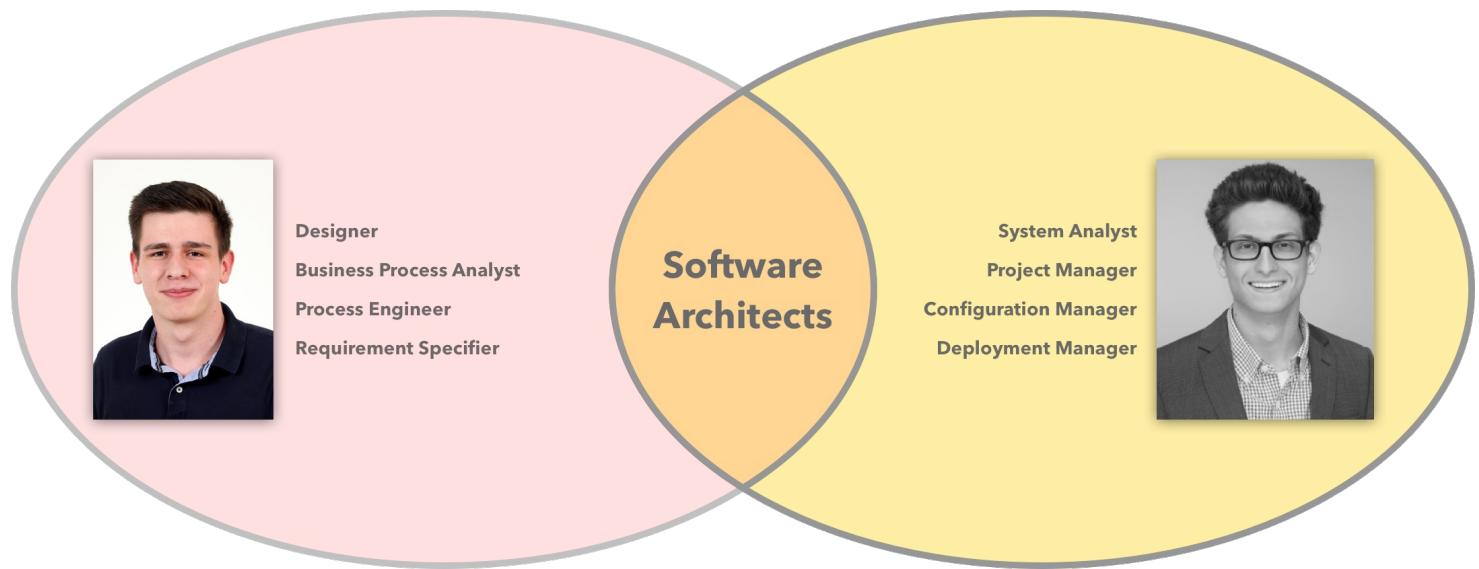




Team



2 thoughts on “Team”



Enrico Kaack says:

18. October 2016 at 13:33

Hello,

it seems to be a good use of the RUP roles. You have chosen the possibility to be both the project manager, I think that should work out, because you can spread the project manager work, although usually, not everyone is a project manager.

Best regards,
Enrico

[Reply](#)



Jan Hendrik Winter says:

18. October 2016 at 18:48

Hey,

I like your task allocation and think it will ensure an efficient workflow!
Unless Deployment Manger also means Test Manager, I would think about this as another role.

Regards,
Hendrik

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

POST COMMENT



The Vision

There are many activities we repeat in our every day life – activities that can be automated or at least simplified.

As young software developers it's our vision to simplify all these daily activities that cost everyone's valuable lifetime!

By understanding your daily habits we offer smart simplifications for example automate your alarm clock according to your calendar. We plan to develop an iOS Application that will assist you in your daily organization and planning. Therefore it should look nice and need to be as intuitive as possible.

Stay Up-to-date in our project blog.

The App

The App will be an alarm-clock App within we will work on a functionality that will set the alarm clock automatically to the right time according to the appointments in your calendar on the next morning. To give an example, when you need to drive to the University at 8 o'clock the app will check how long you need to get to the University and how long you need to get ready in the morning. According to this information the alarm clock will wake you up at the right time.

Our first priority is that our app works. Simple as that. According to this, we will focus on a high stability of the application so that the background service of the application doesn't crash overnight, for example. We want to build a smart alarm-clock so that there is no need to keep the alarm-clock in mind. That would be the Best case.

But there are some technical issues according to our operating system that we need to work on. We don't know yet how much of our ideas we can implement. To name an example, the OS for Apple's mobile devices (iOS) is very strict when it comes to Background-Services.

We plan to split the App in several subsystems. One subsystem will be the connection between the App and the Calendar. This includes that the App gives you the option to define a Calendar that will be used for the automatically wake up.

Furthermore, another subsystem is that we want to implement a Google API Connection to get current traffic status on the route that the User need to take to work in the morning. We want this results to influence the clock-time, too.

Additionally we want to use Spotify-Playlists as alarm-sounds that you can wake up with your own chosen music. That can be considered as a subsystem as well.

At least we want to publish our App in Apples Marketplace for Mobile Apps, simple called AppStore. According to the fact that we never published a App this is a subsystem too.

As we already mentioned before, we have a Dependency in our Project. This Dependency is our Operating System for the App. We never build a so huge App based on Swift. According to this fact we don't know yet which of our ideas we can implement in the way we think. We are limited by the API's that Apple provides with their iOS and we will use some strictly monitored function of the iPhone(iOS) like the Background Services.

8 thoughts on “The Vision”



Steven Kovacs says:

11. October 2016 at 09:35

Hello!

This seems to be a very interesting idea. Especially for people that need a reminder for their meetings and appointments.

What else do you want to provide?

Is it going to be an app with interface and userinteraction or a background service?

Greetings! 😊

[Reply](#)



René Penkert says:

16. October 2016 at 18:26

Thank You for your Comment Steven.

We have some ideas how to implement our app. But we need to proof that the things we want to implement and how we would like to implement them are technically implementable. We wanted to set the default alarm-clock of the iPhone. But this for example is not possible because Apple does not provide an API to create an alarm-clock within the default alarm-clock app.

Finally, there will be both. There will be a user interaction where you can configure the alarms and do some additional stuff. There will be a background service that proofs for example if there are new appointments in the calendar so that the alarm-clock need to be earlier or later.

Cheers,

René

[Reply](#)



Enrico Kaack says:

16. October 2016 at 13:14

Hi,

this is a great idea. Automate things, which can be done automatically is very useful.

I just have one thought, which may be interesting for you:

How automatic should it be? How do you get the users trust, that your application does not fail, when it sets the alarm clock? E.g. if I had a very important meeting in the morning, would I trust an app to set up my alarm clock? Or would I set a second alarm manually?

I would personally set up a alarm manually, because I simply won't trust the application in the beginning.

To sum up, I think you should have a concept of giving the user insights, what you have automated. For example in the evening, a simple notification "alarm set up to 7 o'clock to reach your meeting at 8 o'clock in time" would reassure me.

Greetings,
Enrico

[Reply](#)



René Penkert says:

16. October 2016 at 18:34

Thank You for your Comment Enrico.

You are totally right. That's something we need to think a lot about.

Our first priority is that our app works. Simple as that. According to this, we will focus on a high stability of the application that the background service of the application can't crash overnight, for example. We want to build a smart alarm-clock so that the "user" don't even need to think about the alarm-clock. That would be the best case.

But we have some technical issues to solve according to our operating system. We don't know yet how much of our ideas we can implement.

Apple's iOS is for example very strict when it comes to Background-Services.

Thanks for your nice idea. I think that is really something we need to implement 😊

Cheers,
René

[Reply](#)



Sebastian Luther says:

17. October 2016 at 07:53

Hi,

this is a nice idea! It's very useful to automate such things like a alarm-clock because you don't have to worry about it anymore.

I would definitely use the app, but I have an Android device.

Are you also going to do a Android App?

Greetings
Sebastian

[Reply](#)



René Penkert says:

17. October 2016 at 08:16

Thank You for your Comment Sebastian.

We don't plan to build an Android app. We will focus on our iOS-App.

Cheers,
René

[Reply](#)



Karl Spickermann says:



17. October 2016 at 09:38

Hi,

I like your idea to automate tasks. I often find myself in situations where i would like to have an automation to save me from my own forgetfulness. So the concept really appeals to me. I just ask myself how your app gets the needed information more complex things later on. Will I have to give the App access to some of my other Apps , will there be the possibility to integrate accounts from other services or do I have to provide all Information by hand ?

Greetings

Karl

[Reply](#)



Benedikt Bosshammer says:

17. October 2016 at 12:53

Hi Karl,

thank you for your feedback!

The first feature will use build-in iOS APIs to fetch the required OS Data and public APIs for external data supply. We try to focus on those two sources so there is no need for critical personal data to be provided.

The implementation is focused on a minimum of 'by-hand' information supply.

Best wishes,

Bene

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

POST COMMENT

Proudly powered by WordPress | Theme: Sydney by aThemes.



