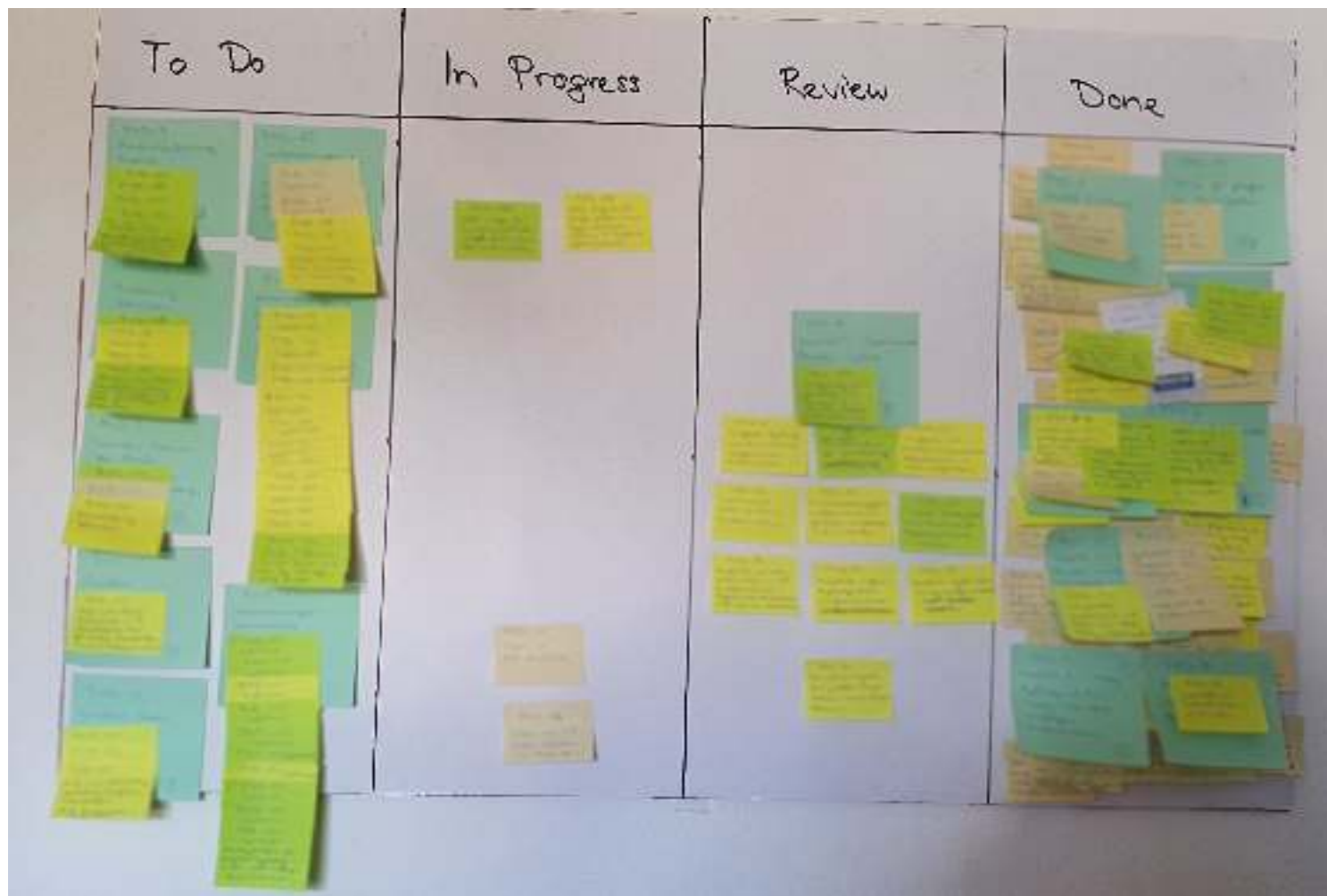


Agile Promotion

Bastian Beskers
Maria Langbecker
Arved Reising
Benjamin Scharf

Einführung

- ▶ Wir hatten am Anfang unserer Arbeit als Software-Entwickler allerlei aha-Momente
- ▶ Es gibt Methoden, die einem helfen Projekte effizienter zu organisieren
- ▶ Es erinnerte uns an Probleme in unseren Doktorarbeiten
- ▶ Wir wollen euch Beispiele vorstellen, was uns geholfen hat oder geholfen hätte



THE FOUR STAGES OF DATA LOSS

DEALING WITH ACCIDENTAL DELETION OF MONTHS OF
HARD-EARNED DATA



www.phdcomics.com

"Piled Higher and Deeper" by Jorge Cham www.phdcomics.com

Versionskontrolle, Paralleles Arbeiten und Backup

- ▶ Dieser Vortrag wurde mit **git** erstellt
 - ▶ für den Anfang reicht auch Dropbox, Google Drive, Nextcloud, ...

Typische Fragen

- ▶ Was passiert, wenn meine Festplatte stirbt?
- ▶ Kann ich nochmal zurück zu meiner Version von gestern?
- ▶ Wieso hat Peter jetzt auch was an der Datei verändert?



JORGE CHAM © 2007



WWW.PHDCOMICS.COM

"Piled Higher and Deeper" by Jorge Cham www.phdcomics.com

How long your Prof.
thinks it should take
to do something



How long it'll
actually take you
to do it



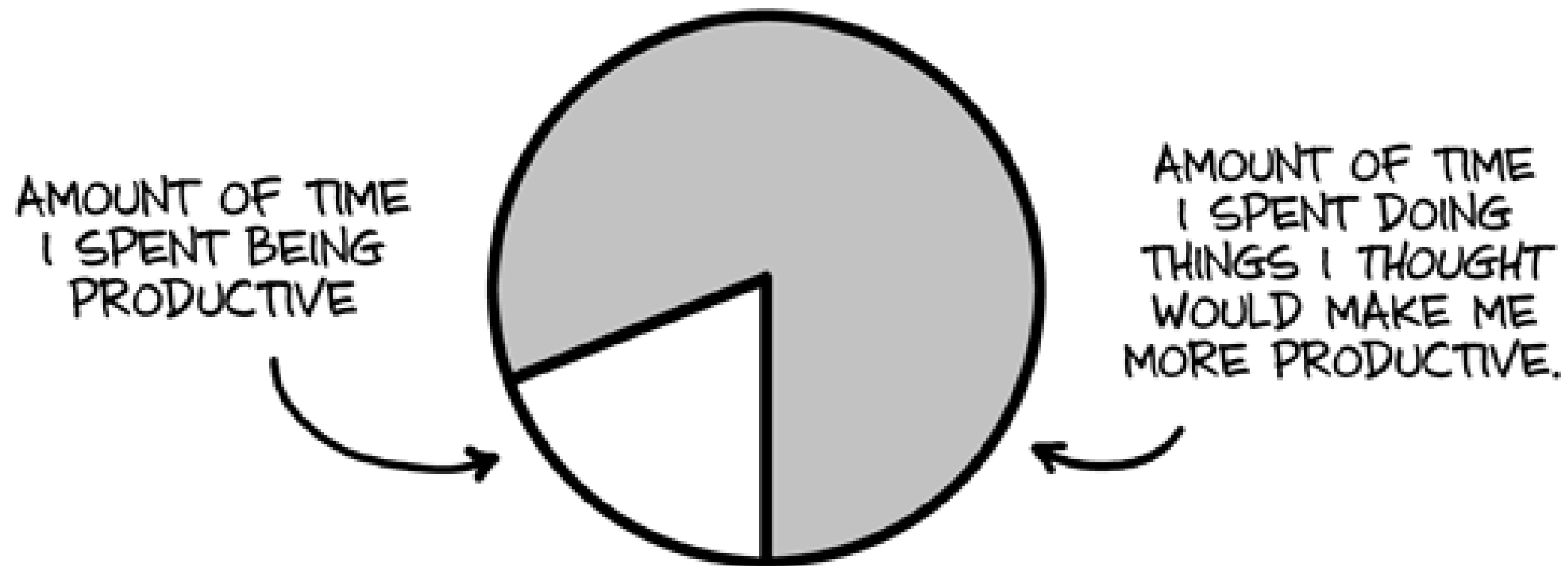
"Trivial"	=	There goes your week.
"Easy enough"	=	Pull your hair out for a month.
"About a week"	=	Actually, this is pretty easy. He/she doesn't know there's technology that will do this for you now. Take the week off!
"Should keep you occupied for the rest of the term"	=	He/she will forget they asked you to do this by the end of the term. Don't even bother.
"This might make a good thesis topic"	=	Say hello to your thesis topic.
"Hmmm..."	=	Uh oh.

