

M. Vinković, D. Kuryshov, G.  
Pea, J. Barre, C. Cardinale, E.  
Siladi

## Alpha Presentation

# Plunner



# Outline

- **Project recap**
- **Current status**
- **Scrum**
- **Communication**
- **Testing and integration**
- **Demo**

# Background

Planner is an application that helps to plan and manage meetings.

## Target customer:

- Small and medium business companies

## Qualities:

- Simplicity
- Automation
- Optimization

# Team

## FROM POLIMI



Claudio Cardinale

claudio.cardinale  
@mail.polimi.it



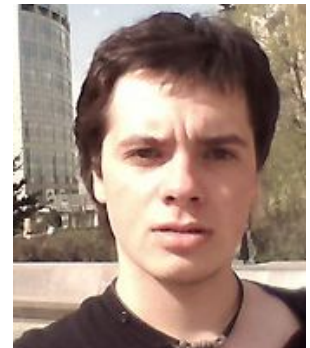
Giorgio Pea

giorgio.pea  
@mail.polimi.it



Jean Barre

jeanperrin.barre  
@mail.polimi.it



Denis Kuryshov

jaibird2493  
@gmail.com

# Team

FROM MDH



Mihovil Vinkovic

mihovilvinkovic  
@gmail.com



Emil Siladi

emil.siladi  
@gmail.com

# Main requirements

## Actors:

- Company
- User
- Planner

## A user must be able to:

- Login and recover his credentials
- See planned/to be planned meetings
- Join meetings
- Upload/import schedules
- Indicate free time slots

# Main requirements - 1

## **A company must be able to:**

- Login and register
- Register and manage users (employees)
- Create and manage groups of users
- Recover its credentials

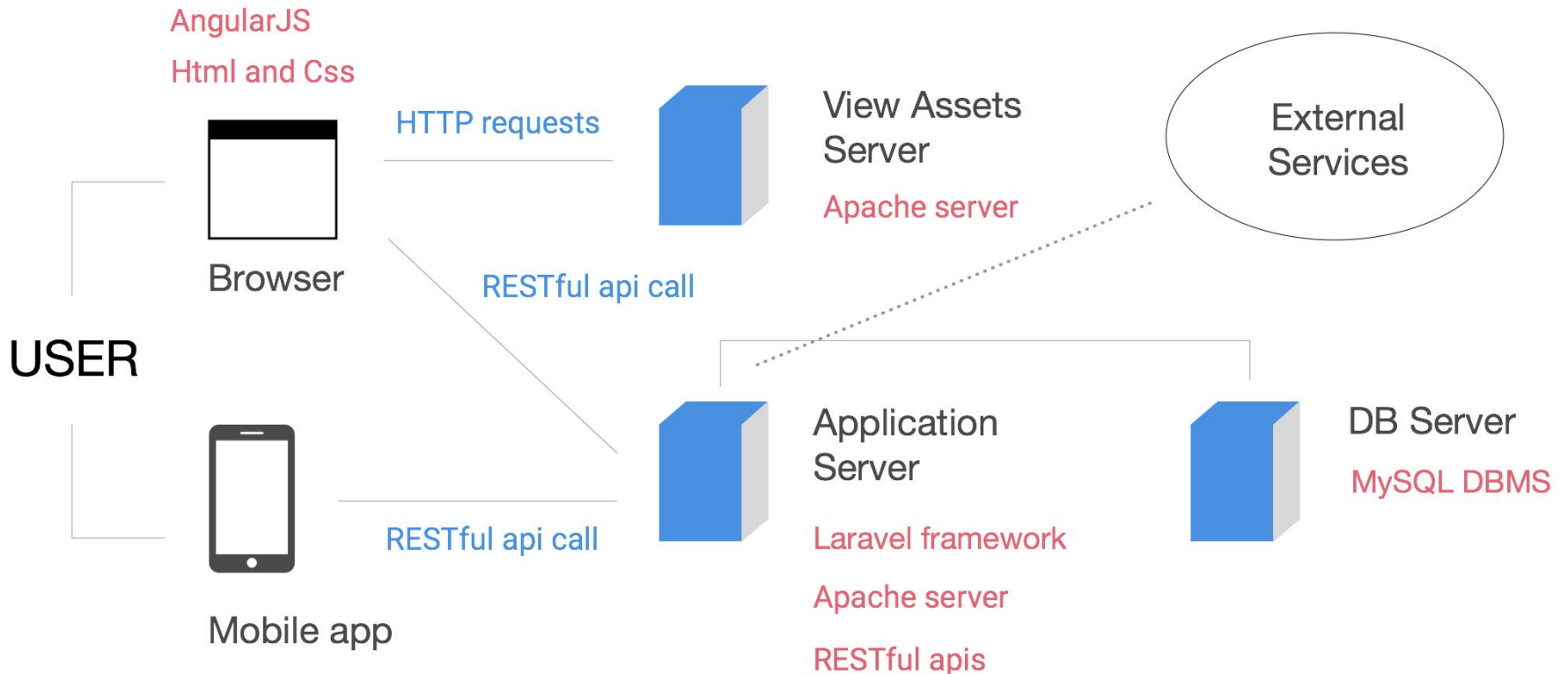
# Main requirements - 2

## **A planner must be able to:**

- Plan a meeting
- Manage and edit a meeting to be planned
- Have access to same functionalities of users