Software Requirement Specifications

Today
- I'm very proud to present
our new SRS to you.
It's the best and gorgeous
SRS we've ever created!



We specified our Software Requirements Specifications.

For an optimal version control we pushed it to github, see the link below:

[SRS](#)

[← Technology Choice](#)

[First two Use Cases →](#)

2 thoughts on “Software Requirement Specifications”



[Enrico Kaack](#) says:

27. October 2016 at 15:48

Hi,

thank you for providing your SRS. It seems like GitHub does not provide a preview for docx documents. I would recommend you to go to a format like .md (markdown), which makes a preview possible and you have a better version control, since you can trace the changes due to the preview capacity (GitHub shows .docx as a binary file, so changes will replace the complete file and not just the lines you modified). Furthermore, users can view it without owning Office, which is a big advantage.

Regarding your content:

The criteria for the SRS was to replace all blue template text with your own text. I have seen two places, in which the template text is still in. It would be good if you could replace it (don't forget that you can use n/a and tbd if you need it).

Furthermore, I can't find an overall use case description, so please provide one.

And as a last hint, I think the SRS should be a stand alone description, so maybe do not refer to your website. Instead copy your project vision in it.

Best regards,
Enrico

[Reply](#)



Karl Spickermann says:

27. October 2016 at 17:20

Hey guys,

you used the right template and have all the needed information except for the UCD, but you still need to replace the template text and update your Table of Contents. Furthermore you forgot to replace all the placeholder, as an example “”, with the relevant content. Moreover you should take a look at your formatting it seems to be a bit off at some places.

Greetings,

Karl Spickermann

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

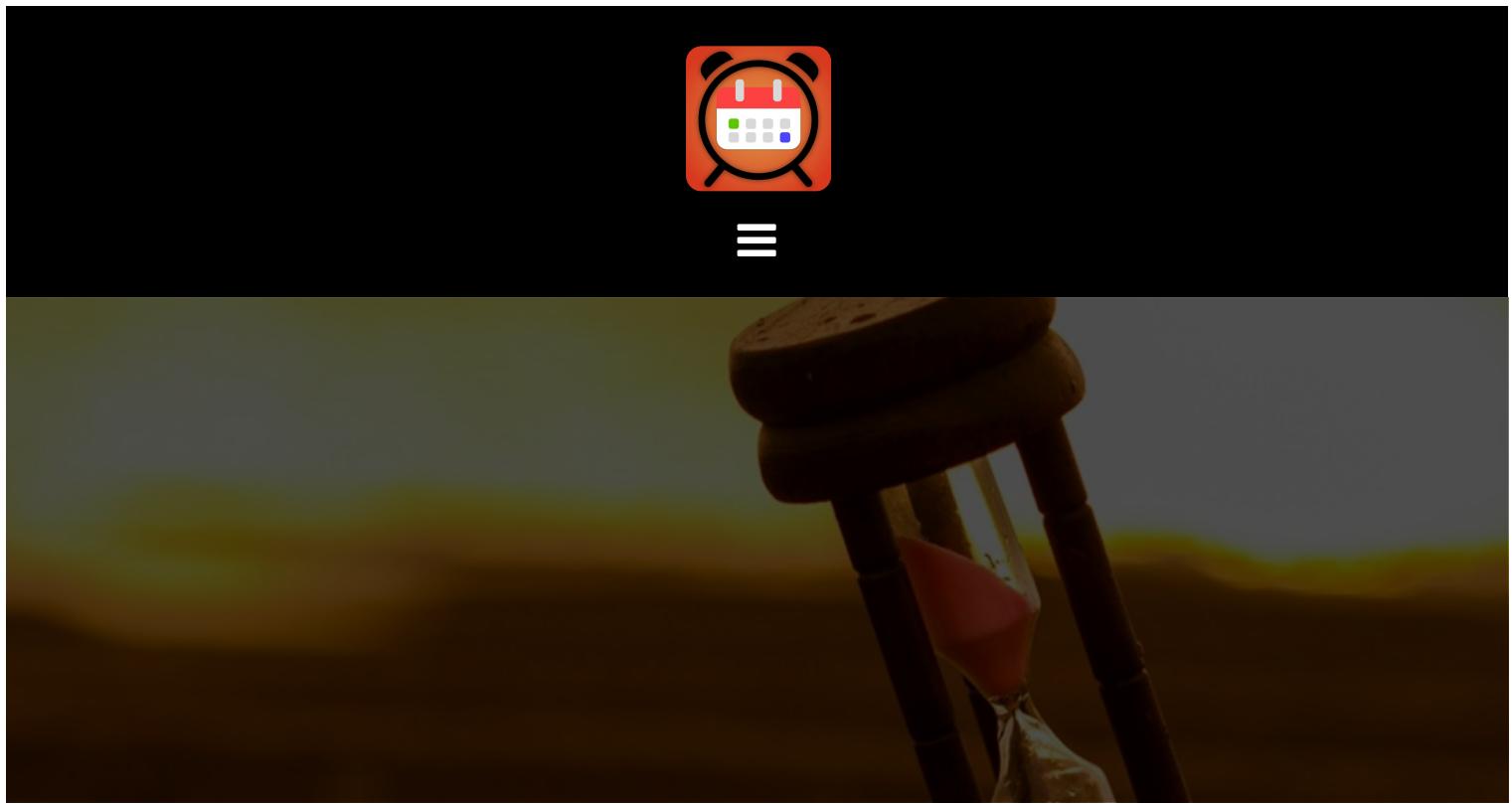
Name *

Email *

Website

Comment

POST COMMENT



Technology Choice

To implement our ideas we use the following technologies:

With Swift 3 and the recently published iOS-APIs Apple provides an excellent environment for iOS-App Development. Therefor our main programming language is Swift 3. As IDE we use Xcode 8 to organize our App according to the Model-View-Controller pattern of Swift. Within Xcode we integrate our GitHub to guarantee a version control and to optimize our team collaboration.



[Software Requirement Specifications →](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

POST COMMENT

Proudly powered by WordPress | Theme: Sydney by aThemes.



First two Use Cases

We created our first two Use Case Documents:

[UC Select Calendar](#)

[UC Create Alarms \(CRUD\)](#)

[← Software Requirement Specifications](#)

[Sprint →](#)

4 thoughts on “First two Use Cases”



Enrico Kaack says:

3. November 2016 at 20:50

Hi,

I had a look at your UC Documents and I really like your Mockups.

I would like to give you the tip to use the “Activity” element for your Flow Diagram, because they have a red border and a yellow background like the “decision diamonds”. Furthermore, don’t forget to make a text label at the “decision diamonds” arrows (yes or no).

Best regards,
Enrico

[Reply](#)



Jan Hendrik Winter says:

3. November 2016 at 21:04

Hi guys,

I like your work and I think you did the most right. However, I noticed a few things.

First, I did not find your use cases in the SRS and according to the task we should link them there.

Furthermore, I would not devide between screen and operating system in your Flow of Events diagram. Because both, the operating system and the screen are not a direct part of your application. So I would recommend a column application which contains the actions your application does during a process.

Best regards

Rainbow Ninjas

[Reply](#)



Daria Seledtsova says:

3. November 2016 at 21:14

Hey guys,

just checked out your UC-docs, I totally agree with Enrico, mockups look very good and give a good overview of what you expect your application to look like. One thing I've noticed though is the document structure. According to the example document

(<https://docs.google.com/document/d/1Ju5lxLI3fsuklXesmP8R882GlcVlmJf39ROfHuACGjo/edit>) your activity diagram would go into the chapter "Basic Flow". That way the Specification is clearer and you can still add an alternative flow in case you come up with one.

I wish you all the best with your project!

Cheers,

Danny

[Reply](#)



Daria Seledtsova says:

4. November 2016 at 11:51

Oh and one more thing: the activities in your diagrams should be in rounded forms instead of rectangles. Did you use draw.io? The right one are described as "activity" there and come with red arrows by default (the colors of your arrows are inconsistent).

Hope it does not cost you to much work to fix those small things 😊

Good luck with the project!

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

POST COMMENT

Proudly powered by [WordPress](#) | Theme: [Sydney](#) by aThemes.



