

Schlussbericht Informatikpraktikum

Oliver Kamer

14. Oktober 2020

Das Praktikum im Rahmen des Informatikpraktikums der Universität Zürich wurde im Sommer 2020 bei der Eawag¹ absolviert. Das Praktikum dauerte 12 Wochen mit einem Arbeitspensum von 100%.

Das Ziel des Praktikums war es, eine alte Kombination von Datenbank und Applikation neu zu entwickeln und die Daten von der Applikation ein wenig zu trennen. Die Datenbank wird von Forschenden an der Eawag zur Ablage der Edelgasdaten benutzt.

1 Praktikumsbericht

Das Praktikum kann grob in drei Teile unterteilt werden.

1. Zuerst musste, ich mich mit dem alten System vertraut machen, die eigentliche Nutzung des Systems verstehen und dokumentieren. Insbesondere brauchte es ein gewisses Verständnis der Arbeitsprozesse und zu einem kleineren Teil auch der Materie.
2. In einem zweiten Schritt musste das neue System skizziert, definiert und geplant werden. In der Diskussion mit dem Betreuer wurde die Datenstruktur und die Funktionen der Applikation geplant und beschrieben.
3. Anhand der Planung im zweiten Schritt habe ich das System dann Schritt für Schritt in Code umgesetzt. Die Umsetzung wurde bereits ziemlich bald auf eine Virtual Machine, welche durch die interne IT bereitgestellt wurde, deployed und getestet. Zuletzt wurden die Daten der alten Datenbank in die neue Applikation migriert.

1.1 Aufnahme

Am ersten Arbeitstag wurde ich mit viel Information versorgt. So habe ich ein Buch mit dem Prozess der Edelgasmessungen bekommen. Der Prozess wurde mir mündlich erklärt und auf meine teils naiven

Fragen wurde eingegangen und sie wurden beantwortet.

Danach wurde mir der Arbeitsprozess im Labor gezeigt. Anhand der vielen Laborgeräte wurde mir die Arbeitsweise im Labor und die dazugehörigen Prozesse in Bezug zur Datenbank und Applikation erklärt. Die Menge der Informationen am ersten Tag war sehr gross, aber ich habe später immer wieder auf diese Ideen und Arbeitsweise und -schritte zurückgegriffen.

Danach habe ich diese Arbeitsprozesse in natürlicher Sprache dokumentiert und diese Dokumentation der Arbeitsschritte in enger Zusammenarbeit mit meinem Betreuer überarbeitet und in teilweise strukturierter Art notiert.

1.2 Ausarbeiten

Auf Basis der Notizen und Dokumentation der Arbeitsprozesse begann ich anschliessend ein neues System zu konzipieren. Dazu kamen auch noch einige wichtige Kriterien an das neue System: die Daten sollten von der Applikation getrennt sein und das neue System soll irgendwo auf einer Virtual Machine laufen.

In einem ersten Schritt wurde die Datenstruktur für das neue System designt. So wurde die Struktur nicht aus dem alten System übernommen, sondern ein neues konzipiert, welches gewisse Vorteile haben sollte. So wurden ungefähr die Hälfte der Felder übernommen, allerdings wurde die Felder neu strukturiert und viele auch weggelassen. Dazu gekommen ist nur ein neues Schema für Benutzer. Abbildung 1 zeigt das alte Datenbankschema, Abbildung 2 das neue.

Danach wurde auch die Umsetzung und die dazugehörige Technologie Stack definiert. Abbildung 3 zeigt eine Folie aus der Präsentation bezüglich den Technologien. Diese Präsentation wurde mit meinem Betreuer und der IT-Abteilung der Eawag besprochen.