JORGE CHAM © 2008

WWW.PHDCOMICS.COM

"Piled Higher and Deeper" by Jorge Cham www.phdcomics.com

HOW MY WEEK WENT:



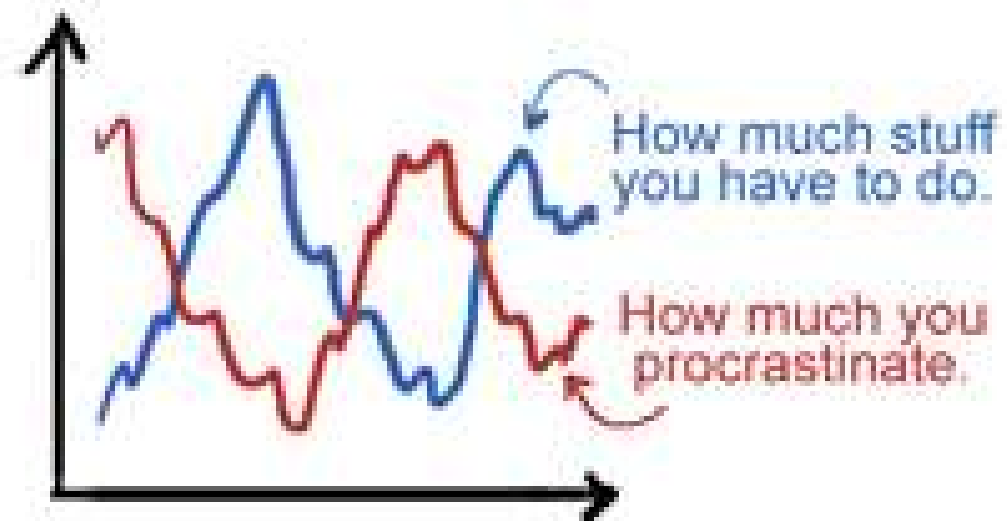
JORGE CHAM © 2014

WWW.PHDCOMICS.COM

"Piled Higher and Deeper" by Jorge Cham www.phdcomics.com

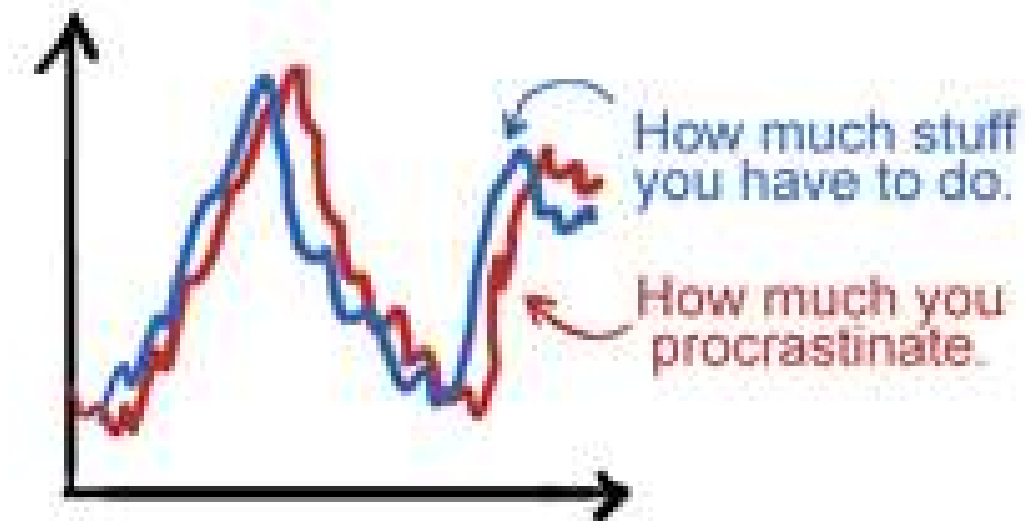
Procrastination

Ideally:



$$\text{Procrastination} \propto \frac{1}{\text{How much stuff you have to do}}$$

In reality:



$$\text{Procrastination} \propto \text{How much stuff you have to do}$$

JORGE CHAM © 2010

WWW.PHDCOMICS.COM

"Piled Higher and Deeper" by Jorge Cham www.phdcomics.com



WWW.PHDCOMICS.COM

"Piled Higher and Deeper" by Jorge Cham www.phdcomics.com

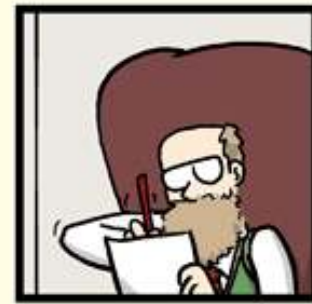
"FINAL".doc



FINAL.doc!



FINAL_rev.2.doc



FINAL_rev.6.COMMENTS.doc



FINAL_rev.8.comments5.
CORRECTIONS.doc



FINAL_rev.18.comments7.
corrections9.MORE.30.doc



FINAL_rev.22.comments49.
corrections.10. #@\$%WHYDID
ICOMETOGRADSCHOOL?????.doc

JORGE CHAM © 2012

WWW.PHDCOMICS.COM

"Piled Higher and Deeper" by Jorge Cham www.phdcomics.com

Erkenntnis #1

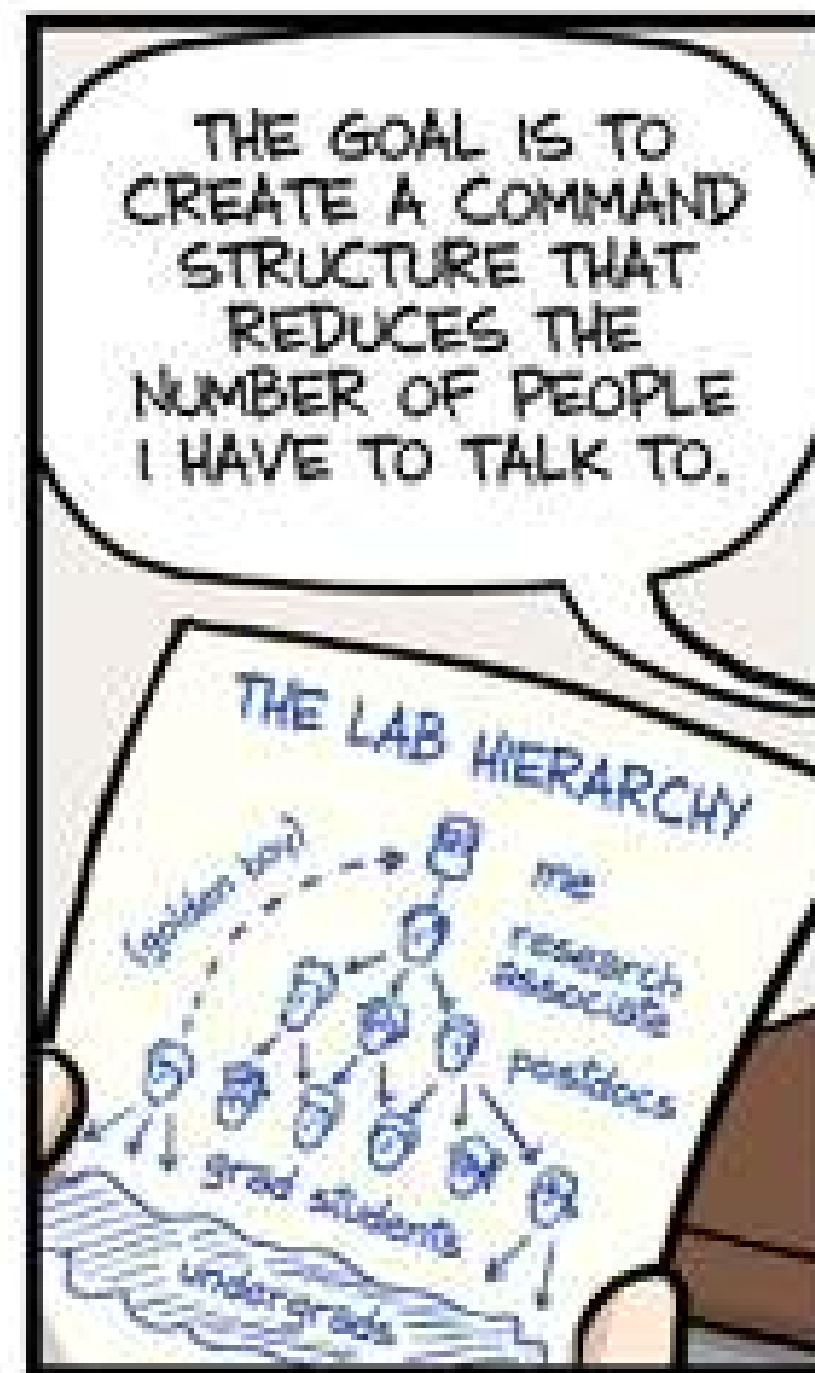
*Ehrlich gesagt, die Probleme habt nicht nur ihr,
sondern alle Unternehmen, die komplexe Probleme (= nicht das gleiche jeden Tag) lösen.*