Sprint

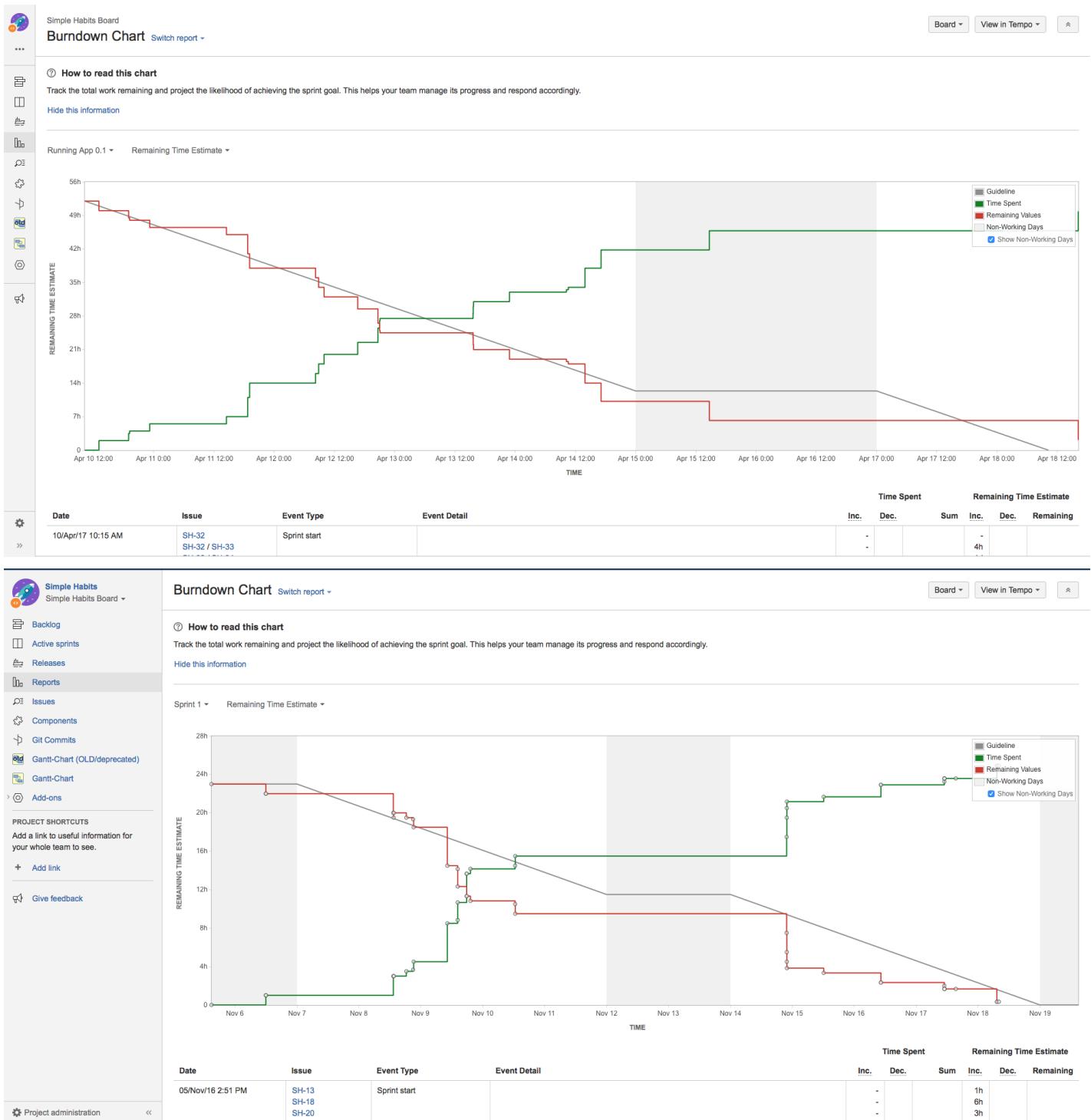
Hey,

check out our Sprint board:

[Sprint Board](#)

As you might see, we don't use any JIRA plugins due to the problem that Xcode as our main IDE doesn't provide any! As an alternative we use github integrated in JIRA. By doing this we are able to link our commits with the corresponding tickets in JIRA just by using the tickets identifier as commit message. If you have any questions concerning this method feel free to ask! 😊

In the pictures below you can see 2 of our healthy sprints:



Best wishes,

Bene

← First two Use Cases

.feature Files →

2 thoughts on “Sprint”



[Enrico Kaack](#) says:

11. November 2016 at 07:56

Hi,

good to see your fist sprint. For the first try, I think your Burndown Chart and your sprint looks very well. I like your structure with sub-taks and your active commenting.

Furthermore, it seems like you estimated the time for your sprint pretty well. Not perfectly, but very good for your first try.

I would suggest, it would be good to rename your sprint to match the correct RUP terminology.

Best Regards,

Enrico Kaack

[Reply](#)



Daniel Wagner says:

11. November 2016 at 07:59

Hi guys,

you already seem to have a very healthy burndown chart, so the grading criteria about sprint and scrumboard is obviously fulfilled. The link is in here as well so that is good.

As far as I was able to see your issues all use speaking names, but I am not sure how the second part of this criteria should be understood. I think you are missing the discipline on some of the issues that can clearly be sorted into one of those RUP categories. The “animation implementation research” is probably part of the elaboration part for example.

You are using sub tasks very nicely, so I think you could just switch the main task to be a story to fulfill the criteria here a little bit better.

I also understand that you cannot fulfill the IDE integration criteria like we can't since you are using XCode and there is no plugin for that.

Overall very good work!

Best regards,

Daniel

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

POST COMMENT

Proudly powered by WordPress | Theme: Sydney by aThemes.



.feature Files

Check out our .feature files!

SelectCalendars

And here you can see that the IDE supports .feature syntax highlighting:

The screenshot shows two instances of the IntelliJ IDEA code editor. Both instances display the same Cucumber feature file for 'SelectCalendars'. The file contains the following steps:

```
Scenario: #Calendar selected
  Given Menu with button "Calendarname 1"
  And the button "Calendarname 1" has the calendar's color as its backgroundcolor
  When I press the button "Calendarname 1"
  Then The background color of button "Calendarname 1" should change to gray

Scenario:
  Scenario Outline:
  Scenario Template:
  ^↓ and ^↑ will move caret down and up in the editor >> π
```

Below the first instance, a tooltip is visible with the text: "#CALENDAR_SELECTED".

The second instance of the editor shows a partially typed scenario outline:

```
#CALENDAR_SELECTED
Given Menu with button "Calendarname 1"
And the button "Calendarname 1" has the calendar's color as its backgroundcolor
When I press the button "Calendarname 1"
Then The background color of button "Calendarname 1" should change to gray

Scenario:
G
Given
Background:
Press ^. to choose the selected (or first) suggestion and insert a dot afterwards >>
```

We tried out the .feature files to come in contact with Cucumber and this Technology. Xcode doesn't provide support for feature files according to this we created this files you see on this page in IntelliJ. Nevertheless we want UI-Testing.

For the UI-Testing in our Project we use the integrated Xcode UI-Testing Environment. That's a very nice tool that Xcode provides. You can for example simply record your Test while you are clicking through the App in a Simulator. The UI-Testing is a "new" feature that Apple presented on their WWDC last year (2015). If you are interested in this technology check out : [Xcode UI Testing \[recorded session from WWDC 2015\]](#)

Here are our Xcode UI-Testing Files:

- [GetOverviewAndSaveTestcase](#)
- [SelectCalendarTestcase](#)
- [SmartAlarmCreationTestcase](#)
- [CrudTestcase](#)

Best wishes!

← Sprint

Class Diagram →

4 thoughts on ".feature Files"



Louisa says:

15. November 2016 at 19:22

Hey guys,

your feature files look pretty good. Also your IDE support is nice. I'm looking forward to your Demo next week. Maybe you will make an Android Version next semester, so I can use it too 😊

Greetings,

Lousia (Team inaeternum)

[Reply](#)



Torben Krieger says:

17. November 2016 at 16:57

Hello Simple Habits Team,

pretty cool to see that your very useful app is in progress!

You wrote the two narratives that were claimed in the grading criteria. Also every narrative includes more than one scenario and the scenarios agree with the UC-Diagram. So the basic requirements are fulfilled already... All your scenarios are relatively short (what not means that they are bad). May you can insert how to come to the described view from the start screen of your app.

The autocompletion and code highlighting of your IDE is working perfect too. So according to the grading criteria there is nothing to criticize.

One last point I want to mention: Why you not added the .feature files at self to the GIT Repository? Or are they in another folder?

Kind regards,

Torben Krieger

[Reply](#)



Karl Spickermann says:

17. November 2016 at 21:02

Hey guys,

Your .feature files fulfill all of the requirements of the Grading Criteria.

You showed that you have short and precise stories with the right syntax,
your IDE is correctly configured and you linked the files in the correct places.

Good Job.

Greetings,



Christoph Emig says:

22. November 2016 at 17:39

Hey guys,

we like your new post about the .feature-files. You specified the two use cases short and well. You have configured your IDE correctly to make the gerking syntax highlighting working and also linked the files to your UC document. As far as the grading criteria says, all tasks are completed. Well done!

Have a nice day,

Team GottMusIg

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

POST COMMENT



Class Diagramm

Check out our Class Diagramm:

[Class Diagram](#)

← .feature Files

Software Architecture Document (SAD) →

3 thoughts on “Class Diagramm”



Torben Krieger says:

25. November 2016 at 07:46

Hey Team,

your class diagram seems to be very good . Unfortunately you are unable to use any IDE tool because XCode doesn't provide anything. So it was extra effort for you to create the diagram. A database schema is also not necessary for you because you not use any database at all. But you have more than five classes in the UML diagram. So summing up with the exceptions that are applied to you you fulfill the grading criteria. Great Work!

Kind regards

Torben

[Reply](#)



Marco Crocoll says:

25. November 2016 at 10:22



Hey guys!

