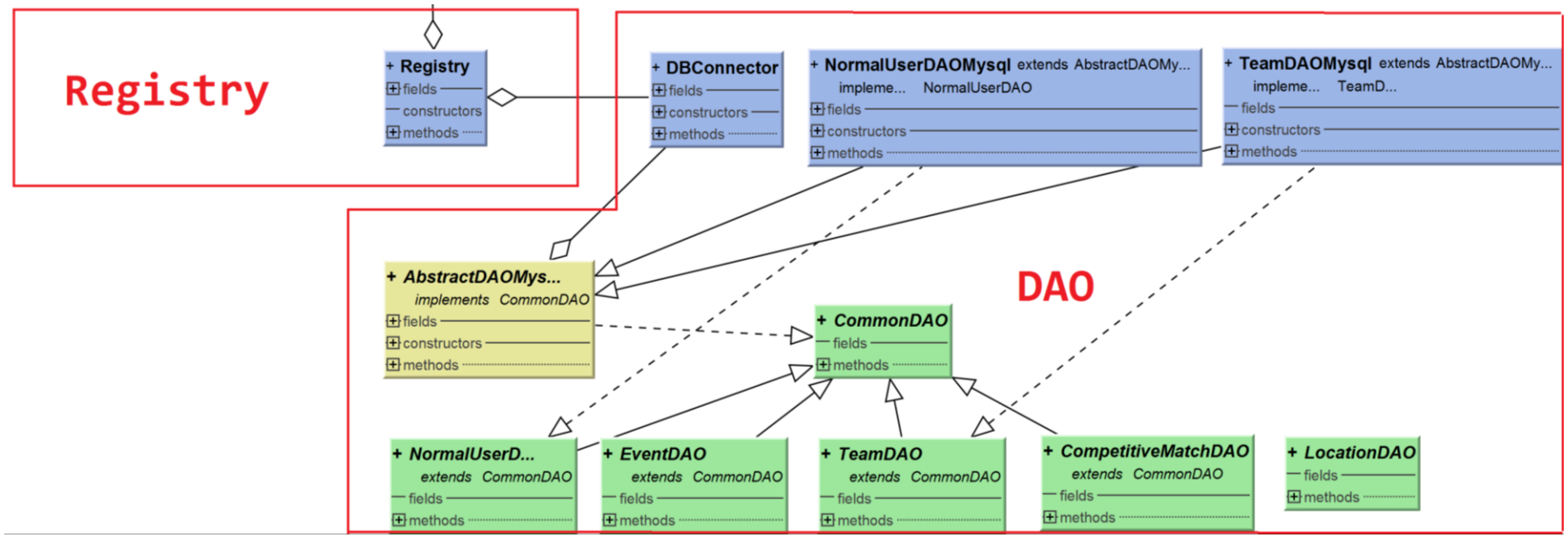
Softwarearchitektur

- Klassen-Diagramm (UML)