Des Weiteren wurden die einzelnen Arbeitsprozesse zu in User Stories erfasst und zu verschiedenen

¹<https://www.eawag.ch/>

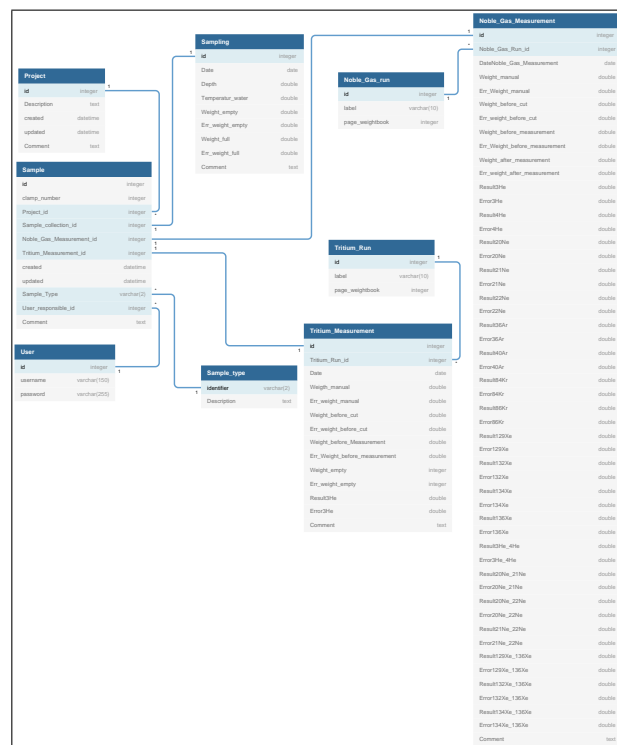
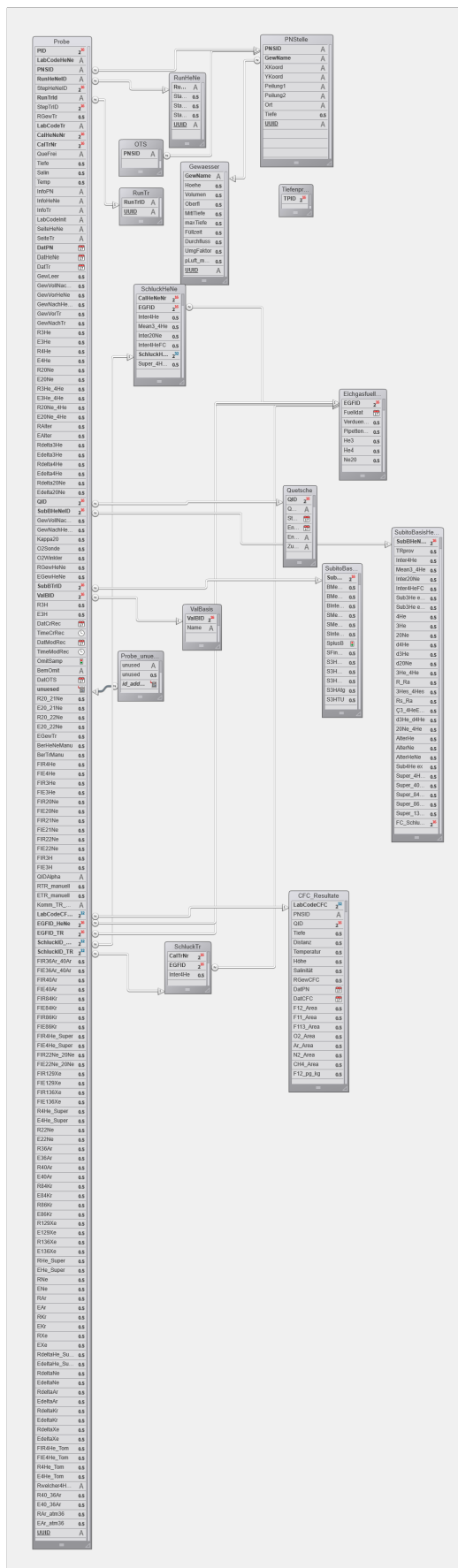
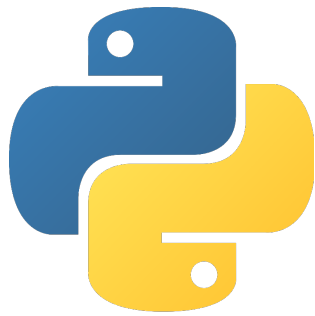