# Plunner's architecture



# Project & Scrum roles

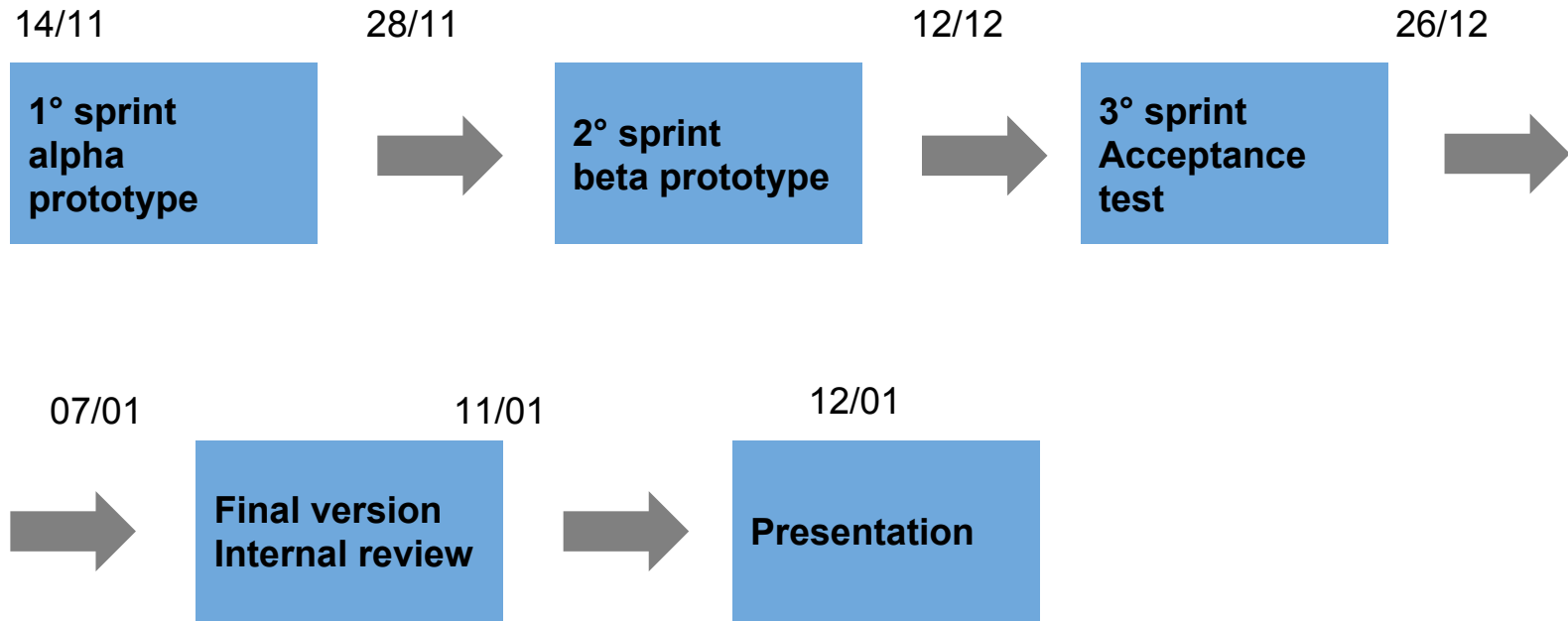
## Front end team:

- Giorgio Pea as Model/Controller developer **SCRUM MASTER**
- Jean Barre as View developer

## Back end team: **PRODUCT OWNER**

- Claudio Cardinale as **Laravel integration manager, architecture manager, external services manager**
- Denis Kuryshov as **Controller/Model developer**
- Mihovil Vinkovic as **Controller/Model developer**
- Emil Siladi as **Controller/Model developer**

# Project Plan



# Current status



# First sprint

→ Requirements completion for this sprint:

## For Users:

- Login
- Import of schedules via calDav
- Join meetings



## For Planners:

- Plan and manage meetings



# First sprint

## For Registered Company:

- Login and registration
- Employees registration and management
- Groups creation and management