Although you could not automatically generate the class diagram, it looks very good!
It has a clear structure and you can easily recognize your classes.

Greetings,

Marco from "in aeternum"

[Reply](#)



Leon Gottschick says:

1. December 2016 at 21:53

Hey dudes,

despite not being able to generate the diagram automatically it looks clear and structured,
good job on that!

You seem to have fulfilled the grading criteria.

Greets,

GottMuslg

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

POST COMMENT

Proudly powered by WordPress | Theme: Sydney by aThemes.



Software Architecture Document (SAD)

Hey,

check out our SAD!

[SAD](#)

Best wishes,

Bene

[← Class Diagramm](#)

[Gantt-Chart →](#)

3 thoughts on “Software Architecture Document (SAD)”



Karl Spickermann says:

1. December 2016 at 19:02

Hey guys,

I like what I see until know. You appointed all topics precise and short so that readers can comprehend the information in a short time.

You are still missing a few points, but you already allocated Rene for this part so I think it is fine.

Furthermore I like that you linked the offical documentation for your MVC in your document, that will prevent the reader from consulting a false documentation from a quick google search.

Greetings



Torben Krieger says:

2. December 2016 at 07:53

Hey Team,

you uploaded a correct version of the software architecture document template. All needed points are in the document and it is customized for your project. The deployment criteria is not relevant for you because you not using any server backend. A UML class diagram for your code is missing up to now. But this is only claimed if you already have enough coding. The deployment view is also not relevant for you... Furthermore you not using any database so it is unnecessary to draw it.

Summing up you have to add a UML class diagram and you fulfill the grading criteria completely!!

Kind regards,

Torben



Christoph Emig says:

8. December 2016 at 14:05

Hey Guys!

Your SAD looks good. However we miss your UML. Maybe there is not enough code at this moment? Would be important that you hand it in later when you have enough code or when there's already one put it in your SAD now :).

Have a nice day!

Team Gottmuig

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

[POST COMMENT](#)

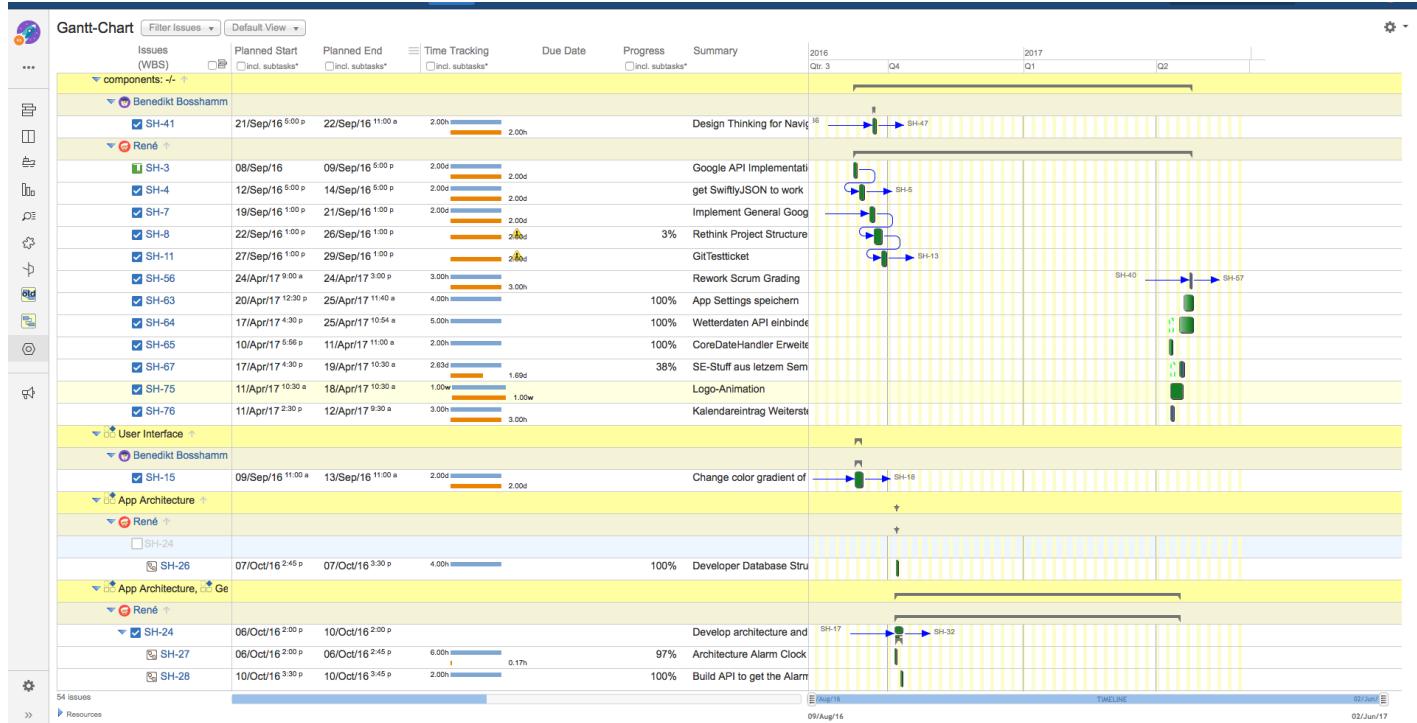
Proudly powered by [WordPress](#) | Theme: [Sydney](#) by aThemes.



Gantt-Chart

Hi guys,

We created a Gantt-Chart with the installed Plugin. In our View the Plugin doesn't work properly because the Gantt-Chart looks very ugly as you can see



The Plugin doesn't integrate all Issues into the Gantt-Chart. The time is tracked on all tickets but the Plugin only integrates a handful of tickets into the Gantt-Chart. We think, that the Plugin may be broken.

We would be pleased to show you our RUP and Milestones. But with this Gantt-Chart it is not possible. We can't show our Project-lifecycle with a

Gantt-Chart that only integrates less than a Quarter of the Issues.

Best Wishes

← Software Architecture Document (SAD)

Midterm →

2 thoughts on “Gantt-Chart”



Tom says:

9. December 2016 at 08:17

Hey guys,

your Gantt-chart looks clean. I'm glad to see you switched from Jira to YouTrack during the couple last days and used this tool to create your chart. My team made some well experiences with YouTrack. Hope you like it as well 😊

Even though, I have to complain about a little point. It would have been nice if you managed to apply the RUP phases to your Gantt chart.

If you do so, keep a closer look about the milestones inception/elaboration and their start and end dates according to the needs of your prof.

Hope to hear from you soon 😊

Cheers

Tom

[Reply](#)



Leon Gottschick says:

9. December 2016 at 09:43

Hey guys,

nice to see you set up YouTrack for your project.

Its pretty easy to import Jira data with it, and your chart looks nice.

Although the dates are a little bit off, your start date is set to 8/12/16, although we set up Jira somewhere in november i think.

Another downside of the YouTrack Gantt chart is the limited configuration possibilities of the chart. The RUP phases are missing as well as milestones.

Greets,

GottMuslg

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

POST COMMENT

Proudly powered by [WordPress](#) | Theme: [Sydney](#) by aThemes.



Midterm

Links to the written documents for the Midterm Testat:

- [Vision](#)(Week 1,2)
- [Team](#)(Week 1,2)
- [Technology Choice](#)(Week 1,2)
- [Software Requirement Specifications](#) (Week 3)
 - [SRS Document on Github](#)
- [First two Usecases](#) (Week 4)
 - [Use case Select Calendar Document on Github](#)
 - [Usecase Select Create Alarm\(CRUD\) Document on Github](#)
- [Sprint](#) (Week 5)
 - [Sprintbord on Jira](#)
- [Feature-Files](#) (Week 6)
- [Class Diagramm](#) (Week 7)
 - [Class Diagram](#)
- [Software Architecture Document](#) (Week 8)
 - [SAD Document on Github](#)
- [Gantt-Chart](#) (Week 9)
- [Midterm-Presentation](#)

You can find our Code on [Github](#).

All the important Files for SE on [Github](#)

← [Gantt-Chart](#)

[Risk Analysis + 5 new UseCases!](#) →

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

POST COMMENT

Proudly powered by WordPress | Theme: Sydney by aThemes.