<http://www.produktmanager-blog.de/wp-content/uploads/2013/01/Produktmanagement-Cartoon.png>

<https://xkcd.com/1425/>

Historische Einbettung - Management

- ▶ Chaos
 - ▶ einfach mal los legen/machen
- ▶ Taylorismus (Management 1.0) - ab etwa 1900
 - ▶ Fließband-Arbeit a la Henry Ford
 - ▶ Command and Control
 - ▶ schlaue Führungskräfte, dumme Ausführende
- ▶ Management 2.0 - ab etwa 1950/60, starke Verbreitung 1990
 - ▶ "People are the most valuable assets"
 - ▶ Servant Leaders
 - ▶ 1on1s
 - ▶ Metrics
- ▶ Management 3.0
 - ▶ Manage the System, not the People
 - ▶ Focus on Communication, Culture, Blockers
 - ▶ als Konsequenz: Autonomie und Alignment



JORGE CHAM © 2009



WWW.PHDCOMICS.COM

"Piled Higher and Deeper" by Jorge Cham www.phdcomics.com



JORGE CHAM © 2010



WWW.PHDCOMICS.COM

"Piled Higher and Deeper" by Jorge Cham www.phdcomics.com

<https://medium.com/@richardlennox/balancing-autonomy-and-alignment-with-accountability-3a0dabe98a95>

Erkenntnis #2

Moderne Software-Entwicklung und die Führung eines Lehrstuhls funktionieren am besten unter Management 3.0.

Werden heute noch oft als Management 2.0 (nicht gut) oder 1.0 (noch schlimmer) geführt.

Darauf habt ihr leider nur bedingt Einfluss, aber ...

- ▶ www.torak.com
- ▶ <https://www.slideshare.net/dimka5/introducing-agile-scrum-xp-and-kanban>

Wissenschaftliche Arbeiten schreiben mit Agilen Methoden

- ▶ Agile Methoden aus der Softwareentwicklung kann man auch für wissenschaftliche Arbeiten nutzen, z.B. zum Schreiben einer Doktorarbeit

Unterteilung der Arbeit in überschaubare Teilaufgaben

Unlösbare Aufgabe:

Ich schreibe eine Doktorarbeit.

Besser:

Ich schreibe jetzt Kapitel 1, das den experimentellen Aufbau zum Atomtransport vorstellt.

- ▶ In der agilen Softwareentwicklung nennt man diese Einheit eine *Story*
- ▶ Vor Beginn des Schreibprozesses sollten alle Stories grob festgelegt werden
 - ▶ Stories in naher Zukunft genauer als die, die am Ende kommen
 - ▶ Die Ausarbeitung der Stories wird kontinuierlich verfeinert
 - ▶ Das kann am Anfang ruhig eine Weile dauern, man spart diese Zeit am Ende ein!

Unterteilung der Arbeit in überschaubare Teilaufgaben

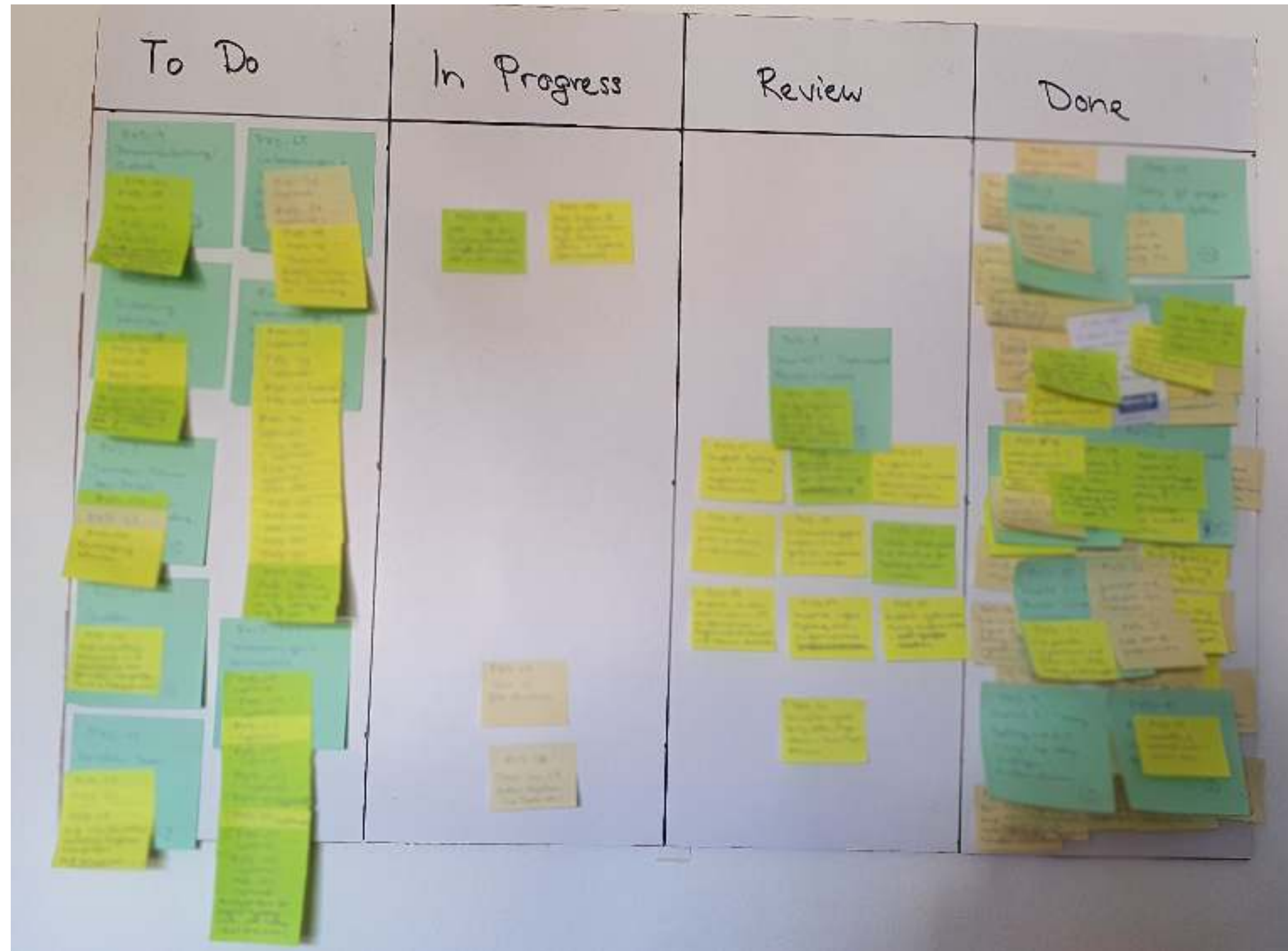
- ▶ innerhalb einer Story gibt es einzelne *Tasks*, z.B.



