Teammitglieder: Sascha Hug, Tobias Lamm, Tim Saupp

Fach: Softwareengeering 1+2

Professor: Berklin, Kay Margarethe

Erstellt: 02.07.2017

Mathinator Blogeinträge als PDF

Month: April 2017

Midterm Handin

Hi there.

We've had a few problems with our blog and had to set up a fresh install.
So here are the documents from the midterm handin again:

GitHub Repo:
– Mathinator
YouTrack:
– Dashboard, Issues, Burndown Chart
Midterm Presenation:
– https://github.com/SaschaHug/Mathinator/blob/master/Pr%C3%A4sentation/Midterm2016.pdf
SRS:
– https://github.com/SaschaHug/Mathinator/blob/master/SRS/SRS-Midterm-Handin.pdf
Use Cases:
– https://github.com/SaschaHug/Mathinator/tree/master/Use%20Cases
SAD:
– https://github.com/SaschaHug/Mathinator/blob/master/SAD/Software%20Architecture%20Document_Midterm.pdf
Gantt-Chart:
– https://github.com/SaschaHug/Mathinator/blob/master/Ganttchart/Ganttchart.gif
Demo:
– https://github.com/SaschaHug/Mathinator/tree/master/Demo

Teammitglieder: Sascha Hug, Tobias Lamm, Tim Saupp

Fach: Softwareengeering 1+2

Professor: Berklin, Kay Margarethe

Erstellt: 02.07.2017

Welcome back !

Hi following community,

unfortunately we have to tell you that our Blog was crashed. But we have restored all important information in the Midterm Post again.

Now the new semester started and so we are getting started again. We defined the project scope for the following months. We will implement all Use Cases on our GitHub. In addition we will use a Framework to realize the AI.
We also checked how much time we spent on the use cases we implemented in the last semester. You can find this list on GitHub and we will create relevant reports on Youtrack (in progress).
Furthermore we thought about risks and made a risk plan, that can be found here.
Regards
Mathinator-Team

1. **Olga** says:
   May 2, 2017 at 9:59 am

   I’m glad to see that you are back!

   Your Risk List seems mostly fine, but I think you made a mistake in Line 2 “can’t find suitable framework”. Will “early testing” really work as a mitigation strategy on that?

   Apart from this, everything looks good.

   Wishing all the best

2. **TextVenturer** says:
   May 2, 2017 at 10:02 am

Teammitglieder: Sascha Hug, Tobias Lamm, Tim Saupp

Fach: Softwareengeering 1+2

Professor: Berklin, Kay Margarethe

Erstellt: 02.07.2017

Hey guys,

it seems like you had a lot of problems, and I hope you have everything back up and running without any issues again!

Your Risk-Management table is looking good so far though. It might not be the biggest list, but as you had all those problems, that is definitly understandable, as you had to focus on those problems first.

Keep it up, TextVenturer!

Teammitglieder: Sascha Hug, Tobias Lamm, Tim Saupp

Fach: Softwareengeering 1+2

Professor: Berklin, Kay Margarethe

Erstellt: 02.07.2017

Month: May 2017

Function Points Calculation

Last week we had a look at function points. So we calculated function points for each use case. With the evaluation about the function points and the spent time we can predict the time we will need for the following use cases.

For the calculation we used a excel sheet. Here you can find our result.
Regards
Mathinator

1. **AnKaLu Team** says:
   May 2, 2017 at 7:07 am

   Hi there,

   your Function Points Estimation looks well structured. I would like to mention one thing, where did you set your DET, RET, FTR for the different Transaction Data (External Inputs, External Outputs…) for each Use Case? Because if we look at your Excel Sheet, it is not specially defined.

   Best regards,
   AnKaLu-Team

2. **SquadIT** says:
   May 2, 2017 at 7:11 am

   Hey Mathinator,

   Your function point calculations look good so far. I would have liked it if you provided an overview as PDF file or picture directly in your blog to get a quick overview at the beginning.

Your Excel sheet looks nice. I like your coloring (1st semester/2nd semester) and you also provided reasons why a use case got the amount of FPs you gave them.

It’s amazing how your FP over ‘time spent’ diagram looks like. Almost all points are off only a little bit.

Keep up the good work,
greetings SquadIT

3. **TextVenturer** says:
   May 2, 2017 at 8:21 am

   Hey there,

   nice work on your Function-Point calculation!

   Having a short reasoning for each point was a great idea, and gives a nice overview, even for people, who just want to take a quick look at your project.

   The points match up with the averaged line fairly good.

   Greetings, TextVenturer.

4. **Sara** says:
   May 2, 2017 at 8:26 am

   Hi guys,
   your post looks really good.
   If you just could add some words about the outliner dots on your graph..
   Cheers,
   Cheetah

Test Plan

Unit Testing is an important part of Software Engineering.

So we decided to use JUnit in order to continually ensure our software's correct behaviour.