# Second Sprint - 1

→ Features covered by this sprint:

## For Users:

- Email notifications
- Indication of free time slots
- Upload of schedules using files
- Integration of planned meetings in the schedules
- Two factor authentication and remember me option

# Second Sprint - 2

## For Planner:

- Logic to find the right meeting time (optimizing the number of participants)
- Constraints concerning meeting planning (required and invited users, time zones constraints )



# Second Sprint - 3

## For Companies:

- Two factor authentication and remember me option
- Employees import using LDAP

+ Features not completed in the previous sprint

# Team roles and membership

- Joan Josep Crespi Villalonga is no longer part of the team
- The role of tester is for now dismissed (testing will be distributed among other members)
- Claudio Cardinale → external services manager
- Emil Silađi, Mihovil Vinković, Den Kuryshov → model/controller developer
- Giorgio Pea → frontend model/controller developer
- Jean Barre → frontend view developer

# Design changes

We have switched the strategy for authentication of users/companies:

## Session Based Authentication



## Token Based Authentication using JWT

See: <http://jwt.io/introduction/>



# The Process



# Scrum

## Qualities and motivations

- The project can respond easily to changes
- Problems are identified early
- Continuous communication with the client
- Measurable productivity (via burndown, velocity, etc.)
- Better organization

# Scrum - 1

## Sprints

- *First sprint* (14/11 → 28/11)  
Already discussed in the Project Status Section
- *Second sprint* (28/11 → 12/11 )  
Already discussed in the Project Status Section
- *Third sprint* (12/11 → 26/11)  
Dedicated to acceptance test

# Scrum - 2

## Meetings

- *Daily scrum meeting (everyday at 18:30 for 15mins):* to decide and to assign tasks for the next day
  - We have varying number of participants
- First sprint review/retrospective (28/11)  
We noticed that some requirements/features were not completed and underlined that some team members need to increase the quantity of work

# Scrum - 3

## New requirements

→ **User story 1 :**

*As a company I want to register and login to the service*

### **New requirement (1.7):**

The service must be able to verify a company email

### **New requirement (1.8):**

The service must allow registered companies to see and edit their personal information (including password)