- ▶ Die Tasks sollten so klein geschnitten und genau genug beschrieben sein, dass man direkt mit ihnen anfangen kann
- ▶ Die Ausarbeitung der Tasks geschieht, bevor man mit einer Story anfängt
- ▶ Man weiß also VORHER schon, was man alles innerhalb einer Story machen möchte

Visualisierung des Arbeitsprozesses

- ▶ Nutzung eines Scrum bzw. Kanban Boards
- ▶ fest vorgegebener Arbeitsablauf



"Work in Progress"-Limit

Uni:

Schreib erstmal überall irgendetwas hin. Überarbeiten ist immer viel leichter als Neuschreiben.

Kanban:

Mach Dinge fertig.

- ▶ Es sind immer nur genau so viele Tasks *In Progress* erlaubt wie das Limit erlaubt
- ▶ für eine Person sinnvollerweise nicht deutlich deutlich mehr als ein Task
- ▶ der nächste Task wird erst angefangen, wenn der erste fertig fürs Review ist
- ▶ vermeidet hunderte lose Enden
- ▶ auch schwierige Aufgaben werden durchgezogen (idealerweise setzt man sich dafür ein Zeitlimit)

Definition of Done

Wann ist ein Task oder eine Story fertig?

Der bearbeitete Abschnitt ist "druckfertig".

Kriterien können z.B. sein:

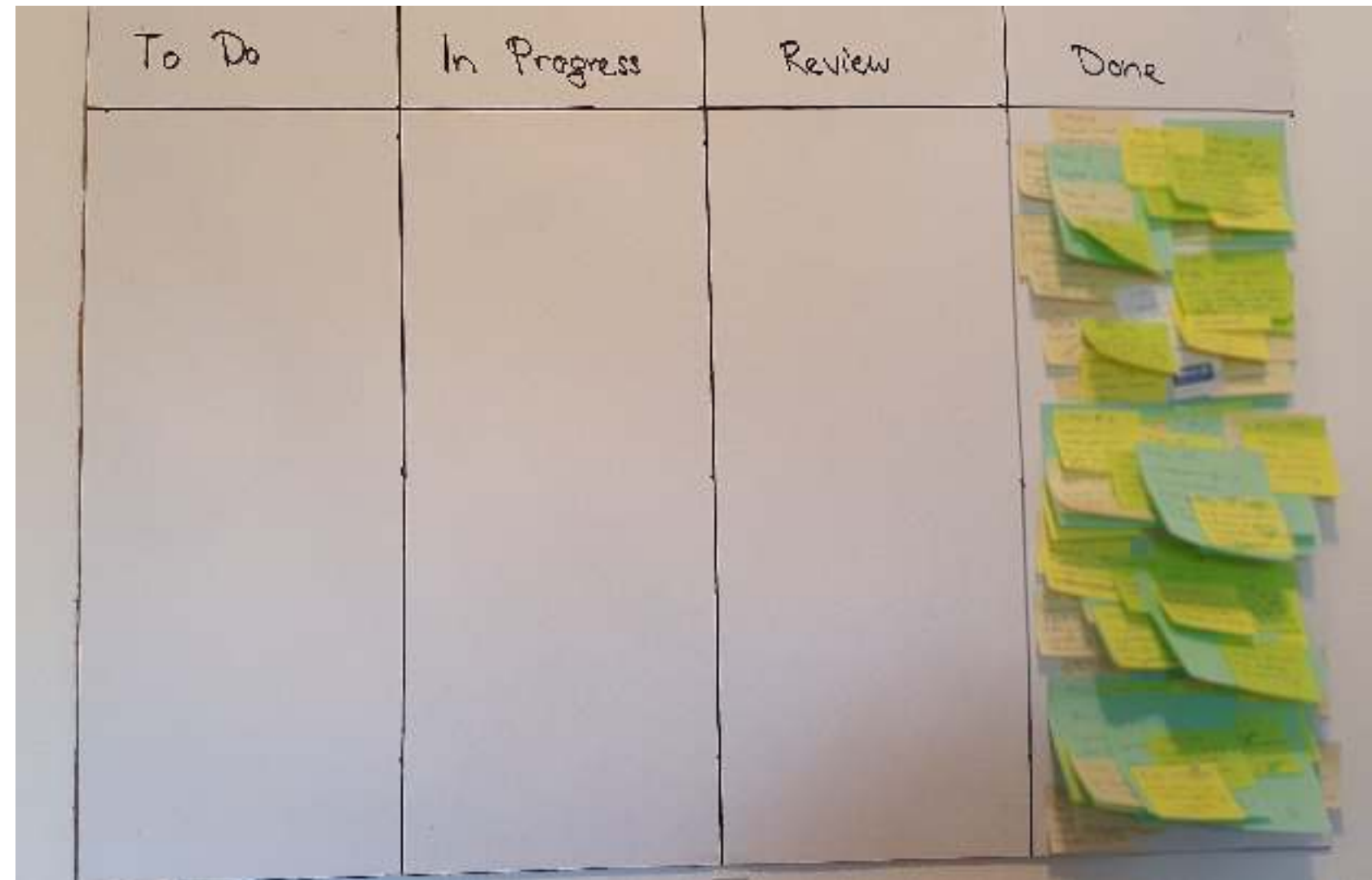
- ▶ für einen Text:
 - ▶ Alle Sätze sind ausformuliert
 - ▶ Es gibt eine Einleitung und einen Abschluss für jeden Gedankengang
 - ▶ Alle Quellenangaben sind vorhanden
 - ▶ Der Betaleser (Freund/Freundin, Mitdoktorand/in, Familie, ...) hat den Text Korrektur gelesen
- ▶ für eine Grafik:
 - ▶ Achsenbeschriftungen sind vorhanden
 - ▶ Alle Datenpunkte oder Fitkurven sind klar erkennbar
 - ▶ Eine Legende ist vorhanden, wenn gewünscht
 - ▶ Die Grafik hat eine Beschriftung, die sie hinreichend gut erklärt