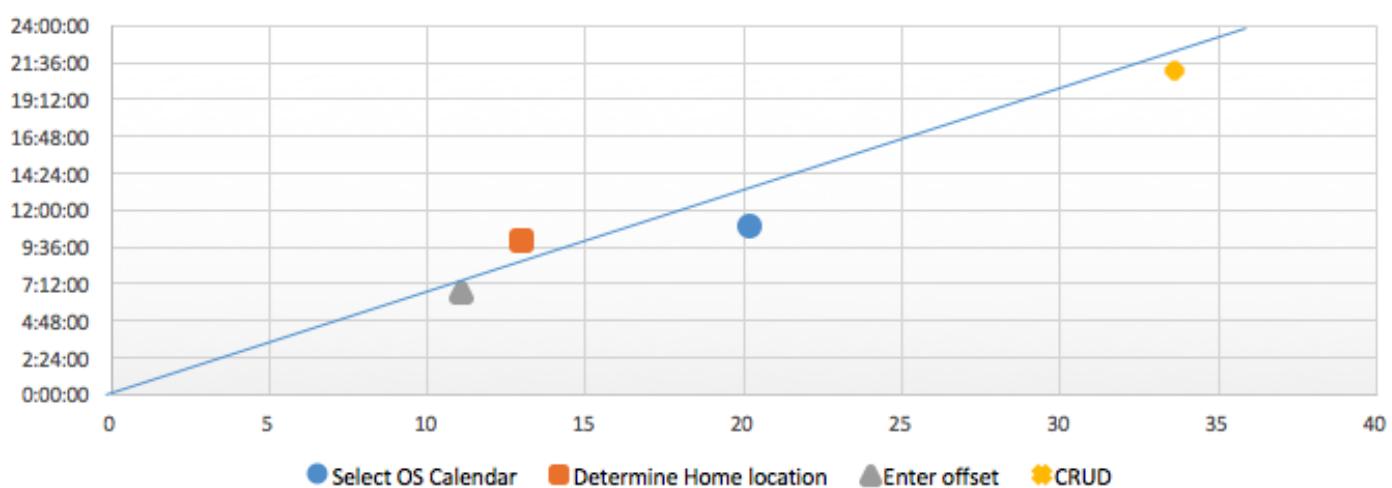
Function Points

Hey,

we calculated our Function Points for future time estimation.

Check out our [Excel](#).

This graph shows our calculation:



The Outliers are invisible in this Graph. There were two big outliers in presented Graph. Both of the Outliers trace back to the fact, that there were huge involvement of the Apple API's and in the first place we didn't know how to use them correctly. For example the simple Use case "Save Alarm" was far beyond the Graph because in this Use case there was much more than we expected. We needed to handle the Persistenz-Service for iOS Devices called "Core-Data" among the fact that we needed to design our complete Architecture and the API's for using it just to save Alarms and get the App working. We had outliers, yes. But we could explain them to ourselves.

Furthermore, with this great Function Points Graph we are now able to estimate our next Use Cases more correctly. The Use Case "Determine Home

Location" was almost exactly estimated with the amount of time we used to implement it. It just works.

Best wishes,

Benedikt

← Risk Analysis + 5 new UseCases!

Continous Integration and Testing →

3 thoughts on “Function Points”



Torben Krieger says:

24. April 2017 at 09:14

Hey Team,

great work. Your homework fulfills the grading criteria. Now you can you use your calculation for the estimation of your further UCs.

Keep going!

Best regards,

Torben

[Reply](#)



Karl Spickermann says:

24. April 2017 at 13:28

Hey guys,

good job on your FP estimation. You fulfilled all the grading criteria for the assignment. You made a function point estimation for every use case and collected them all in a single table. Furthermore you generated a neat graph from the resulting data .

One suggestion so: Maybe you could label the points in your graph so that it becomes clearer which point represent which UC.

Kind regards,

Karl

[Reply](#)



Christoph Emig says:

11. May 2017 at 07:13

Hey guys,

your graph looks very well. You did a great job with estimating your FP. Everything should be fulfilled for the grading criteria. All UCs were estimated and collected. Good job!

As Carl already mentioned: It would be nice if you could label your FB in the diagram. But these are peanuts ;).

Greets,

GottMusIG

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

POST COMMENT

Proudly powered by WordPress | Theme: Sydney by aThemes.



Risk Analysis + 5 new UseCases!

Hi,

check out our [Risk Analysis](#).

We added five further use cases to the scope of the second term.

[SRS](#)

Have a nice day!

[← Midterm](#)

[Function Points →](#)

3 thoughts on “Risk Analysis + 5 new UseCases!”



Tom bendlrath says:

13. April 2017 at 07:29

Hey guys,

I took a look on your risk management table and it seems that you are prepared for possibly upcoming risks. In addition, I would add some kind of social conflicts within the team members to your risks. Furthermore, I think that hardware problems might also be a huge risk to your work.

Regarding your new use cases, I guess you did a good planning here. Even though, it is quite hard for me to get an overview about the use cases you have done so far. You might highlight both your previous and new use cases in different colors. All in all, I really would like to use your application by the end of this semester. I hope you will achieve your goals!

Kind regards,

Tom

[Reply](#)



The Meets Team says:

13. April 2017 at 08:55

Hi there, you meet the requirement of adding 5 new use cases to the scope of the 2nd semester. I like the use cases, they are pretty self-explanatory, also since you have only one role in your UCD.

As for your risk analysis you found out the most dangerous risks and described them well. I also liked how you calculated the risk factor by the probability of occurrence and impact and sorted your table by that criteria.

I do however think that should pick a responsible person for mitigating and dealing with the risks since otherwise it might get a little messy and nobody feels responsible for it.

Best regards, The Meets Team

[Reply](#)



Franziska says:

14. April 2017 at 16:42

Hi Simple Habits,

am I right to assume, that the blue bubbles in your Overall Use Case Diagram are the new use cases? Maybe a legend would help.. Furthermore, to what refers the CRUD? Is it for creating,... an alarm?

Your Risk Analysis is well done. There are no such risks as snow storm in June and I like the last one: "An apple a day..." 😊

Finally, the table of spend hours for the use cases of last term is missing, so just add it.

Kind Regards,

BestPlaces

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

[POST COMMENT](#)

Proudly powered by [WordPress](#) | Theme: [Sydney](#) by aThemes.



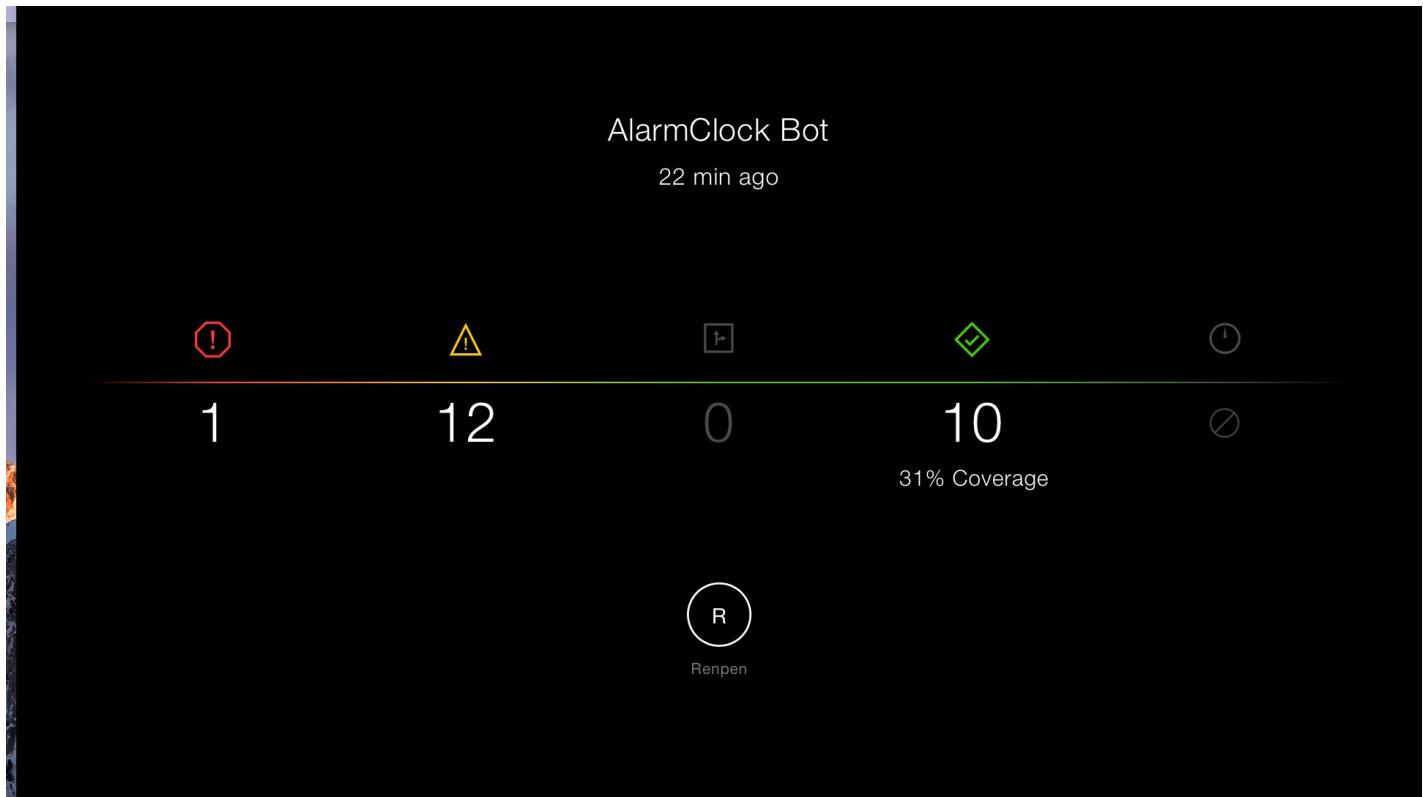
Continuous Integration and Testing

Hi guys

we started with Continuous Integration of our App. As always in the Apple World we can't use something like Jenkins for this task. Apple offers a solution called xCode Server that is a feature of the Mac OS Server App. This App can run on every Mac and turns it into a Server with additional features.