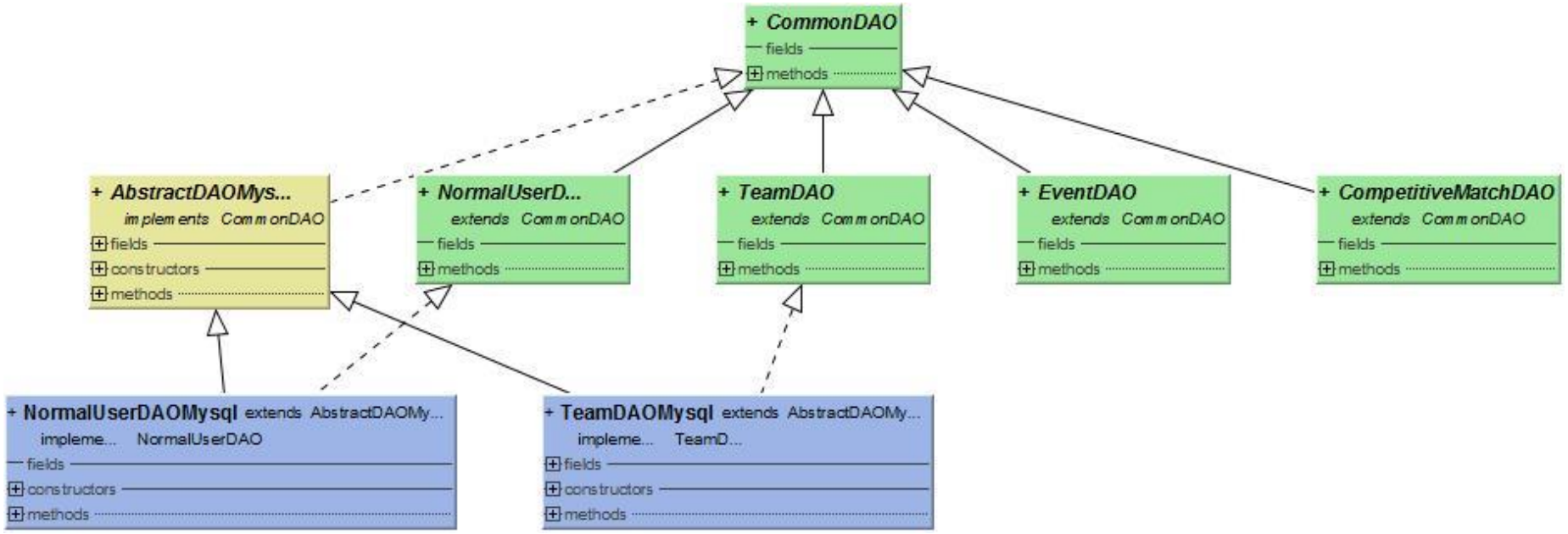
Softwarearchitektur

- Klassen-Diagramm (UML)



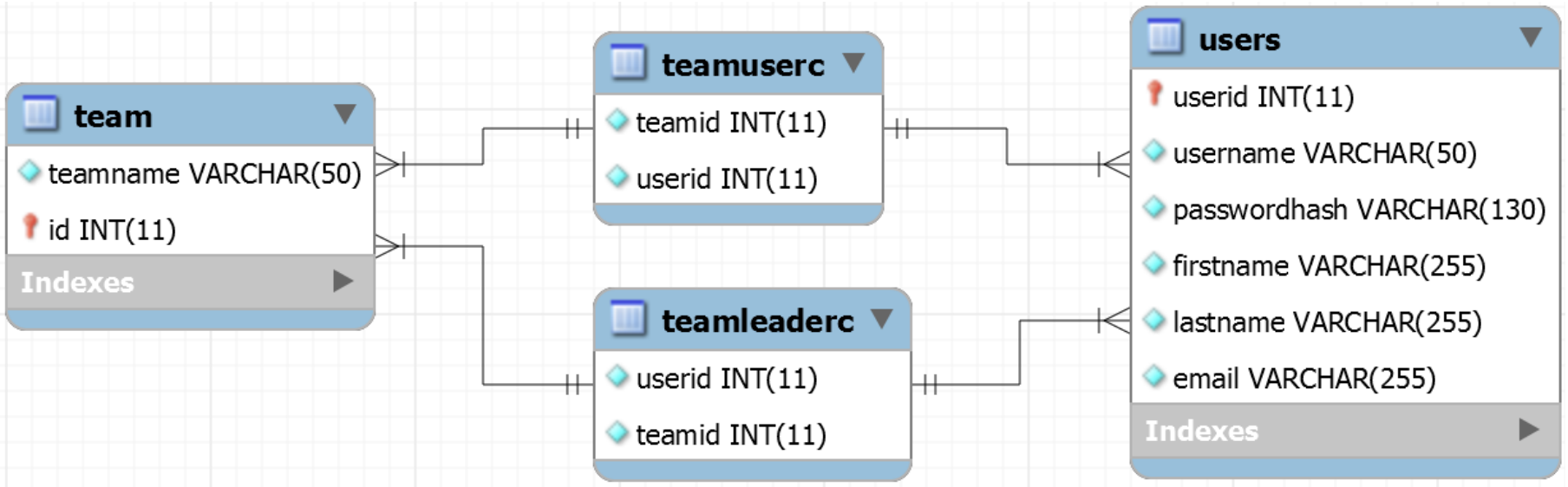
Patterns

- DAO (Data Access Object)



