Andreas Lenz:

Localization
Who? Andreas Lenz
When? 17.05.2017 - 18.05.2017 about 1 hour
Why? To be able to change languages in the app at runtime.
What? Added Resource bundles with localized texts and a resource manager class to support localized texts.

Team Meeting
Who? Andreas Lenz
When? 19.05.2017 about 1 hour
Why?To organize and discuss progress
What? We discussed the work performed so far.

Extreme coding session
Who? Andreas Lenz
When? 20.05.2017 about 9 hour
Why?To make progress on the project
What? We met up and coded all day. I Implemented the table model and connection of the swing classes to our internal representation (model classes). We made extensive use of gitflow to exchange temporary progress and discuss minor changes. Our code went through several revisions and we discussed and adapted the code. Since we didn't have a complete class diagram we kind of had to make it up as we went.
Started work on the documentation.

Factory Package
Who? Andreas Lenz
When? 20.05 about 30 mins
Why? To be able to use the Factory
What? FIxed up the ABstract factory pattern and defined the factory for cell creating. Implemented all the functions.

File handling strategy
Who? Andreas Lenz
When? 20.05 about 2 hours
Why? To have an interface for loading and saving files
What? Created the file loader and file saver classes that use strategy pattern to either save as csv or ods. Also integrated the CSV and ODS functionality as soon as the interfaces where available.

Swing Frontend TableModel

Who? Andreas Lenz

When? 21.05.2017 about 2 hour

Why? To allow dynamic exchange of table data and more control over the internal representation of our Cells

What? Modified the SwingFrame class to act as a wrapper for all spreadsheet related work. Created the NAPATableModel and a wrapper class SwingTable. Partially defined the interfaces and integrated it with the SwingFrame class.

Swing Frontend Cell Editing

Who? Andreas Lenz

When? 23.05, 24.05. And 25.05.about 2 hours

Why? To control cell editing

What? Created a custom cell editor NAPACellEditor.

Aldin Bradaric:

- **charts**

  Who? Aldin Bradaric supported by input from Nikola Golubovic

  When? 23rd of May, for roughly two hours

  Why? The purpose was to implement line and bar charts. Nikola and I discussed the solution and while I was coding it, he made comments and suggestions as to how particular parts could be improved.

  What? The implemented line and bar charts for the suggested datasets were completed and fully functional.

- **csvparser**

  Who? Aldin Bradaric

  When? 21st and 22nd May iirc, a few hours were at first spent to find a suitable parser, the rest was then done rather first (ie. coding)

  Why? In order to be able to read and write/save CSV files

  What? A fully functional CSV parser that takes CSV files and displays them inside a NAPATable model

- **cells**

  Who? Aldin Bradaric, a few things were later edited by Andreas Lenz

  When? 19th of May iirc, roughly two hours

  Why? In order to use an observer pattern for the cells, I created two interfaces and four classes, each one being either an “Observer” or a “Subject”.

  What? An observer pattern for the different types of cells, enabling cells to listen to changes of other cells they are "subscribed" to (ie. those they are observing) and on the other hand be Observables themselves (ie. give other cells the opportunity to subscribe to them). Number cells for examples extend the Subject interface whereas Formula cells implement both the Subject and the Observer interface as they require all of the functionalities.

- **charts**

  Who? Aldin Bradaric + Andreas Lenz

  When? 24th + 25th of May, 3-4 hours

  Why? As the chart creation needed to be dynamic, appropriate frontend plus backends needed to be implemented. The above mentioned work wasn’t sufficient. We chose to go with Linecharts and Barcharts again.

  What? I started working on the structure of the code (ie. finishing interface and creating+working on the classes of the backend) and then asked the other guys for help as we were running low on time.

- **documentation**
  Who? Aldin Bradaric
  When? 24th of May, roughly an hour + 25th of May, an hour as well I'd reckon
  Why? .docs needed to be finished
  What? As Andreas Lenz had started working on the barebone of the respective .doc files, I finished them by adding the needed code examples from our project and adding a line of text here and there. The barebone itself was anything but actually.
Additionally, I went through my code and changed it to conform to various coding and documentation standards.