The xCode Server for instance provides the possibility to create something called xCode Bot. These bots can be created on every Development Mac and can then be deployed at the Mac OS Server. The bot has several properties that can be manipulated by the user. It needs a remote repository from where the bot will checkout the code. The bot can be configured so that it always runs when a new commit arrives at the remote git repository. The UI Tests and the xCodeTests will be triggered every time someone pushes something into the Repository. Furthermore, there are several options to define post or pre scripts.

The current status of the Bot/App can be checked on the Mac OS Server Dashboard for Bots:



In Addition, we implemented Testing for our iOS App. In xCode the Tests are called “xcTest” that obviously mean Xcode Test.

The Tests for iOS and all other Operating Systems from Apple like tvOS, watchOS, macOS, are written in Swift or objective C. According to the fact that our App is written in Swift we developed the Testcases in Swift, too. Find more about Testing with xCode at the Apple Homepage: [Apple Developer Documentation “About Testing with xCode”](#). Furthermore, the xcTests provide some APIs to make testing easier. For example there are two methods to separate the test code and the test data. “setUp()” is called before the test is called. With this method you can load data and or create new data before the actual test begins. “tearDown()” is called after the test is finished. It is meant for cleaning up the mess the test left behind 😊. The two methods results in a good separation of test data and the actual test code.

See here some example Tests:

- [AlarmClockTest](#)
- [WeatherAPITest](#)
- [TravelCalculationTest](#)

In the following Screenshot you can see on the left hand side the success of the Tests:

AlarmClock | iPhone 6s AlarmClock | Build AlarmClock: Succeeded | Today at 18:28

```

10 tests, 1 failing
  ▾ AlarmCoreData
    ▾ testPerformanceExample()
    ▾ testAlarmCreationAndSaving()
    ▾ testAlarmDeletion()
  ▾ TravelCalculation
    ▾ testPerformanceExample()
    ▾ testTravelCalculation()
    ▾ testPerformanceExample()
    ▾ testTravelCreation()
    ▾ testTravelDeletion()
  ▾ WeatherTesting
    ▾ testWeatherAPIs()
    ▾ testPerformanceExample()
  ▾ AlarmClockUITests 0 tests
    ▾ AlarmCoreData
    ▾ UITestCRUD
    ▾ UITestGetOverviewAndSave
    ▾ UITestSelectiOSCalendar
    ▾ UITestStartSmartCreation

```

```

import Foundation
import SwiftyJSON

class GoogleDistanceMatrixTravel : Travel {
    var offset : Int?
    var calculatedJsonObject : GoogleDistanceMatrixObject?
    var departure_time : Int?
    var arrival_time : Int?
    var destination : String?
    var transitmode : TransitMode?
    var trafficModel : TrafficModel?
    var mode : Mode?
    var representingCoreDataObject : TravelC?

    private func isValid() -> Bool { //determine if the minimum that the request need is set
        if(offset != nil && source != nil && destination != nil && mode != nil)
        {
            return true
        }
        return false
    }

    func calculateTimeInS() -> Int
    {
        return calculatedJsonObject!.durationValue + offset!
    }

    func calculationFinished()
    {
        print("FINISHED! YEEESS " + (self.calculatedJsonObject?.durationText)!)
    }

    func calculateTravelTime(closure: @escaping (_ : Int) -> Void)
    {
        if(isValid())
        {
            RestAPIManager.sharedDistanceInstance.request(url: generateUUID()) { (json: JSON) in
                self.calculatedJsonObject = GoogleDistanceMatrixObject(json: json)
                DispatchQueue.main.async(execute: { //the thing that need to do when the Request is finished
                    closure(self.calculatedJsonObject?.durationValue!)
                })
            }
        }
        else{
            closure(0)
            //throw some kind of exception TODO
        }
    }

    func generateURL() -> String {
        var url = properties["GoogleDistanceMatrixBaseUrl"] as! String
        print(url)
        url += "?origins=(source)&destinations=(destination)&key=\\" + properties["GoogleAPIKey"] + "\\"
        if(departure_time != nil && departure_time > 0){
            url += "&departure_time=" + departure_time
        }
        if(mode != nil)
        {
            url += "&mode=\\" + mode +
        }
        if(mode == .transit & (transitmode != nil))
        {
            url += "&transitmode=\\" + transitmode +
        }
        if(mode == .driving & (trafficModel != nil & departure_time != nil))
        {
            url += "&traffic_model=\\" + trafficModel +
        }
    }
}

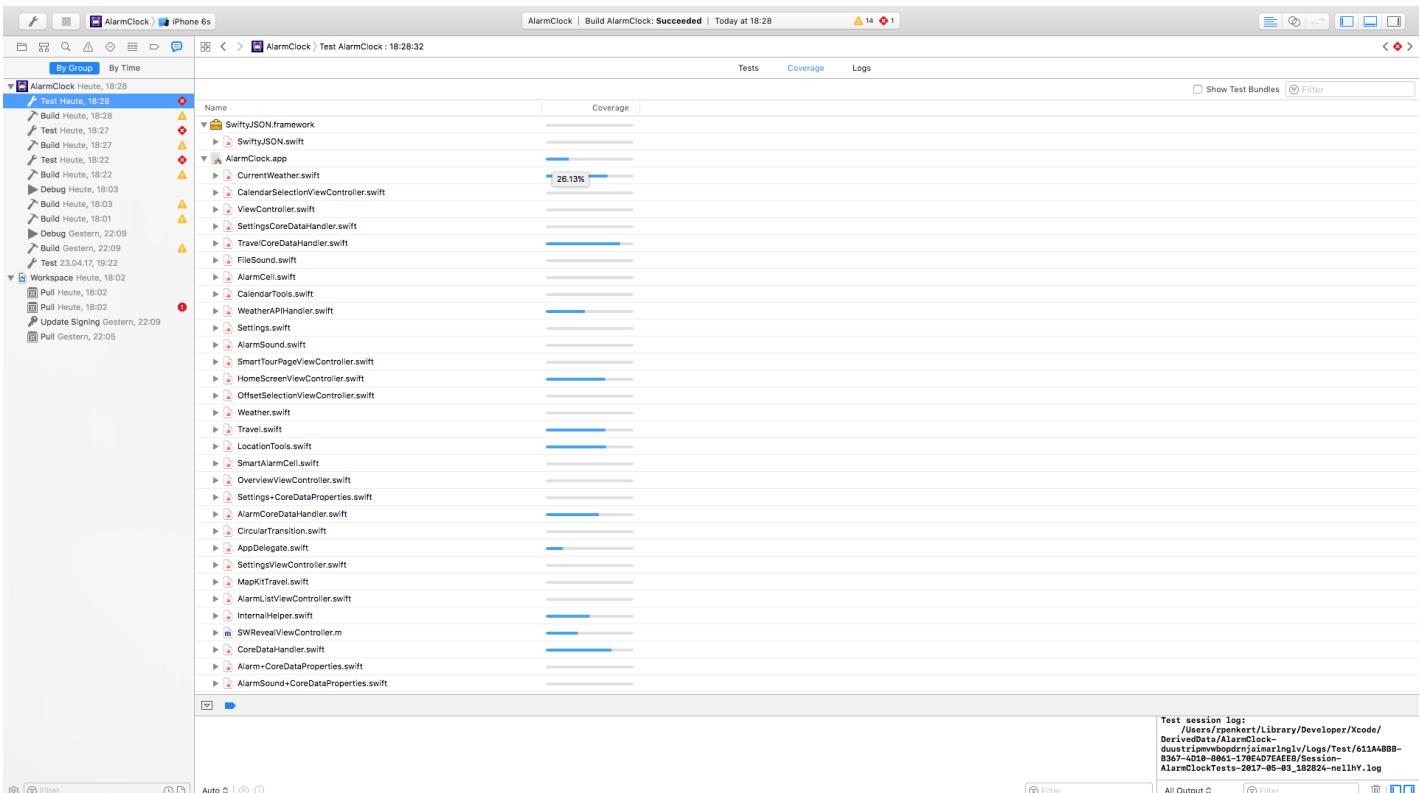
```

Test session log:
 /Users/rpenker/Library/Developer/Xcode/DerivedData/AlarmClock-doustrimjwvqyqzqjngl/Logs/Test/611A4B8B-B3E7-4D18-8941-17854D7EAE9B/Session-AlarmClockTests-2017-05-03_182824-nellhy.log

As you can see they are some succeed tests and one failed test.

In the Screenshot there is opened the Editor on the Source Code. You can see on the right side if your Source Code is covered by a test. If there is a red area with a zero the code in the editor is not covered. The number shows how often the Codeblock will be executed within all written tests.