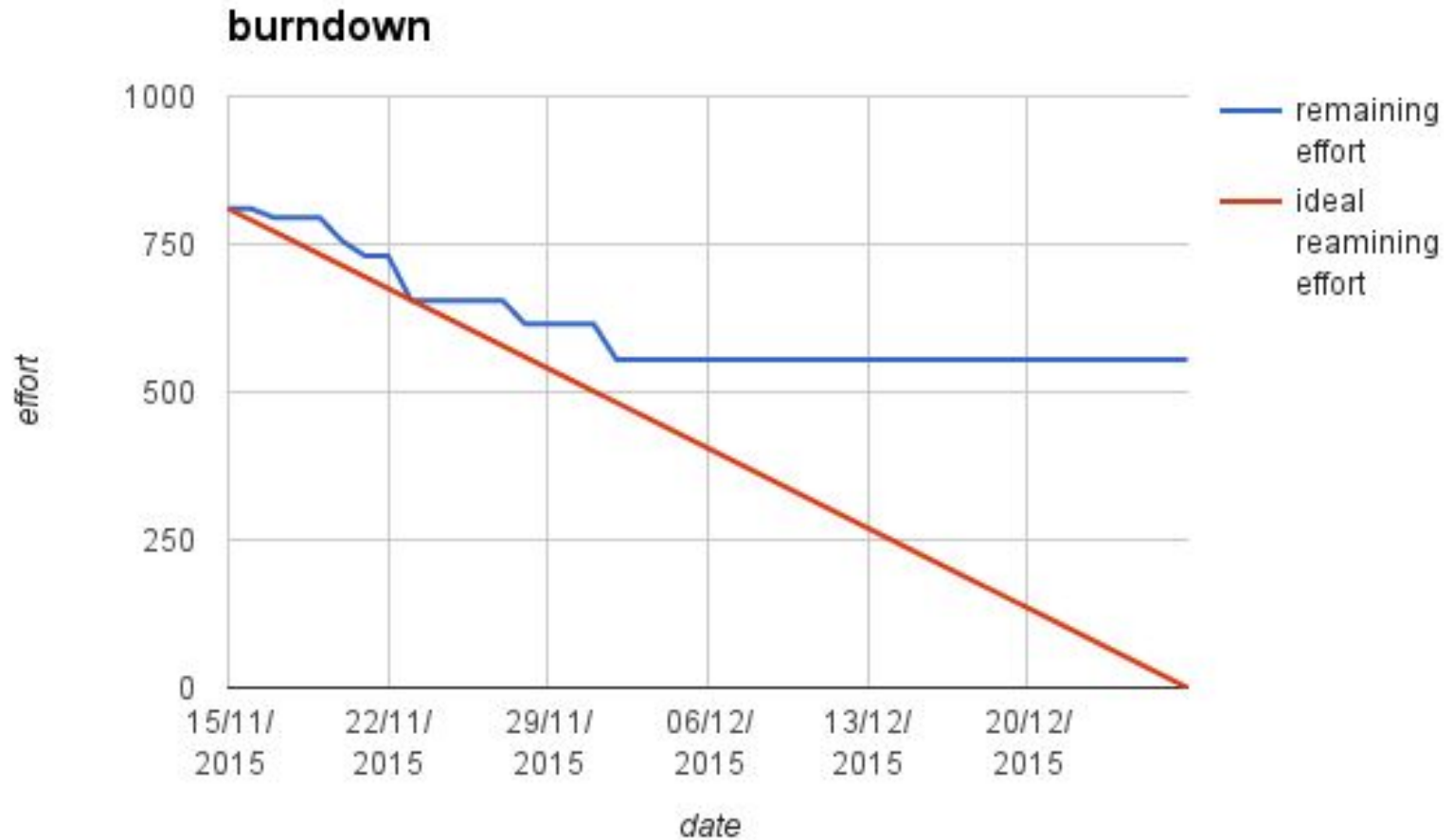
# Scrum - 4

→ **User story 3** : *As a user I want to login in the context of a registered company*

## **New requirement (3.8):**

The service must allow registered companies to see and edit their personal information (including password)

# Burndown diagram



# Working hours

## Working Hours

Member	W4	W5	W6
M. Vinković	6	18	1
D. <u>Kuryshov</u>	7.5	7	16.5
G. Pea	12	31	34.5
J. Barre	9.5	14	33.5
C. Cardinale	37	67	48.5
E. Silađi	6	18	33
Total	78	155	167

# Communication


- Daily meetings: **slack**
- Private messages to resolve small doubts: **facebook/slack**
- Sprint meetings: **skype**
- Extraordinary meetings to increase the work productivity: **skype**

# Testing and Integration



# Continuous Integration - 1

## Build Jobs


✓	# 218.1		</> PHP: 5.5
✓	# 218.2		</> PHP: 5.6



## Allowed Failures

✓	# 218.3		</> PHP: 7
✓	# 218.4		</> PHP: hhvm

## Badges

Scrutinizer	9.95	
coverage	72 %	
build	passed	

# Continuous Integration - 2

- We used Continuous integration system to test builds with different environments
  - high classes point (low CRAP, low coupling, low cyclomatic complexity and so on)
  - high test coverage
  - builds work with the newest php versions
  - our application doesn't depend on environment -> reliability
  - find problems before merge



# Simple stress test

```
ab -c 10 -t 3 http://api.plunner.com/stress
```

Percentage of the requests served within a certain time

(ms)

50% 740

66% 787

75% 811

80% 842

90% 875

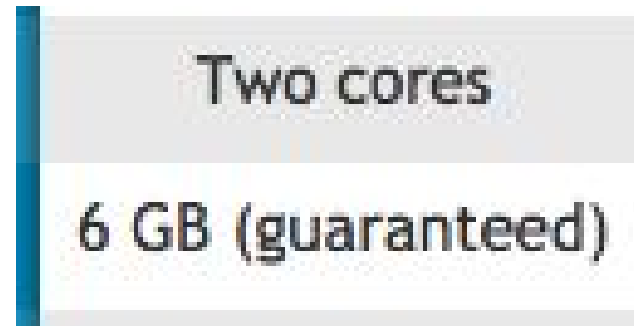
95% 899

98% 950

99% 950

100% 950 (longest request)

Intel Core i7 9xx  
(Nehalem Class Core i7)





# Professional approach addressed to real environment



# Packaged

- Use of git versions for demo to avoid problems due to new functionalities (in demo server we checkout to this version)
- Application packaged in <http://packagist.org/> to distribute our application to everyone

Latest release

v0.1.0

c606512

composer create-project dsd-meetme/backend

# Domain

We bought a domain for our application and we installed it on a real web server

- <http://plunner.com> -> presentation site
- <http://admin.plunner.com> -> frontend site
- <http://api.plunner.com> -> backend site

# Other

- Auto deploying to web server to test immediately
- Work with branches, new functionalities must be tested (100%) before adding them to the master
- Push notification on slack
- CI errors notification on slack to fix immediately the branch

# Demo

→ [admin.plunner.com](http://admin.plunner.com)

## We will show:

- Registration of a company
- Company and employee login
- Credential recovery for companies and users
- Group creation and management
- Users creation and management

# Questions?