Abbildung 2: Neues Datenbankschema



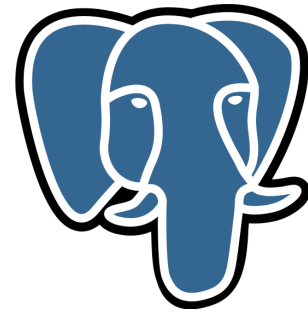
Python

- Weit verbreitet
- Viele Pakete
- Lesbar

django

Django

- Schnelle Entwicklung
- Batterien inklusive
- Rad nicht neu erfinden



PostgreSQL

- Zuverlässig
- Schnell
- Von IT gewünscht/vorgeschrieben

Abbildung 3: Folie aus der Präsentation zu den Technologien

Sprints zugeteilt. Die einzelnen Sprints wurde auf eine Dauer von ungefähr zwei Wochen gesetzt, so dass es mehr oder weniger auf das zweiwöchentliche Gruppen Meeting fiel.

1.3 Umsetzung

Die Umsetzung der Applikation erfolgte nach dem ausgearbeiteten Plan in der Planungsphase. Die Entwicklung mit dem Django Framework wurde auf *GitHub* gehostet und auch versioniert. Als Entwicklungsablauf habe ich ein Modell benutzt, welches stark von *GitFlow*² und weniger auch von *GitHub flow*³ inspiriert war. So wurde alle abgeschlossenen Feature Branches in der Main Branch auch bereits auf dem Web zum Ausprobieren bereitgestellt.

Ungefähr einmal pro Woche habe ich den aktuellen Stand der Dinge meinem Betreuer präsentiert. An den Teammeetings habe ich den jeweiligen Stand kurz präsentiert und nach Feedback gefragt. So kamen auch immer wieder gute Rückfragen und Kritikpunkte, welche geklärt oder verbessert werden konnten.

Viele Probleme konnten bereits früh erkannt werden. So zum Beispiel das Problem der Dateiberechtigung in der Applikation, und auch verschiedenste

Probleme mit der Konfiguration des Webservers⁴.

Während der Entwicklung merkte man dann bald, dass es keine gleichzeitigen Schreibzugriffe geben wird. So konnte man dann die Datenbank *SQLite*⁵ anstelle von *PostgreSQL*⁶. Dies hatten den entscheidenden Vorteil, dass wir uns nicht um einen Datenbank-Server kümmern mussten und der Prozess für das Backup sehr einfach wurde. So konnte man ganz simpel eine Kopie der `db.sqlite3` angefertigt werden und dann an einem anderen Ort abgesichert werden.

Am Ende wurde dann auch noch ein wenig Caching für zwei Operationen eingeführt. So wurde zum Beispiel der Grossteil des Dashboards für einige Zeit im Cache abgelegt, da die Berechnung nicht ganz sofort war und sich die Daten nur sehr träge veränderten (siehe Abbildung 4).

2 Schlussreflexion

Über alles gesehen würde ich mein Praktikum als sehr erfolgreich beschreiben. Das Projekt wurde umgesetzt und läuft aktuell für die Forschenden und wird auch genutzt. Zudem habe ich viel gelernt, insbesondere auch die Projektarbeit in einem Team,