Xcode have a Code Coverage Engine implemented. In the following Screenshot you can see how that is presented in Xcode:



You see all your Projectfiles with a bar that symbolizes how much of the file is covered by a test. When you scroll over a bar you can see a percentage. Our current overall Code Coverage is 26%.

We do not have a link for a maven file or something similar because for our project it is automatically build in our IDE “xCode”.

We provided a Code Coverage Badge on our Github Repo.

We documented everything in our [Test Plan](#)

← Function Points

Refactoring – Fowler →

2 thoughts on “Continuous Integration and Testing”



Torben Krieger says:

4. May 2017 at 07:13

Hi Team,

glad to see that you managed to set up CI for the Apple Universe.

Do you know where the warning come from?

Your work fulfills all grading criteria.

Best regards,

Torben

[Reply](#)



Dominik Wunderlich says:

11. May 2017 at 06:48

Hi Benedikt and René,

it's always very interesting to see how everything is done in the wo called Apple world.

I'm very happy to see that you have 31% code coverage as it is crucial to the quality of your iOS app.

Can you manage to get rid of the 12 warnings?

I think it's cool that you already managed to get a github badge. How did you do that?

i wish you all the best for your application and continued fun with software engineering homeworks 😊

Dominik from team SAM

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

POST COMMENT

Proudly powered by [WordPress](#) | Theme: [Sydney](#) by aThemes.



Patterns

Hey guys,

we implemented the Factory-Pattern.

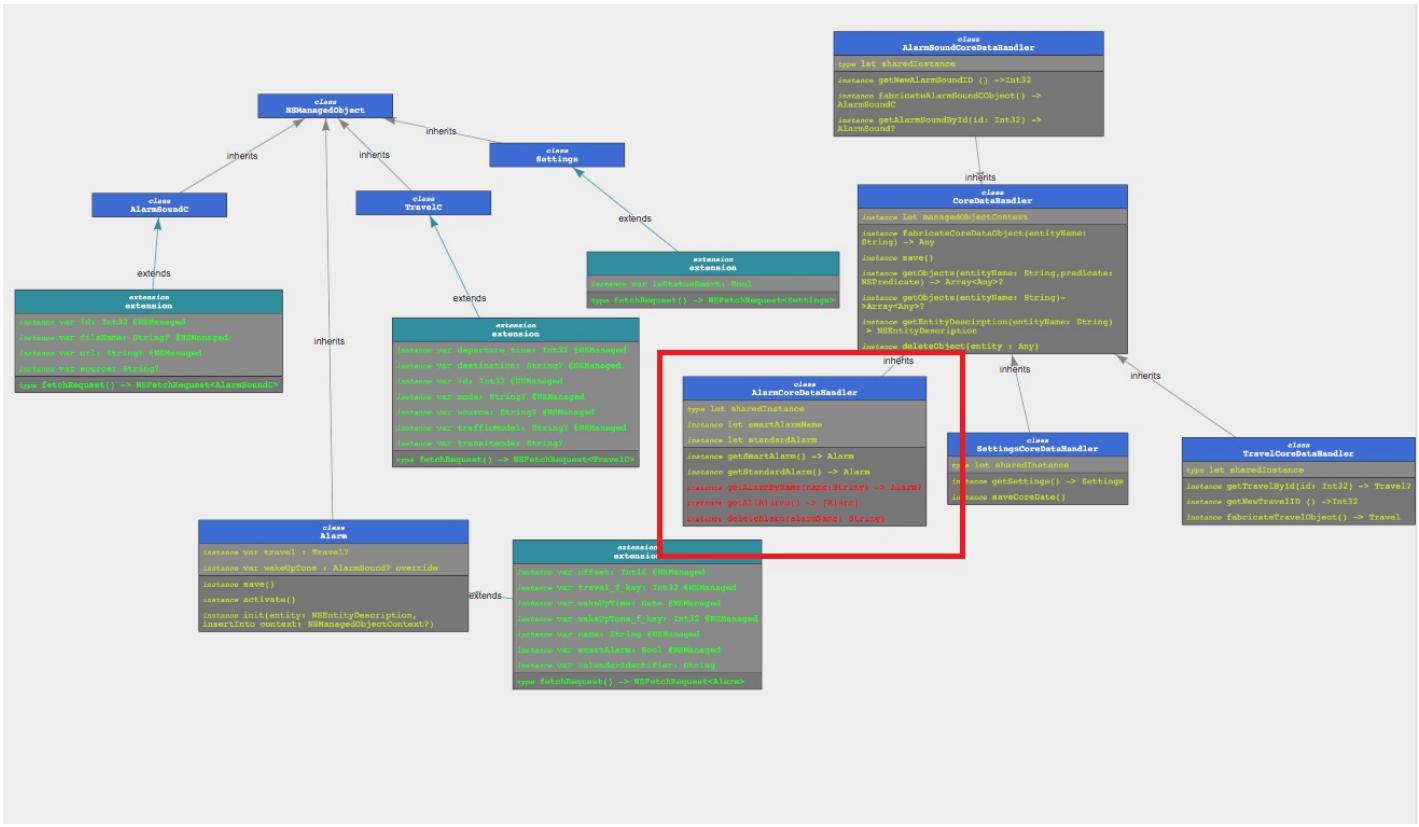
Our reasons for the Factory-Pattern:

- Wrap the constructor
- Generate a [Core Data Object](#) when generating a class instance

See the following code snippet showing the implementation:

```
func fabricateAlarm() -> Alarm {  
    let alarm = self.fabricateCoreDataObject(entityName: "Alarm") as! Alarm  
    alarm.id = getNewAlarmID()  
    return alarm  
}  
(source)
```

See the marks in our class diagram:



(source)

Warm regards

← Refactoring – Fowler

Metrics →

2 thoughts on “Patterns”



Dannynator says:

18. May 2017 at 07:56

Hey guys,

nice job implementing design patterns! You showed the changes in the code and the spot where you applied those changes in the overall class diagram, just as required in the GC. It would also be very nice if you could explain the Factory-Pattern itself, why is it necessary and why it fits into your application. Otherwise it is a bit hard for someone outside of the project to get a clear idea of what you have done there 😊

Also, when i click on the image it still kinda stays pretty small, so that it is not easy to read the text. Is it possible to make it a bit bigger or maybe scalable?

Hope your project is doing well and wish you all the best:)

Cheers!

[Reply](#)



Enrico Kaack says:

18. May 2017 at 08:17

Hey,

nice to see you are using the factory-pattern. It would be cool to see a previous/after UML diagramm to see the difference.

Best,

Enrico

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

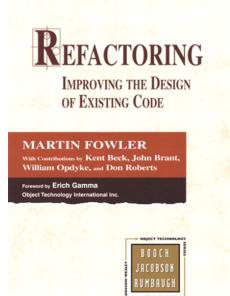
POST COMMENT

Proudly powered by WordPress | Theme: Sydney by aThemes.



Refactoring – Fowler

Check out our results of Refactoring based on Martin Fowler's [Refactoring](#):



To fulfill the grading criteria both of us done the exercise.

Benedikt Boshammer:

[GitHub](#)

René Penkert

[Github](#)

Warm regards!

[← Continous Integration and Testing](#)

[Patterns →](#)

2 thoughts on “Refactoring – Fowler”



[Dominik Wunderlich](#) says:

10. May 2017 at 21:59

Hi PenBo team,

cool to see that you read such a famous book like refactoring to improve your software engineering skills!

Unfortunately, I can only see Benedikt's coding exercise as there is no link provided for René's repository, is it?

As far as I looked over your code and commits, Benedikt, it looks really fine to me!

I just have one tip for you: Put your ".idea/workspace.xml" file to .gitignore so it does not manipulate your statistics or spams your commit report like here: <https://github.com/benediktbosshammer/Refactoring/commit/93399c83cbe546cb252d0f9b401adccd73cd748>

Keep on rocking!

Dominik from SAM

[Reply](#)



Torben Krieger says:

11. May 2017 at 07:12

Hello PenBo Team,

Fowlers book is a standard reference for software engineering... Or better to say, a book that every developer should read!

Your task was to reproduce the sample from the first semester but in contrast to the book you should use the facilities of your IDE. (The time the book was written this features are simply not available 😊)

To control the task on a proper way it was a guideline to execute this task under version control (-or better to say use Git + GitHub to host your results). Every refactoring step is recorded through a single commit. As result of this you should have at least 7 commits for refactoring.

Both of you exceed this threshold so everything is perfect! The links are also present. So everything that is verifiable is good the remaining points are not controllable from my view.

Best regards,

Torben

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

POST COMMENT

Proudly powered by WordPress | Theme: Sydney by aThemes.



Metrics

Hey guys,

this week we were asked to analyze our code base using a metrics tool.

Our analysis is based on two tools. The first one is the Command Line Tool [Taylor](#) working together with Xcode. Due to two problems that [Taylor](#) only produces JSON as the analysis result we decided to develop our own Command Line Tool based on Node.js around it named [Taylor-Parser](#) to show the results nicely in the browser.