Since we're on Android, this is not as easy as one would expect. There are a few things we need to sort out before Unit testing can be automated. Here's a bit more Info for those who are interested and / or also on Android.
Mocking Dependencies in Android.
We hope to be able to sort this out until Week 7. Stay tuned!
We already integrated JUnited in our Gradle Build File, as you can see here.
You can view our test code here.
This is our Test-Plan, we invite you to take a look.

(Click to enlarge)

Until next week!

1. **Andreas** says:
   May 8, 2017 at 8:05 pm

   Hi Math Freaks,

   Thanks you for sharing your testing stuff with us! Your test plan looks good and is understable! Maybe you have a look at The Software "Sikuli" in combination with Android Studio / an other Android Desktop emulator for more UI testing!

Keep going!
Andreas, Project Ortus

2. **GTA Group**says:
   May 8, 2017 at 9:31 pm

   Hi Mathinator,
   nice to hear about you again!
   It seems, like you really know what you are doing and what tools could be the best choice for your projekt 😉
   We are really excited to hear about your results in several weeks.

   I looked your text documentation and noticed, that it contains emply tables, are you going to fill them later? If not, it is better to delete it from documentation.

   Best regards and have a nice week!
   <3

Teammitglieder: Sascha Hug, Tobias Lamm, Tim Saupp

Fach: Softwareengeering 1+2

Professor: Berklin, Kay Margarethe

Erstellt: 02.07.2017

Refactoring

This week we refactored a sample repository for training. We used the book:

Refactoring: Improving the Design of Existing Code by Martin Fowler.

- Sascha’s Repository
- Tim’s Repository
- Tobi’s Repository

See you!

1. **Project Ortus**says:
   May 16, 2017 at 8:37 am

   Hi,

   nice commits.
   I bet you saved yourself some time with your IDE. 😉

   Cheers,

   Team Ortus

2. **JoinSports**says:
   May 16, 2017 at 9:00 am

   Hi,

   I think you did a good job. Your refactored code looks nice.
   Sascha could have split his changes in more different commits. So he would have the required number of commits.

   Greetings,
   the JoinSports-Team

3. **GTA Group** says:
   May 16, 2017 at 9:31 am

   Hey Mathinator!
   It seems like you have made your job successfully! Most of you have more that 6 commits – it is really nice to know, you are taking it serious.

   Keep going!
   We are waiting for your next homework already 😉
   <3

4. **SquadIT** says:
   May 22, 2017 at 10:41 am

   Hey guys,

   Your repos look good. You have tests and multiple commits with several changes. Keep up the good work.

   Greetings,
   SquadIT

Teammitglieder: Sascha Hug, Tobias Lamm, Tim Saupp

Fach: Softwareengeering 1+2

Professor: Berklin, Kay Margarethe

Erstellt: 02.07.2017

Patterns

Today we'd like to talk about patterns in Software Engineering.

In larger software projects, patterns will make your life easier (that is, if you use them properly). They will speed up the development process by providing tested and proven development methods.

In our project we used a so-called "Interpreter Pattern" which is supposed to implement a domain specific language (in our case a calculator interpreting arithmetic statements).

There are serveral steps involved in using and creating such a pattern:

1. Design a grammar for said language
2. Each production (e. g. PlusOperation -> Number '+' Number) needs to be mapped to a function
3. All of these functions are specified by a composite pattern (We used an Interface that defines the relevant methods)
4. The context of each method must then be interpreted. This is done by a Visitor and a VisitorImplementation Class (which implements the methods given by the interface)
5. The context is then interpreted until all nodes of the syntax-tree are resolved (and thus the result is calculated)

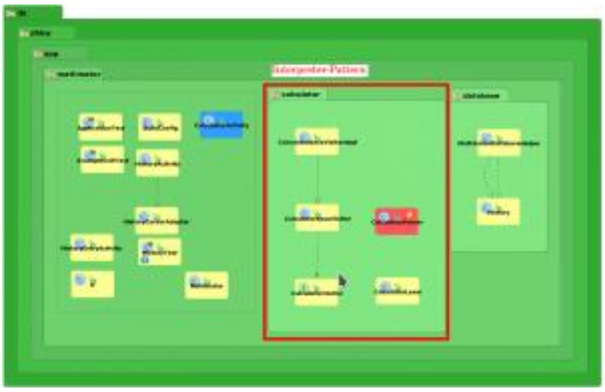
   We used this pattern because is makes implementing a calculator a lot easier. When you have to solve a lot of similar problems, it makes sense to define a interpreting language for it (e. g. regular expression or arithmetic operations). Also, we don't have to rely on complex algorithms that sort out operator precedence for us. We are also able to extend the grammar quite easily using this pattern.

Teammitglieder: Sascha Hug, Tobias Lamm, Tim Saupp

Fach: Softwareengeering 1+2

Professor: Berklin, Kay Margarethe

Erstellt: 02.07.2017



(click to enlarge)

1. **GTA Group** says:
   May 22, 2017 at 11:48 pm

   Hey Mathinator!