Überblick über den Arbeitsaufwand

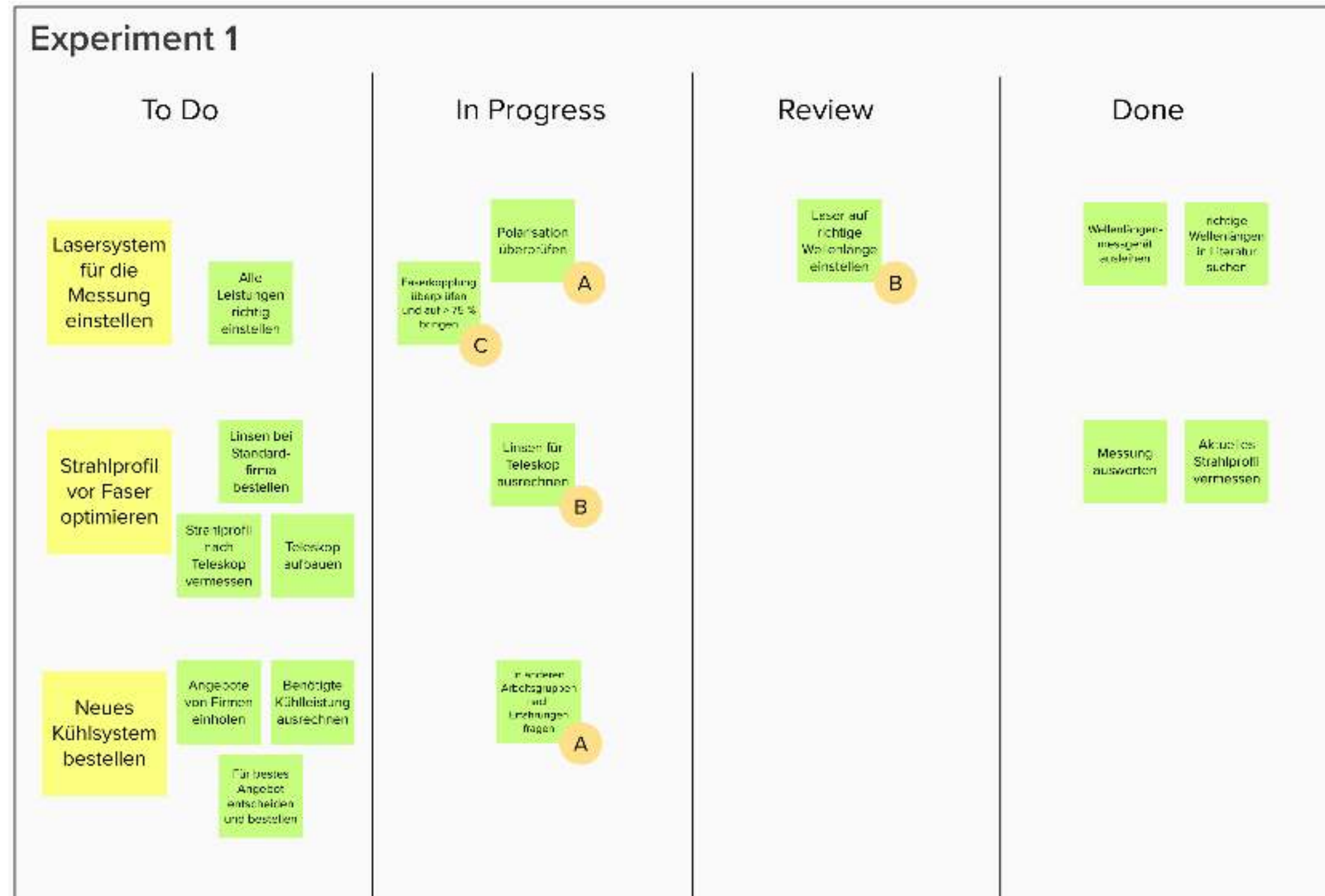
- ▶ Schätzen des Aufwands der Stories
 - ▶ so früh wie möglich für Gesamtüberblick
 - ▶ Berücksichtige sowohl Komplexität als auch Zeitaufwand
 - ▶ Gesamtpunktzahl sollte zum Zeitrahmen der Arbeit passen ;-)
- ▶ Bei fester Deadline: Priorisieren!
 - ▶ Kontinuierlich die Priorität der Tasks evaluieren
 - ▶ Tasks mit niedriger Priorität in eine optionale Story verschieben
 - ▶ notwendige Tasks vor den optionalen erledigen
- ▶ Wenn zwischendurch neue Aufgaben anfallen:
 - ▶ neue Story schreiben, schätzen und priorisieren
 - ▶ oder neuen Task zu existierender Story hinzufügen

Fazit

- Agile Methoden aus der Softwareentwicklung lassen sich sehr gut auf das Schreiben einer Doktorarbeit anwenden!



Agile Methoden im Team



Retrospektive

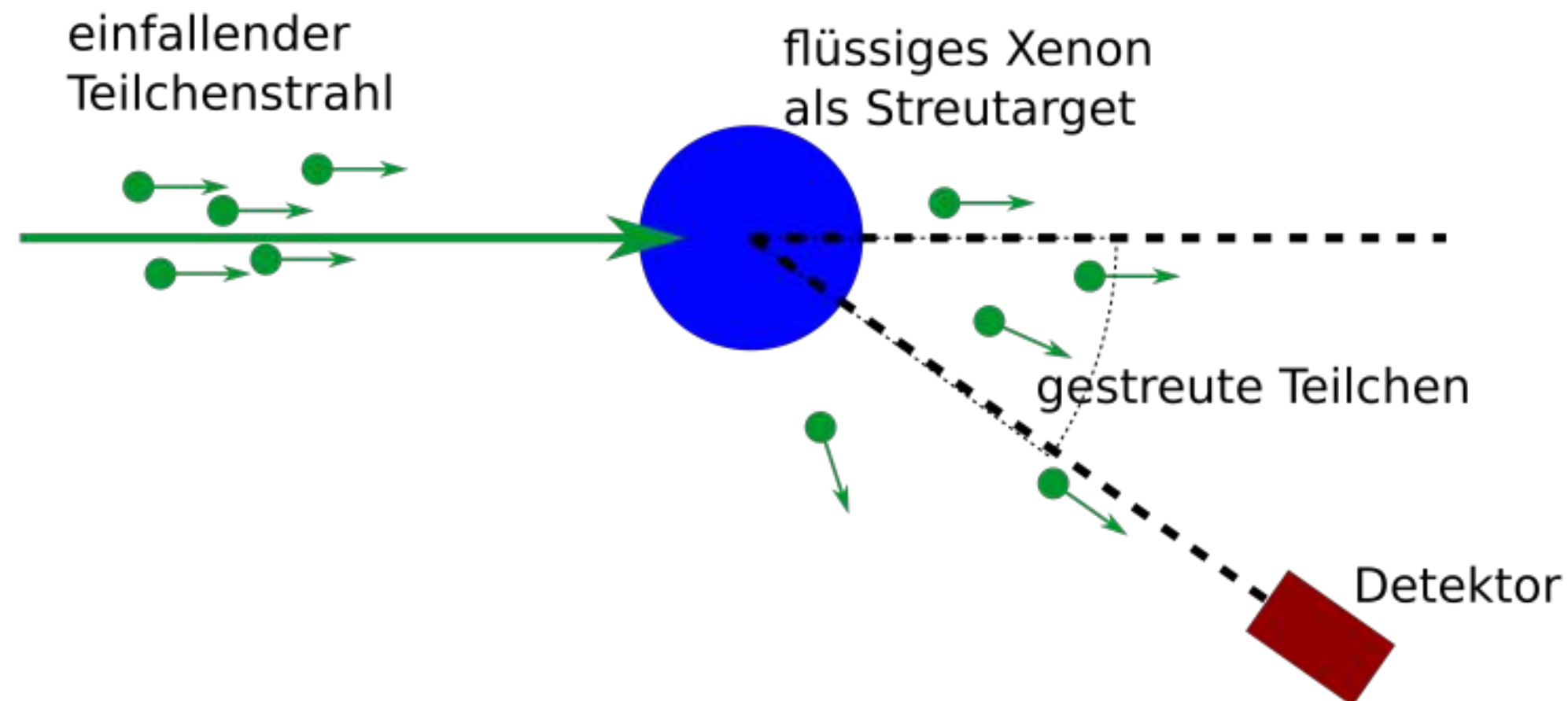
- ▶ Inspect and Adapt
- ▶ Wichtigstes Meeting in der Agilen Softwareentwicklung
- ▶ Softwareentwicklung: fester Turnus / am Ende des Sprints (alle 2 Wochen)
- ▶ Wissenschaft: fester Turnus und/oder nach bestimmtem "Ereignis"
- ▶ Idee im Rückblick: z.B.: nach einer "Mess-Session" mit dem Experiment