²<https://nvie.com/posts/a-successful-git-branching-model/>

³<https://guides.github.com/introduction/flow/>

⁴<https://httpd.apache.org/>

⁵<https://www.sqlite.org/>

⁶<https://www.postgresql.org/>

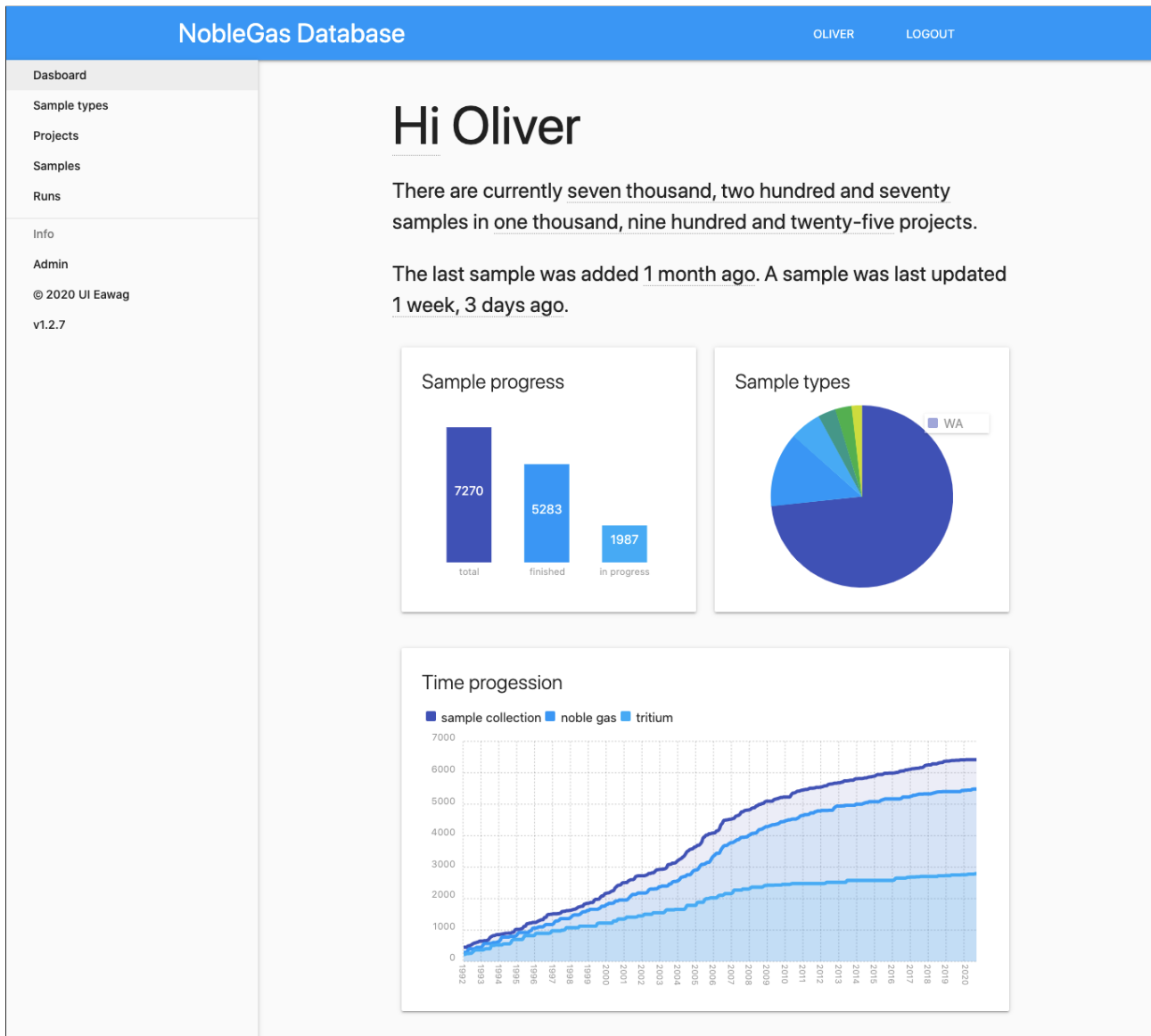


Abbildung 4: Screenshot des Dashboards

die effiziente Kommunikation mit Personen, welche nicht den gleichen Hintergrund haben und auch das eigentlich Hosting von einer Webapplikation.

Als Nebeneffekt habe ich auch noch einen Einblick in die Forschung an der Eawag erhalten und so ein interessantes, bis jetzt mir unbekanntes, Arbeitsfeld gesehen.

2.1 Analyse und Spezifikation

2.1.1 Dokumentation der aktuellen Prozesse

In der Planung wurde für die Analyse und Spezifikation zwei Wochen geplant. Die Dokumentation der Prozesse bin ich dann gleich in der ersten Woche angegangen. Da ich aufgrund der COVID-19 Situation im Homeoffice war und so manchmal ein wenig auf Antworten zu den aktuellen Prozessen warten musste, habe ich auch bereits mit einem Entwurf für das neue Datenbankschema begonnen.