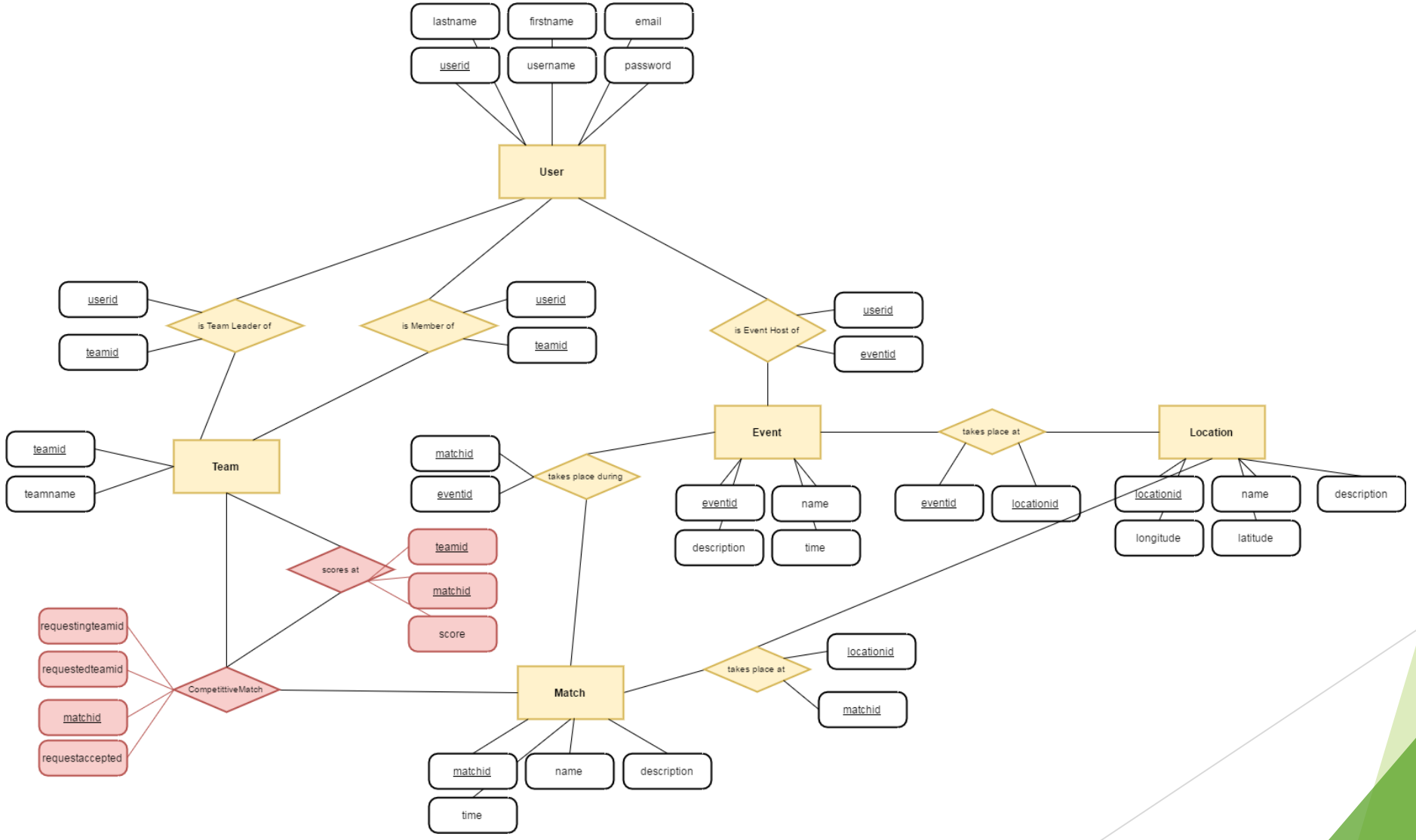
Softwarearchitektur

- Entity-Relationship-Modell (ERM) → MySQL



Softwarearchitektur

- Entity-Relationship-Modell (ERM)