Experiment

Streuexperiment

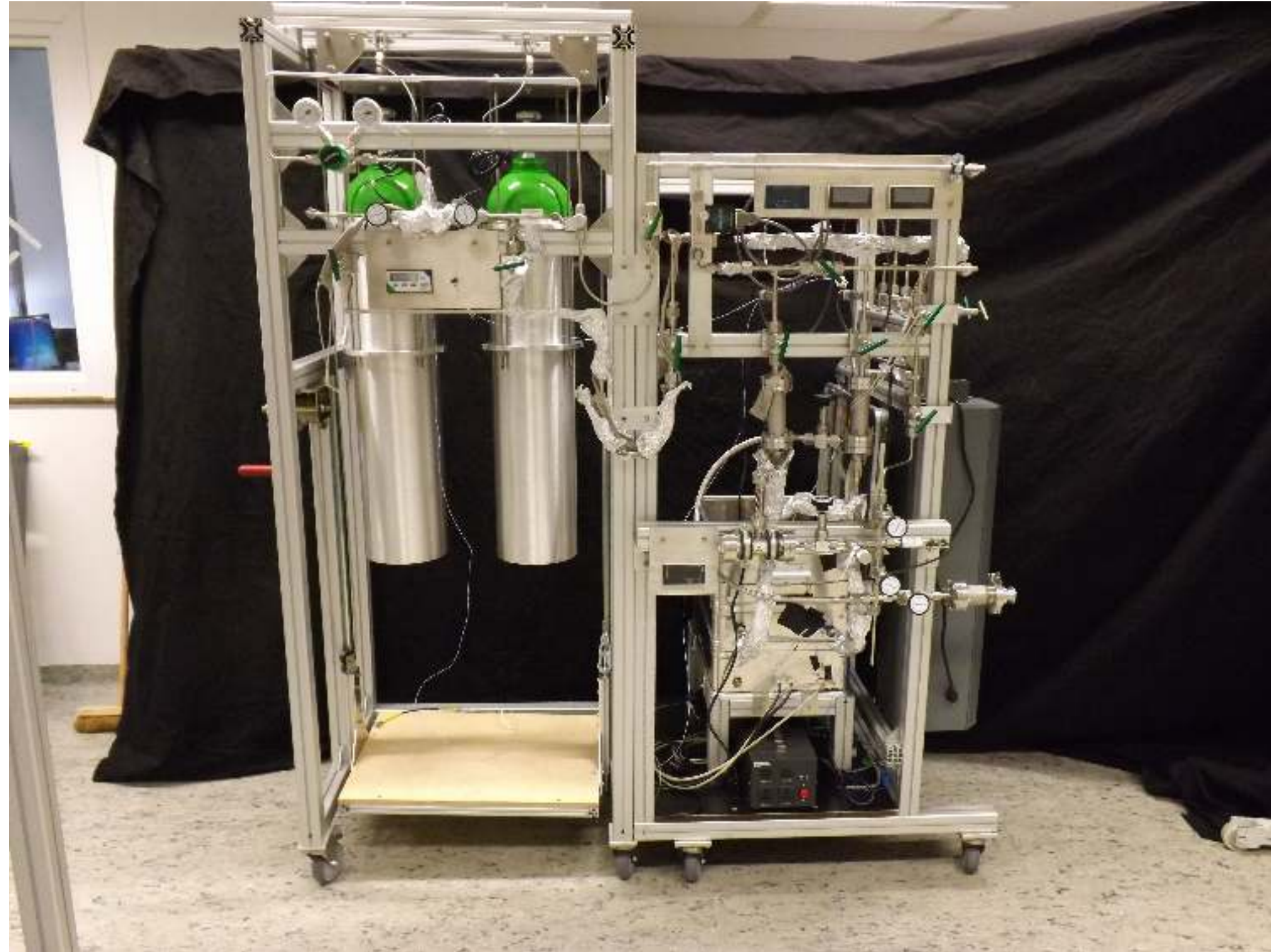
- ▶ Streutarget ist selbst Detektor mit mehreren Lichtsensoren
- ▶ Sekundärdetektor zum Messen des gestreuten Teilchens
- ▶ Gas-Kühl-System mit diversen Systemparametern zum Betrieb



Experiment




Experiment



Ablauf Mess-Session

Mo	Di	Mi	Do	Fr	Sa	So
		1	2	3	4	5
6	7 <i>Zusammenbauen</i> <i>Alice & Bob</i>	8 <i>Zusammenbauen</i> <i>Bob & Chris</i>	9 <i>Abpumpen</i> <i>Alice</i>	10 <i>Abpumpen</i>	11 <i>Abpumpen</i> <i>ca. 18:00, Spülen Bob</i>	12 <i>Spülen</i>
13 <i>Rühten Alice</i> <i>ca. 14:00 Befüllen Alice</i> <i>Messen Chris</i>	14 <i>Messen</i> <i>6:00 Alice</i> <i>18:00 Chris</i>	15 <i>Messen</i> <i>6:00 Bob</i> <i>18:00 Chris</i>	16 <i>Messen</i> <i>6:00 Alice</i> <i>18:00 Bob</i>	17 <i>Messen</i> <i>ab 13:00 Leeren</i> <i>7:00 Chris</i>	18	19
20	21 <i>13-14 Uhr: Retro</i>	22	23	24	25	26
27	28	29	30			