Die Dokumentation der aktuellen Prozesse wurden ungefähr Mitte der zweiten Woche durch meinen Betreuer abgenommen. Dies war allerdings kein grosser Verzug, da die Konstruktion des neuen Datenbankschemas bereits schon fortgeschritten war.

2.1.2 Spezifikationen des neuen Systems

Die Spezifikation des neuen Systems ging erheblich schneller als geplant. Mein Vorschlag für die Technologie wurde sofort akzeptiert, die Funktionen waren ziemlich klar. Die verwendeten Technologien waren also bereits eine Woche vor dem Zeitplan klar.

Hier zeigten sich bereits erste Kommunikationsschwierigkeiten. Ich habe nicht immer ganz verstanden, was mein Betreuer meinte, aufgrund von fehlendem Chemie/Physik Kenntnissen und umgekehrt verstand meine Informatikerklärungen nicht immer. Dies erkannten wir sehr früh und so mussten wir früh ein gemeinsames Vokabular entwickeln und die Kommunikation mit vielen Rückfragen verbessern.

Die Spezifikation des neuen Systems inklusive der ganzen Technologie wurde am Ende der zweiten Woche präsentiert und so auch akzeptiert. Natürlich wurde im Laufe der Entwicklung immer einige Aspekte überarbeitet und verbessert.

2.2 Entwicklung

2.2.1 Definieren der User Stories

Die User Stories wurden nach der Spezifikation schnell definiert. Das Format der User Stories wurde nur ungefähr eingehalten. Generell war die Idee

die User Stories einzuhalten, wann immer möglich, aber nicht dadurch eingeschränkt zu werden.

2.2.2 Klassendiagramm und Datenbankstruktur

Die Datenbankstruktur wurde bereits in einem früheren Schritt erstellt und diese Aufgabe entfiel also in dieser Woche. Auf das Klassendiagramm wurde dann auch verzichtet, da es wenige Unterschiede zum Datenbankdiagramm gab und die Funktionen der einzelnen Klassen noch nicht ganz klar waren.

2.2.3 Programmiersprints

Die Programmiersprints haben sich alles ziemlich arbeitsintensive, aber klar definierte Schritte, gezeigt.

Das Vorgehen war jeweils ziemlich identisch für jede Woche: Ich habe ein neues Feature anhand der User Stories implementiert. Nach der Implementation und dem Testen habe ich dann das Feature deployt. Danach habe ich den aktuellen Zustand (inklusive neuem Feature) meinem Betreuer präsentiert (manchmal auch der ganzen Gruppe) und nach Feedback gefragt. Meistens konnte dieses Feedback dann sehr schnell eingearbeitet werden, manchmal habe ich das Feedback auch einfach in den Backlog gespeichert und dann in einem späteren Schritt umgesetzt.

Dies hatte den Vorteil, dass Probleme in der Umsetzung, sei es im Umfeld des Servers oder auch im Verständnis oder Logik von einer End-Benutzersicht, früh erkannt und dann auch gleich gelöst werden konnten.

Aufgrund der zusätzlichen Arbeiten wie Feedback einholen und dieses auch umzusetzen ging das Ganze länger als in der Zeitplanung vorgesehen. Am Ende konnten aber alle Aufgaben erfolgreich umgesetzt werden (insbesondere auch, weil dafür spätere Schritte weggefallen sind).

2.2.4 Feedback einarbeiten

Dieser Schritt entfiel praktisch komplett, da im Entwicklungsprozess immer wieder und zeitnah Feedback eingeholt wurde.

Der Schritt, das neue System dann auch zu nutzen, gestaltete sich dann auch schwierig, da das neue System nicht sofort akzeptiert wurde. Die Umstellung kam dann erst richtig, als das alte System unsichtbar gemacht wurde.

2.3 Migration

2.3.1 Planung der Migration

Die Migration habe ich zusammen mit meinem Betreuer geplant, in dem wir aus der alten Datenbank alle Attribute sammelten, welche dann in irgend einer Form in der neuen Datenbank landeten.