Testing

- Espresso UI-Tests

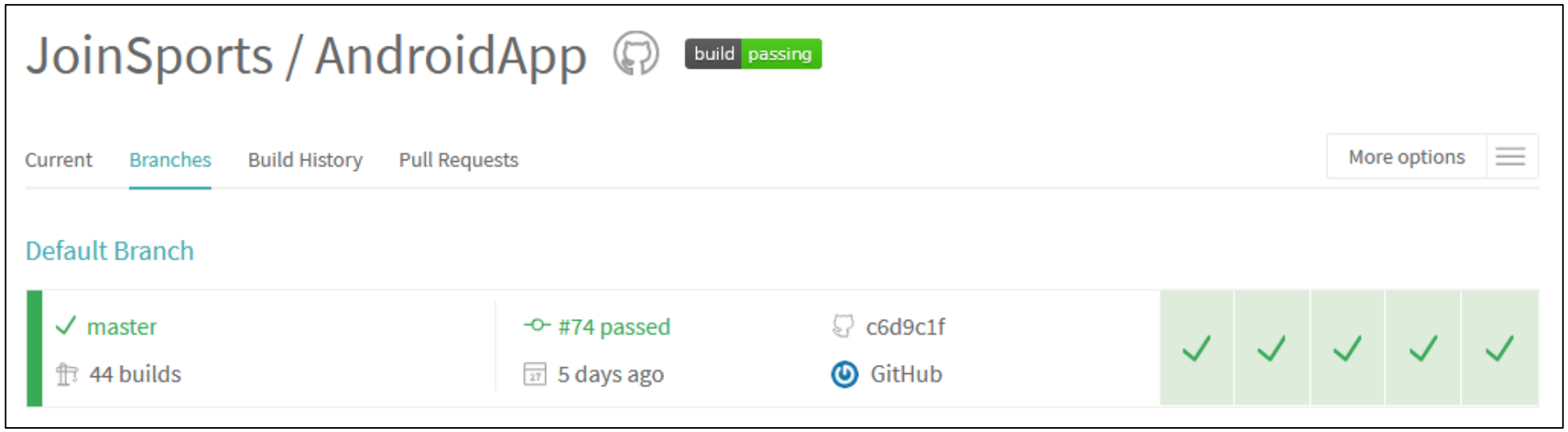
- JUnit Unit-Tests

- User-Tests
- Auswertung mit Coveralls

- Automatisiertes Tests mit TravisCI

Automatisierung

- TravisCI

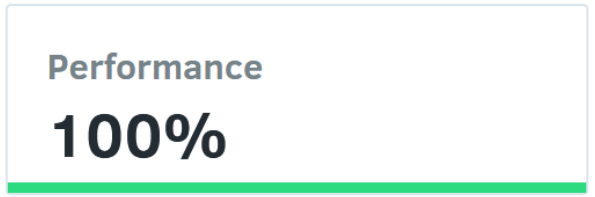


Metrics

- Codacy

CODACY

COMPLEXITY
31
30
15
15