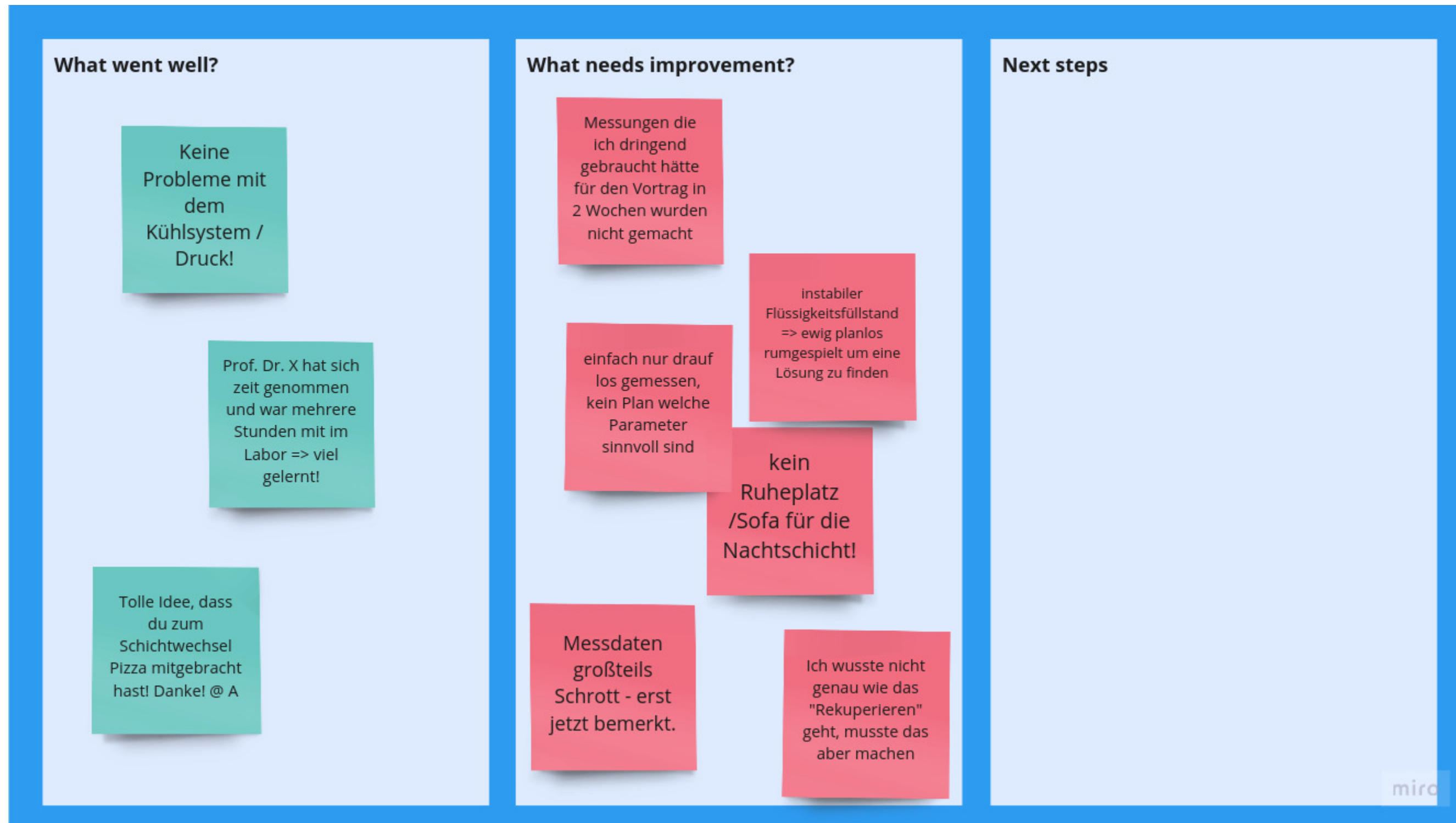
Retro Whiteboard

What went well?	What needs improvement?	Next steps
		

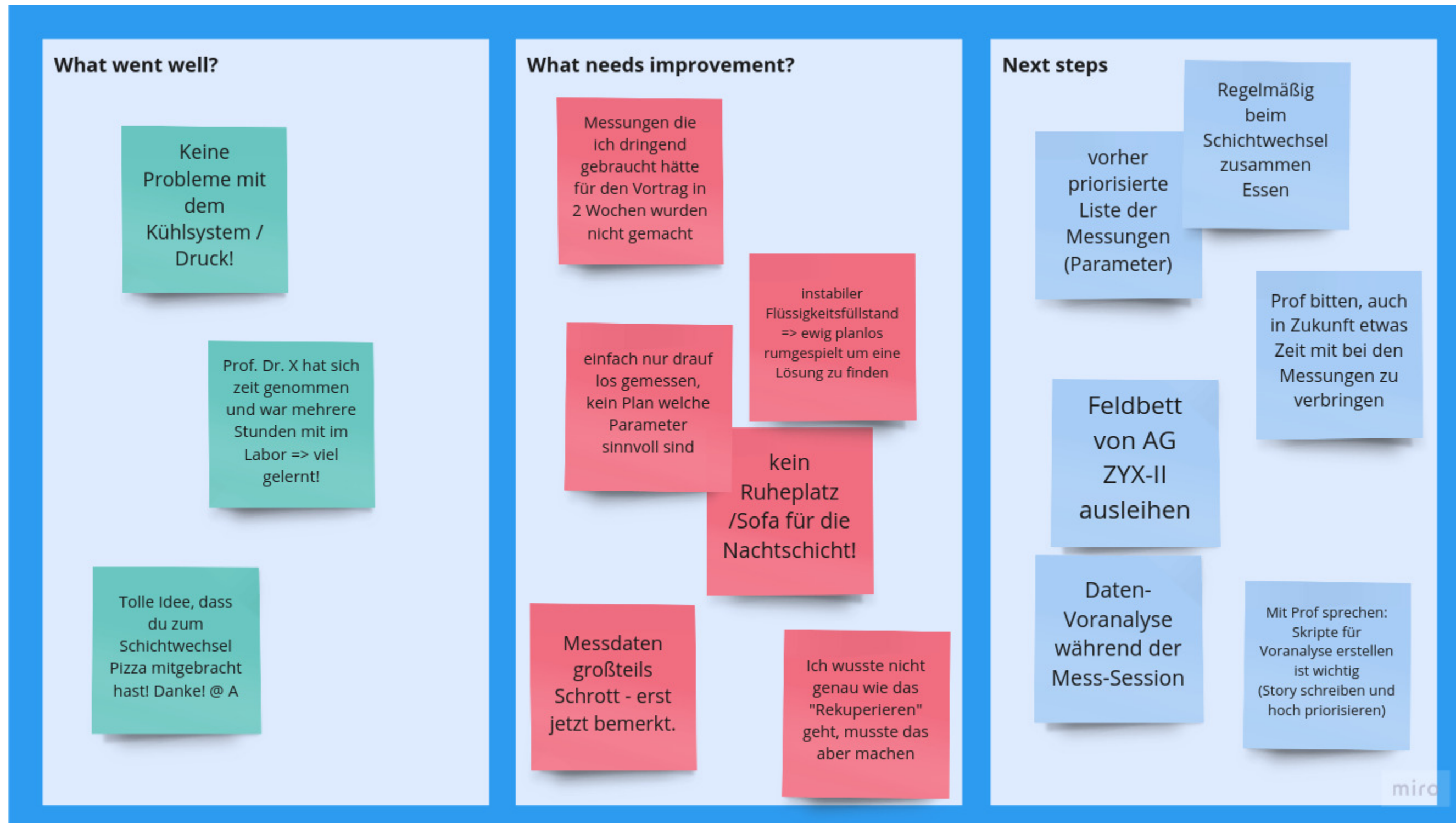
Gutes/Schlechtes sammeln



Ans Board damit



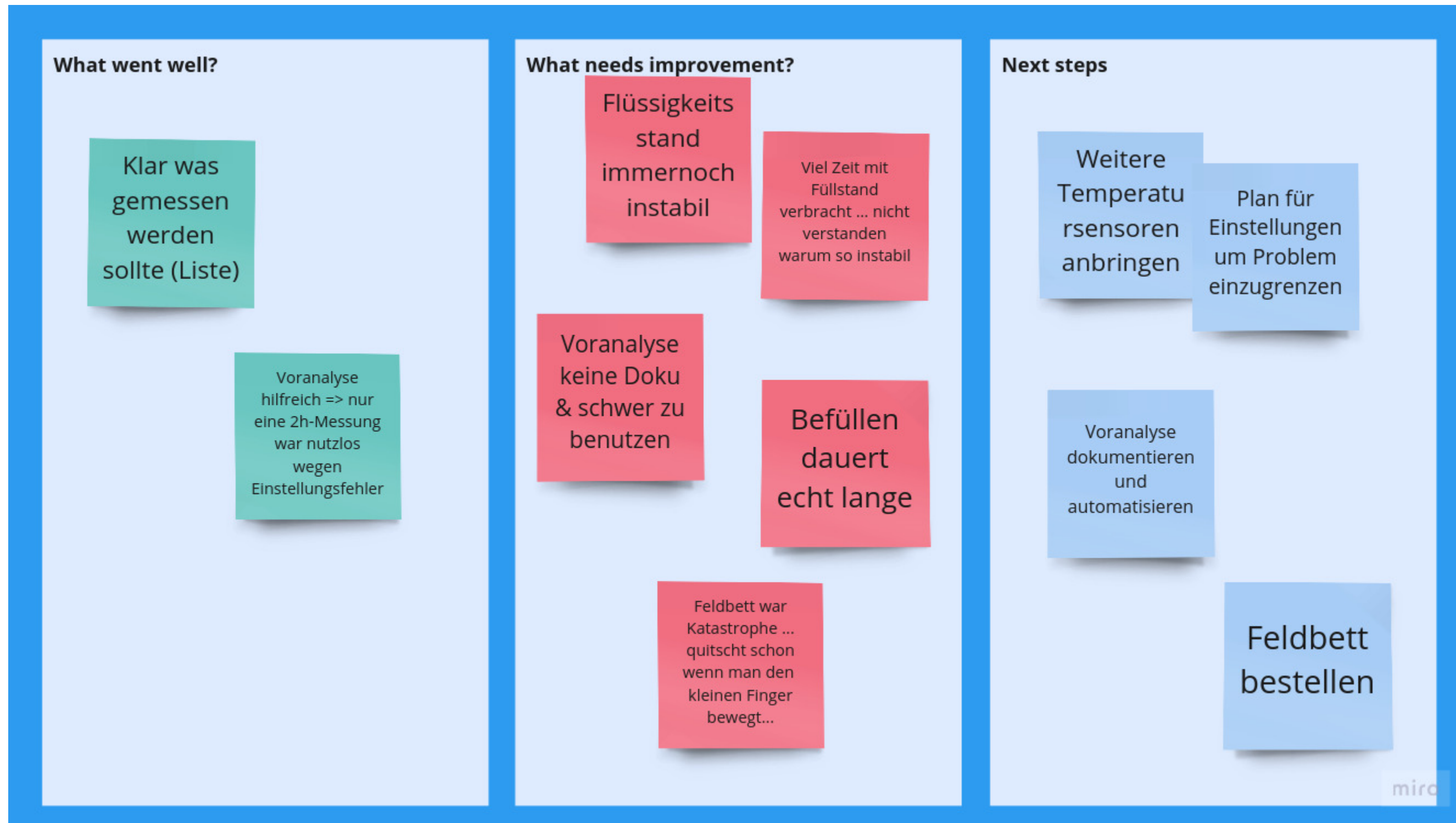
Diskussion und Actions



Planung nächste Session

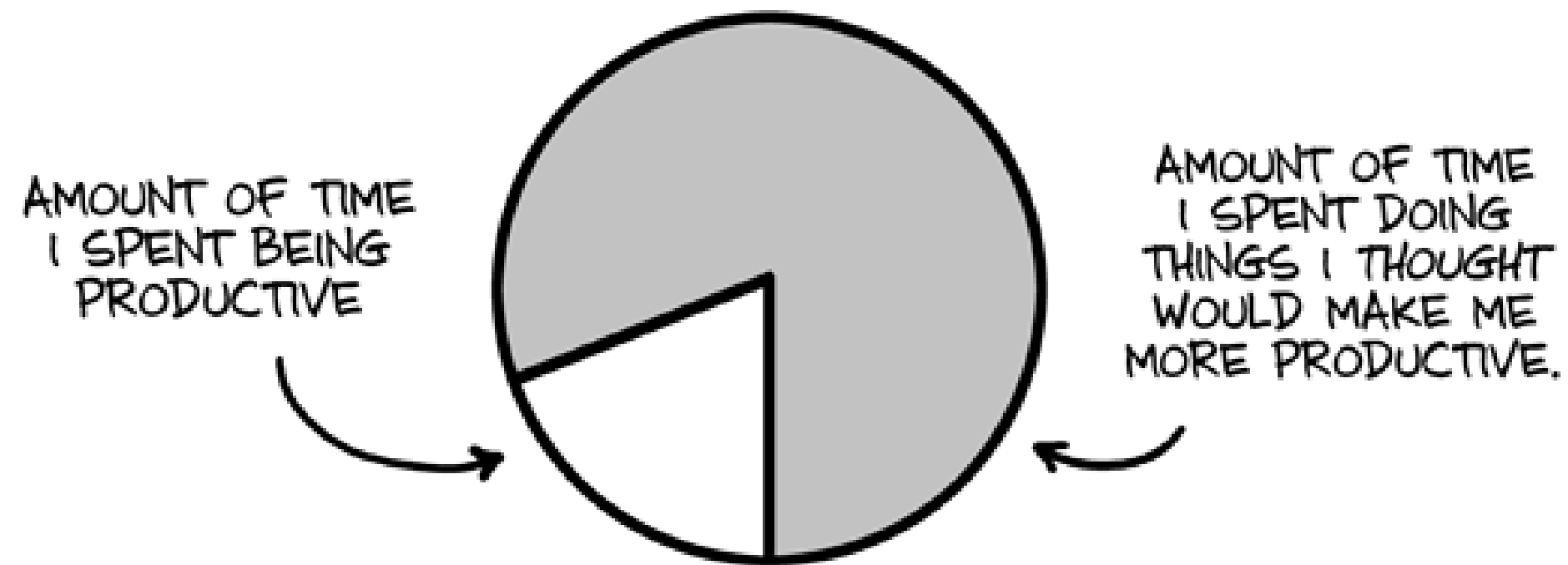
Mo	Di	Mi	Do	Fr	Sa	So
		1	2	3	4	5
			Vorbereitung			
6	7	8	9	10	11	12
Vorbereitung		Messen	Messen	leeren		
13	14	15	16	17	18	19
13-14 Uhr Retro						
20	21	22	23	24	25	26
27	28	29	30			

Retro nach der nächsten Session



Retro - wirklich produktiv werden

HOW MY WEEK WENT:



JORGE CHAM © 2014

WWW.PHDCOMICS.COM

"Piled Higher and Deeper" by Jorge Cham www.phdcomics.com

Zusammenfassung

- ▶ Andere Idee für mehr Individuelles Arbeiten
 - ▶ Retro für sich selbst in festem Turnus
 - ▶ Was lief gut? Was nicht so gut?
 - ▶ Probleme mit einem Partner (Freund/Freundin, anderer Doktorand) diskutieren um Verbesserungsideen abzuleiten
- ▶ Generell: Agile Methoden im wissenschaftlichen Kontext unserer Meinung nach sinnvoll
 - ▶ Vermutliche keine "Scrum-by-the-book" Umsetzung möglich => kreativ sein, "Inspect and Adapt"
 - ▶ Uns interessiert auch, wie das konkret aussehen kann (und ob es funktioniert) => wir unterstützen euch gerne

Literatur

- ▶ Boris Gloger – Scrum: Produkte zuverlässig und schnell entwickeln
- ▶ <https://www.scrumguides.org>
- ▶ David J. Anderson – Kanban: Successful Evolutionary Change for Your Technology Business
- ▶ Gerold Patzak, Günter Rattay – Projektmanagement: Leitfaden zum Management von Projekten, Projektportfolios und projektorientierten Unternehmen
- ▶ Jurgen Appelo – Management 3.0: Leading Agile Developers, Developing Agile Leaders
- ▶ Team-Versionskontrolle: <https://github.com>

FAQ - Fragen von früheren Terminen

Welche Präsentationstools gibt es für kollaboratives Arbeiten und Versionierung mit git

Prinzipiell lässt sich alles mit git versionieren, besser als binäre Dateien lassen sich alle textbasierten Datenformate damit verwalten, weil diff(ferenzier)bar