Philipp Zettl

Erstentwurf des Pattern
Who? Philipp Zettl
When? 19.05.2017 ca. 2.5 h
Why? Wir hatten zuvor ein Meeting wo wir das finale Klassendiagramm besprochen hatten.
What? erster Entwurf des Patterns für die Formel Verwaltung

Shunting-Yard-Algorithmus
Who? Philipp Zettl
When? 21.05.2017 ca. 3h
Why? Da wir die Formeln in einem String übergeben bekommen von der Formelzelle, war es notwendig diese so zu zerlegen, damit die einzelnen Strings dann in einen berechenbaren Datentyp zu parsen.
What? In der Klasse "Formula Evaluator" wurde der Shunting-Yard-Algorithmus implementiert und getestet um Infix-String-Arrays in Postfix-arrays zu sortieren.

Zerlegung der Formel-Strings
Who? Philipp Zettl
When? 22.05.2017 ca. 4 h
Why? Um den Shunting-Yard-Algorithmus verwenden zu können ist es notwendig aus einem String, ein String-Array zu machen um es dann entsprechend umsortieren zu können.
What? Es wurde eine Split-Methode implementiert welche "=, (,),+,-,/,*,SUM,COUNT,MEAN,:,$" erkennt und sie splittet. Danach wurde mit einem Evaluator die Ergebnisse aus den Funktions-Formeln ermittelt und im String-Array durch diese ersetzt. Danach konnte das Array dann postfix-sortiert werden.

Funktionsklassen
Who? Philipp Zettl
When? 23.05.2017 ca. 3.5 h
Why? Für die finale Berechnung der Formeln sind die Ergebnisse aus den Funktionsklassen SUM, COUNT, MEAN nötig.
What? Implementierung der Calculations Logik in den jeweiligen Funktionsklassen.

Finale Formelberechnung
Who? Philipp Zettl
When? 24.05.2017 ca. 5 h
Why? Für die finale Berechnung der Formeln sind die Ergebnisse aus den Funktionsklassen SUM, COUNT, MEAN nötig.
What? Implementierung des Refactoring der Ergebnisse der Formeltypen in die original-Formel und Erstellung einer Rechnerarchitektur für die finale Berechnung des Postfix-Ausdrucks. Überprüfung der Ergebnisse und Anpassungen aufgrund einiger Fehler im System.

Bug fixing, Kommentieren, Dokumentation

Who? Andreas Lenz, Philipp Zettl

When? 25.05.2017 ca. 4 h

Why? Um den Code den Richtlinien entsprechend zu halten, und das System fehlerfrei lauffähig zu machen.

What? Kommentieren der restlichen Methoden, Formatierung des Codes nach Richtlinien. Erweiterung der Falsch-Eingabemöglichkeiten ohne Systemabsturz und damit Gewinnung von Systemstabilität. Fertigstellung der Dokumentation.

Nikola Golubovic

- **Charts**
  - ➢ *Who?* Nikola Golubovic and Aldin Bradaric
  - ➢ *When?* 23.05.2017, about 2 hours.
  - ➢ *Why?* The purpose was to implement line and bar charts. Aldin and I discussed the solution and while Aldin was coding it, I made comments and suggestions as to how particular parts could be improved.
  - ➢ *What?* The implemented line and bar charts for the suggested datasets were completed and fully functional.

- **ODF Parser**
  - ➢ *Who?* Nikola Golubovic
  - ➢ *When?* 20.05.2017, about 7 hours.
  - ➢ *Why?* In order to be able to read and write/save ODF files.
  - ➢ *What?* A fully functional ODF parser that takes ODF files and displays them inside a NAPATable model.

- **Factory Package**
  - ➢ *Who?* Nikola Golubovic, later it was controlled by Andreas Lenz
  - ➢ *When?* 19.05.2017
  - ➢ *Why?*
  - ➢ *What?*