12
12
11



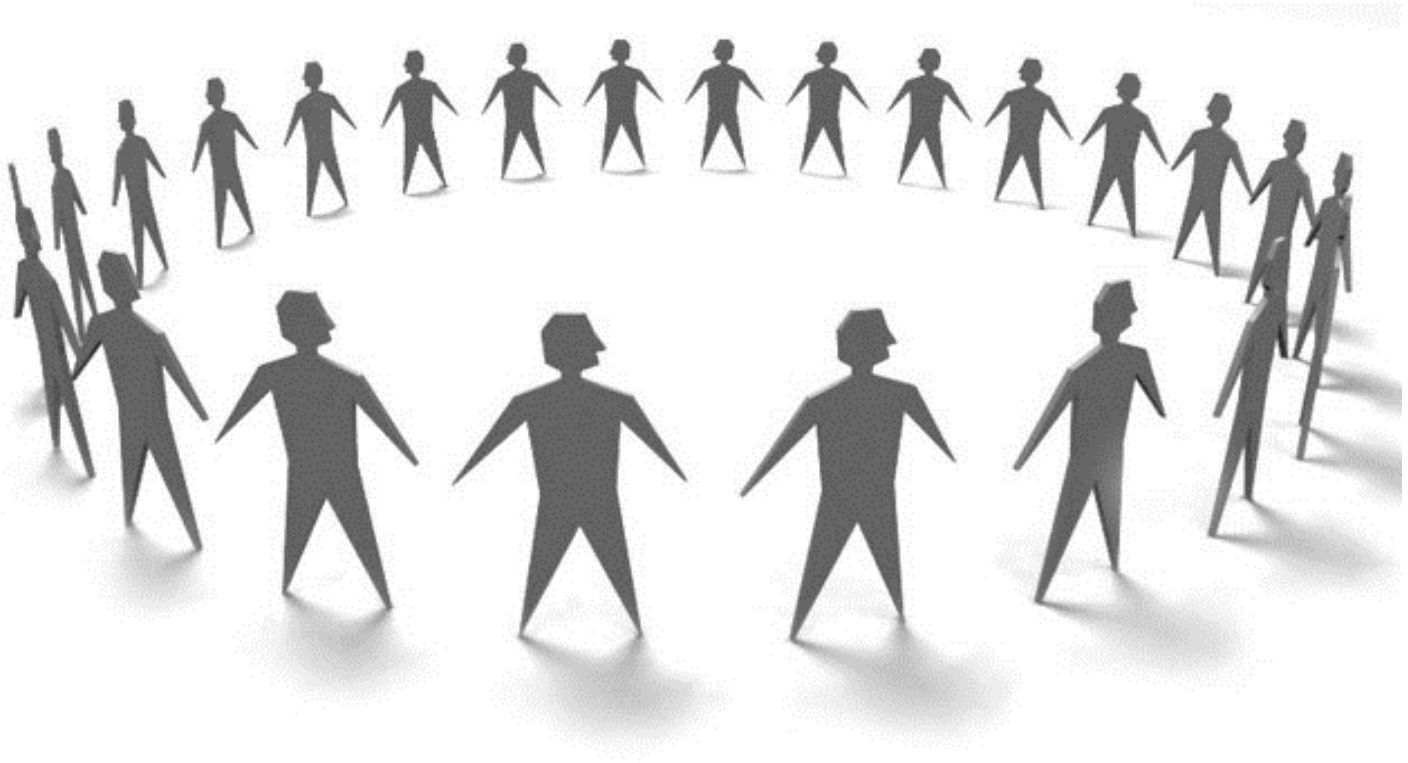
Continious Integration/Lifecycle Management (Automatic Deploy)

- TravisCI
- Möglichkeit App automatisch in den Google Play Store zu stellen

Live Demo


1:23
JoinSports
Herzlich willkommen
bei JoinSports!
JOIN
SPORTS
USER ERSTELLEN
LOGIN