Präsentationstools

- ▶ reveal.js - diese Präsentation und Quasi-Standard bei TNG, verwendet JavaScript, CSS, html, Markdown ... alles textbasiert
- ▶ LaTeX - Präsentationen und Abschlussarbeiten lassen sich ebenfalls hervorragend mit git versionieren

Rückmeldungen aus dem Publikum

- ▶ PowerPoint-Präsentation lassen sich gut mit MTeams gemeinsam verwalten
- ▶ overleaf.com bzw. <https://github.com/overleaf/overleaf>

Welche Tools sind für digitale Boards und Projektplanung empfehlenswert

- ▶ Kommt darauf an ...
- ▶ Benutzen was man hat bzw. kennt. Nicht viel Zeit damit verbringen das ultimative Tool zu finden. Was ihr braucht, merkt ihr während ihr es benutzt.
- ▶ Im Zweifel lieber das einfachere Tool
- ▶ Inspect and Adapt ... wenn das anfangs gewählte Tool wirklich nicht zum Einsatzzweck passt, dann merkt ihr das (z.B. in einer Retrospektive) und kennt dann auch genau die Schmerzpunkte die mit einem alternativen Tool gelöst werden sollen

Aufgabenverwaltung

- ▶ Trello
- ▶ Todoist
- ▶ TaskCoach (OpenSource)
- ▶ notion.so
- ▶ Jira - Konfiguration nötig, häufig in großen IT-Projekten eingesetzt
- ▶ TaskWarrior - Konfiguration nötig
- ▶ Taiga
- ▶ monday.com
- ▶ basecamp.com
- ▶ ora.pm
- ▶ Remember the milk
- ▶ gitlab - hat auch gleich ein verknüpftes Repository, CI/CD und ein Wiki
- ▶ ...

Die Tools sind hier in zufälliger Reihenfolge aufgelistet. Es sind auch Empfehlungen aus dem Publikum und aus dem "TNG-Brain" enthalten, die wir nicht selbst verwendet haben uns jedoch sinnvoll scheinen.

Digitale Whiteboards

- ▶ Miro
- ▶ Mural
- ▶ ExcaliDraw
- ▶ ...

Die Tools sind hier in zufälliger Reihenfolge aufgelistet. Es sind auch Empfehlungen aus dem Publikum und aus dem "TNG-Brain" enthalten, die wir nicht selbst verwendet haben uns jedoch sinnvoll scheinen.

Priorisierung

- ▶ Verschiedene Swimlanes, Farbschemata oder auch eine Sortierung nach Priorität innerhalb einer Spalte
- ▶ Die Eisenhower Matrix
- ▶ Gantt-Charts oder -Diagramme zum Projektmanagement

So long, and thanks for all the fish!

Fragen?

recruiting@tngtech.com