Check out our tool! (on [github](#) or [npm](#))

We analyzed our code and decided to look at the issues with the following two metrics:

- Too Many Methods (TMM)

Number of methods in a class must not exceed given limit

- Cyclomatic Complexity (CC)

Methods must not exceed an admitted value of [Cyclomatic Complexity](#) (check for coding example)

Complexity is determined by the number of decision points in a method plus one for the method entry.

Our self written tool gave out the following report:

This is your result

33 Violations!

Rule	Violations
Excessive Class Length	0
Excessive Method Length	7
Too Many Methods	3
Cyclomatic Complexity	19
Nested Block Depth	2
N-Path Complexity	1
Excessive Parameter List	1

We picked the following:

The screenshot shows a UI element with a light blue header bar. Below it, a card-like interface displays a violation. The title is "TooManyMethods". The message says: "Class 'AlarmListViewController' has too many methods: 11. The allowed number of methods in class is 10". There is a grey button labeled "Expand" and the class name "AlarmListViewController" in orange.

and merged two into one to confirm to the maximum of 10 methods per class.

(See [this](#) commit with the changes)

This really improved the clarity of the class!

The tool also marked the following:

The screenshot shows a UI element with a light blue header bar. Below it, a card-like interface displays a violation. The title is "CyclomaticComplexity". The message says: "The method 'getAlarmSoundByld(id):' has a Cyclomatic Complexity of 6. The allowed Cyclomatic Complexity is 5". There is a grey button labeled "Expand" and the class name "AlarmSoundCoreDataHandler" in red.

But we decided not to change it because the Cyclomatic Complexity results from Apple's Core Data Mechanisms which *AlarmSoundCoreDataHandler* wraps.

This is our new results after the changes:

This is your result

32 Violations!

Rule	Violations
Excessive Class Length	0
Excessive Method Length	7
Too Many Methods	2
Cyclomatic Complexity	19
Nested Block Depth	2
N-Path Complexity	1
Excessive Parameter List	1

[Share results](#)

[Detailed Analysis](#)

32 left yeay 😊

We don't use automated deployment due to the fact that deployment means publishing to the app store in our case. That means that our tool won't be part of the deployment process!

Warm regards

← Patterns

Test & Installation →

3 thoughts on “Metrics”



Eynorey says:

1. June 2017 at 07:26

Hey there simplistic habitantes,

good to see you putting that much passion and effort into the project! You even wrote your own tool as there wasn't any for your environment.

Since you provided it on Github or via npm also, it'll likely help others in the future who find themselves in the same situation as you. So good work!

Also nice that you decided to check cyclomatic complexity (some cross-course knowledge going on here? 😊)

Anyhow, I got why you decided not to do something about it and instead will focus on the other fields to improve your code.

Kepp up the good work!

Greetings from Eynorey@SAM

[Reply](#)



Louisa says:

1. June 2017 at 07:40

Hey guys,

it looks pretty good! Cool, that you create your own tool!

You fulfil all of the grading criterias! You handed in code snippets, refactored your code and explained the metrics you use.

May you do well,

Louisa

[Reply](#)



Tom bendlrath says:

1. June 2017 at 07:42

Hey guys,

your metrics look great! I like the way you used your own command line tool which works fine with NodeJS. Even though, I am wondering if your metrics is part of your deployment? Moreover, you fulfilled the grading criteria by using 2 different metrics.

You linked your changes and also reached some improvement before-after. Additionally, you gave a comprehensive reason why you didn't change your code regarding the cyclomatic complexity violation.

By the way, you guys may want to chill a little bit when coding to prevent those excessive method length violations 😊

Kind regards,

Tom

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

POST COMMENT

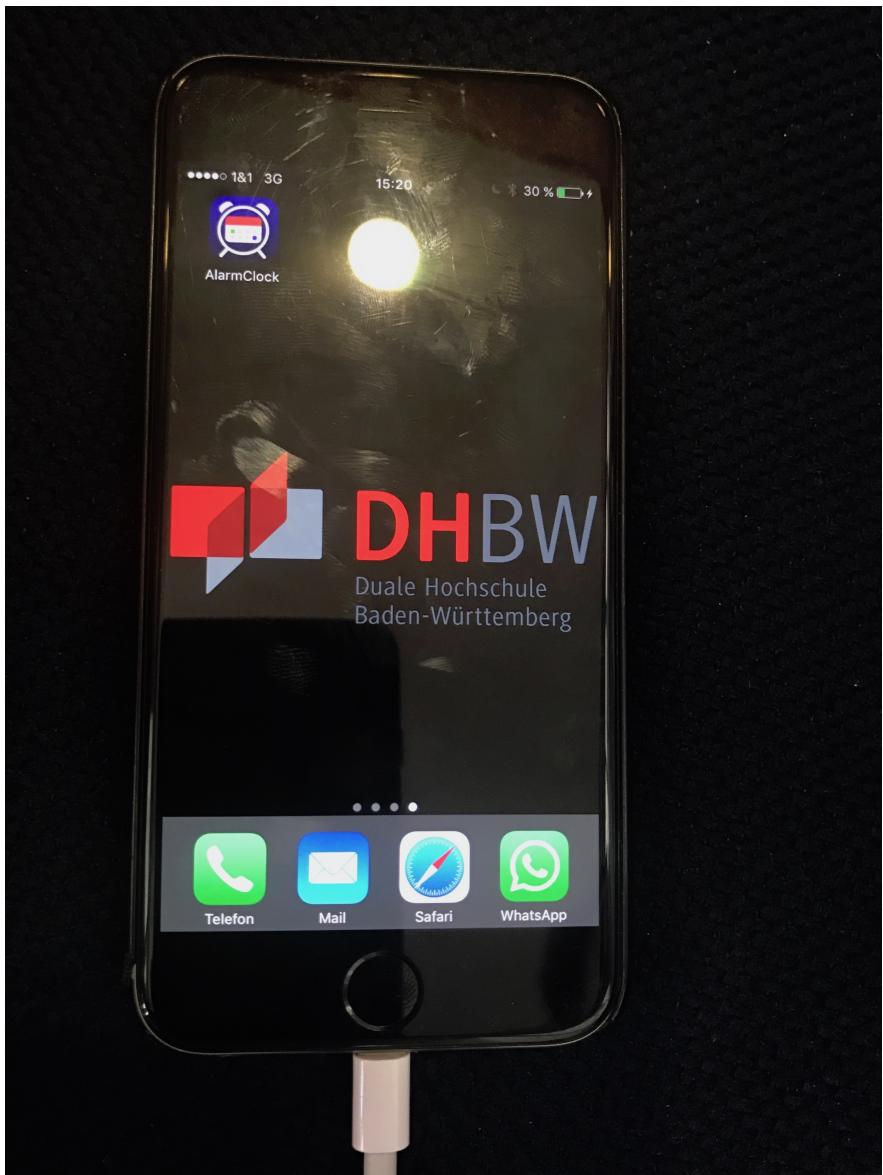


Test & Installation

Hey guys,

we installed the application to an iPhone 6 and iPhone 7 Plus:

iPhone 7 Plus:



As you see everything works fine!

Check out our Testing Blog Post for our Test / CI Setup.

As those of you who followed our History of Pain during Software Engineering class know that we had struggles to setup everything for conservative software engineering (as asked in the grading criteria). To fulfill at least the most grading criteria for CI & Testing we did everything possible for our technology.

Warm regards

[← Metrics](#)

[The END →](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

POST COMMENT

Proudly powered by WordPress | Theme: Sydney by aThemes.



The END

Links to the written documents for the Exam:

- [Vision](#)(Week 1,2)
- [Team](#)(Week 1,2)
- [Technology Choice](#)(Week 1,2)
- [Software Requirement Specifications](#) (Week 3)
 - [SRS Document on Github](#)
- [First two Usecases](#) (Week 4)
 - [Use case Select Calendar Document on Github](#)
 - [Usecase Select Create Alarm\(CRUD\) Document on Github](#)
- [Sprint](#) (Week 5)
 - [Sprintbord on Jira](#)
- [Feature-Files](#) (Week 6)
- [Class Diagramm](#) (Week 7)
 - [Class Diagram](#)
- [Software Architecture Document](#) (Week 8)
 - [SAD Document on Github](#)
- [Gantt-Chart](#) (Week 9)
- Midterm-Presentation
 - [.pptx](#)
 - [.pdf](#)
- [Function Points](#) (Week 10)
 - [Function Point Calculation on Github](#)
- [Continuous Integration and Testing](#) (Week 11)
 - [AlarmClockTest on Github](#)

- [WeatherAPITest on Github](#)
- [TravelCalculationTest on Github](#)
- [Testplan on Github](#)
- [Refactoring](#) (Week 12)
- [Pattern](#) (Week 13)
- [Metrics](#) (Week 14)
 - [Testplan on Github](#)
- [Test and Installation](#) (Week 15)
- The-End Präsentation
 - [.pptx](#)
 - [.pdf](#)

You can find our Code on [Github](#).

All important Files for SE on [Github](#)

Congratulations, that's it



Have a chocolate!

[← Test & Installation](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

POST COMMENT

Proudly powered by WordPress | Theme: Sydney by aThemes.