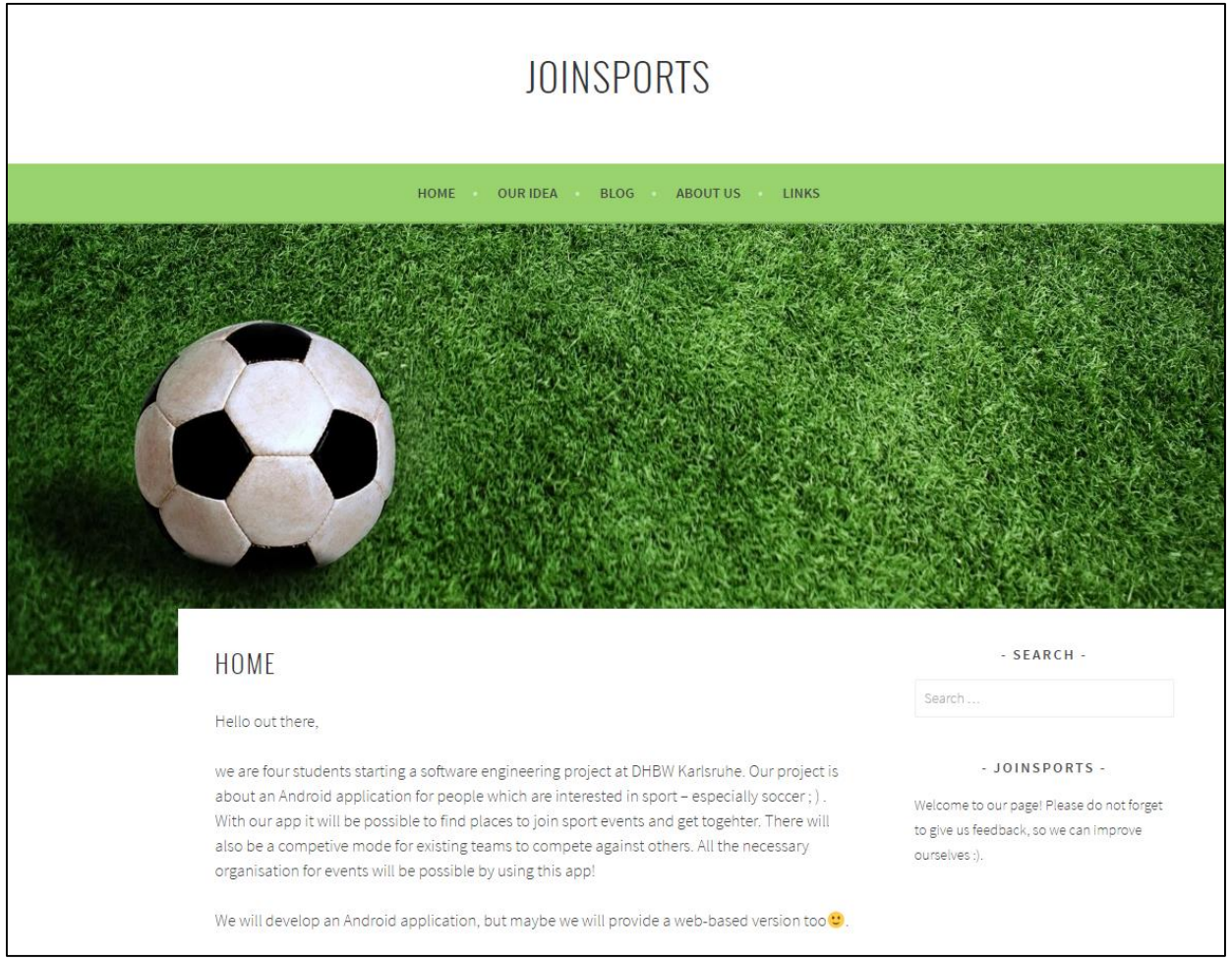
Warum wir ?



Blog-Site

- Bleiben Sie auf dem aktuellsten Stand & besuchen Sie unseren Blog!

  https://joinsportsblog.wordpress.com/



Vielen Dank für Ihre Aufmerksamkeit!