   It seems like Interpreter Pattern really suits for your project. Have you also used SOLID principles for your code?

   Thank you for the detailed explanation and the visual presentation!

   Keep in touch!
   See you:)

2. **Sara** says:
   May 23, 2017 at 6:27 am

   Hey guys,
   nice pattern.
   Still, could you add a comparison between the picture with the pattern and without pattern.
   Cheers
   Cheetah

Teammitglieder: Sascha Hug, Tobias Lamm, Tim Saupp

Fach: Softwareengeering 1+2

Professor: Berklin, Kay Margarethe

Erstellt: 02.07.2017

Metrics

To assess our code quality we integrated 'Metrics Reloaded' into our workflow.
You can see the results here: before and after.

We took a closer look at the following metrics:

WMC – Weighted methods per a class's *weighted methods*. WMC metric is simply the sum of the complexities of its methods.
LCOM – Lack of cohesion in methods
A class's *lack of cohesion in methods* (LCOM) metric counts the sets of methods in a class that are not related through the sharing of some of the class's fields.
Average Operation Complexity
The average cyclomatic complexity of all the non-abstract methods in the project.

The Metrics tool gave us a bad score for some classes, but those were autogenerated by our Interpreter Engine (see the previous post for more info on that), so we are obviously not going to change anything about that.

So long,

Team Mathinator

1. **TeamOrtus** says:
   May 29, 2017 at 9:36 pm

   Hi Team Mathinator,

it is a valid point to keep auto-generated code. So we agree with what you did.

Keep it up.

Regards,

Team Ortus

http://projectortus.abuchmann.de/

2. **Waldemar** says:
   May 30, 2017 at 7:07 am

   Hello Mathinator,

   good work! You really explained good understandable how your metrics work.
   Is this tool a part of your deployment process? If yes can you tell us where you start it?

   Best regards
   Waldemar

3. **SquadIT**says:
   May 30, 2017 at 7:52 am

   Hey Mathinator 🙂

   The metrics tool you use is a plugin for your IDE?

   Your metrics look good so far – we also agree with you regarding the auto-generated code.

Teammitglieder: Sascha Hug, Tobias Lamm, Tim Saupp

Fach: Softwareengeering 1+2

Professor: Berklin, Kay Margarethe

Erstellt: 02.07.2017

Could you include your “nachher” link in your blog post? That way it would be easier to reach.

Greetings,
SquadIT

Teammitglieder: Sascha Hug, Tobias Lamm, Tim Saupp

Fach: Softwareengeering 1+2

Professor: Berklin, Kay Margarethe

Erstellt: 02.07.2017

Month: June 2017

Installation guide

Hey Guys,

today we will show you how to install our app on your Android phone:

Please follow these steps:

1. You will need an Android phone with version 5.0 or higher in order to proceed
2. Go to Settings -> Security -> Unknown sources. Allow installation of apps from unknown sources here (meaning the .apk file will not come from the PlayStore, but from your SD card / our GitHub instead)
3. Download and open the .apk from here.
4. Simply follow the install procedure
   Congrats! You should now have our app installed.

If you have quetions, feel free to ask. We will be trying to assist you.

Greetings, Mathinator

1. **Waldemar** says:
   June 13, 2017 at 9:50 am

   Hi Mathinator,

   well done so far!
   I have tested your installation guide and installed your app.
   It works fine. Your installation process is very clear and easy to carry out.

   Keep on!
   Greetings, Waldemar

2. **AnKaLu Team** says:
   June 19, 2017 at 9:49 am

   Hi,

   we have tested your App, everything worked fine. Nice design and the instructions are very detailed.

   Keep up the good work!
   AnKaLu-Team

Teammitglieder: Sascha Hug, Tobias Lamm, Tim Saupp

Fach: Softwareengeering 1+2

Professor: Berklin, Kay Margarethe

Erstellt: 02.07.2017

Month: July 2017

Final Blog Post

The semester is over and we've already taken the oral exam at this point. Thus, this will be our last blog entry, in which we sum up our progress so far.

Mathinator on GitHub

Our repository can be found on GitHub.
The files and blog posts for the written hand-in are divided into categories as follows:

Requirements

- Use Case Documents
- Software Requirements Specification (SRS)

Testing

- Test Cases (.feature files) and Test Screenshots:
- Unit Tests and Test Screenshots:
- Test Plan

Project Management

- MS Project / Gantt Chart
- YouTrack Dashboard / Burndown Charts
- Long term planning with function points (FPs)

Ability to Execute

- Installation Guide
- Peer Review for above guide
- Demo and APK for installation

- The actual code

Quality

- Metrics

- Risk Management
- Automatic Integration with TravisCI

Architecture:

- Software Architecture Document (SAD)
- Our approach to patterns (also explained in the SAD)

Our final presentation

The presentation can be found here.

Phew! Looks like this is about it.