Glücklicherweise konnten die meisten Attribute mehr oder weniger einfach übernommen werden. Einige Attribute mussten jedoch verrechnet werden, was eigentlich der grösste Teil des Aufwandes darstellte.

Am Ende hatte ich ein Skript, welches aus den `.xml` Dateien der alten Datenbank die Daten direkt in die neue Datenbank importierte. Das Skript stand rechtzeitig für die Migration bereit.

2.3.2 Durchführung der Migration

Durch die ausgiebige Planung und Vorbereitung der Migration, gestaltete sich die Durchführung dann umso einfacher.

So konnte die Migration dann einfach an einem Abend, um niemanden bei der aktiven Arbeit zu stören, in vier einfachen Schritten durchgeführt werden:

1. Aktueller Stand der alten Datenbank in `.xml` Dateien exportieren.
2. Die `.xml` Dateien auf den neuen Server hochladen.
3. Das Import-Skript starten.
4. Erfolgreicher Abschluss des Skripts und Imports überprüfen.

Am nächsten Morgen habe ich dann den Import kontrolliert und gesehen, dass alles erfolgreich abgeschlossen wurde. So brauchte es keine zusätzlichen Korrekturen von Hand.

2.3.3 Allfällige Verbesserungen

Aus eigener Beobachtung und auch Feedback von den Benutzern gab es dann noch vereinzelte Probleme, welche behoben werden mussten.

So gab es zum Beispiel ein Problem beim Import von Messdaten in einem gewissen Fall. In Zusammenarbeit mit dem betroffenen Nutzer konnte das Problem dann in ungefähr zwei Stunden gelöst und am gleichen Tag auch deployt werden.

Die Berechnung eines Wertes aufgrund der Messdaten war am Anfang auch noch falsch. Mit meinem Betreuer konnte auch dieses Problem angegangen und schnell gelöst werden. Hier zeigte sich, dass trotz verschiedenen Tests während der Entwicklung

noch Fehler einschleichen können, welche dann bei den richtigen Daten sichtbar werden.

Eine gewünschte Verbesserung war auch noch der Export von einzelnen Proben (anstelle von einer gewissen Anzahl von Proben). Da der Mechanismus für den Export bereits vorhanden war, konnte auch dies noch am selben Tag freigeschaltet werden.

Die Fehler welche hier gefunden worden waren, konnte alle ausgebessert werden.

2.4 Abschluss

Der Abschluss zeigte das eigentliche Problem an exakten Planung. Zum Ende gab es einige Aufgaben, welche nicht in einer Planung vorgesehen werden können.

Aufgaben wie die Erstellung einer Dokumentation oder die Einführung in das neue System fielen weg, da diese Aufgaben bereits während der Entwicklung immer wieder parallel dazu gemacht wurden.

Das Projekt konnte erfolgreich abgeschlossen werden und ist nun in Verwendung an der Eawag.

2.5 Zusammenfassung

Zusammenfassend kann man sagen, dass die Planung relativ gut geklappt hat und alle Aufgaben erfüllt werden konnten. Allerdings wurde ziemlich früh sichtbar, dass die strikte Separierung der einzelnen Phasen und Aufgaben nicht praktisch oder nützlich war. So gingen viele der Phasen ineinander über. Des Weiteren war die Idee, am Anfang alle Spezifikationen zu definieren und diese dann umzusetzen nicht praktisch.

Abschliessend würde ich in Zukunft auch nicht mehr eine strikte Zeitplanung verteilt auf Wochen zu machen, da dies in Realität nicht umsetzbar ist. Oft arbeitet man abwechselungsweise an verschiedenen Aufgaben, um ein wenig Abwechslung zu behalten. Manchmal wartet man auf Antworten oder die Arbeit von anderen Personen, in diesen Fällen ist es dann sehr praktisch wenn man noch an etwas anderem Arbeiten kann.

Als Alternative würde ich eine hierarchische Liste von Aufgaben mit dem jeweiligen geschätzten Zeitaufwand machen und dann einfach jeweils an den Aufgaben arbeiten, aber nicht zwingend mit fixen Zeitfenstern, welche zu unflexibel für den Arbeitsalltag sind.