



Project Bookkeeper

Final Presentation - Team Rot

SSE 2022

Hasso-Plattner-Institut

Wie ist diese Präsentation entstanden?



Project Objective
& Learnings



Development



Development



Kanban & Scrum



Demo



Technikteil



Project Bookkeeper

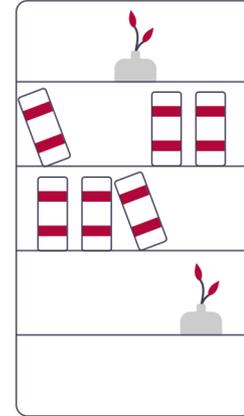
Final Presentation - Team Rot

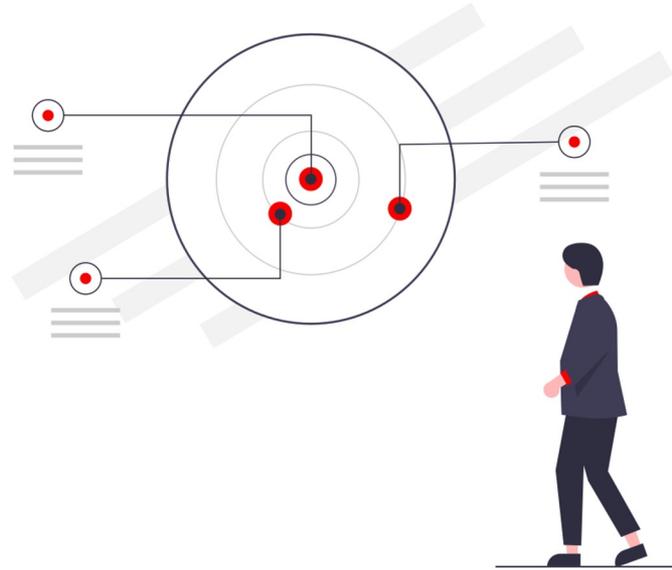
SSE 2022

Hasso-Plattner-Institut

Agenda

1. Project Objective
2. Feature Roadmap
3. Demo
4. Technology Overview
5. Kanban vs Scrum
6. Statistiken & Feedback
7. Lessons Learned





Project Objective

Project Objective

*"Create a **compelling** private library
and asset managing system."*



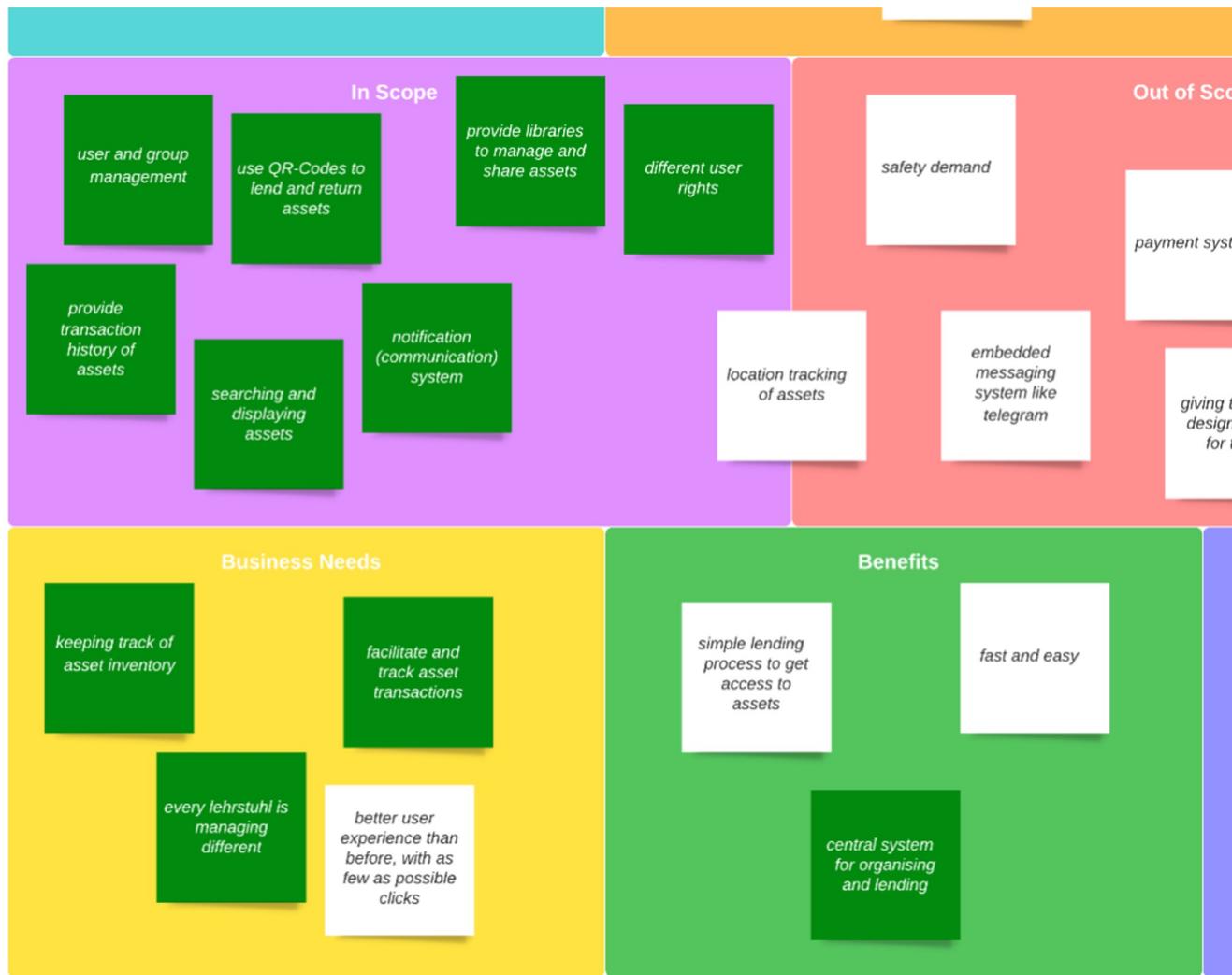
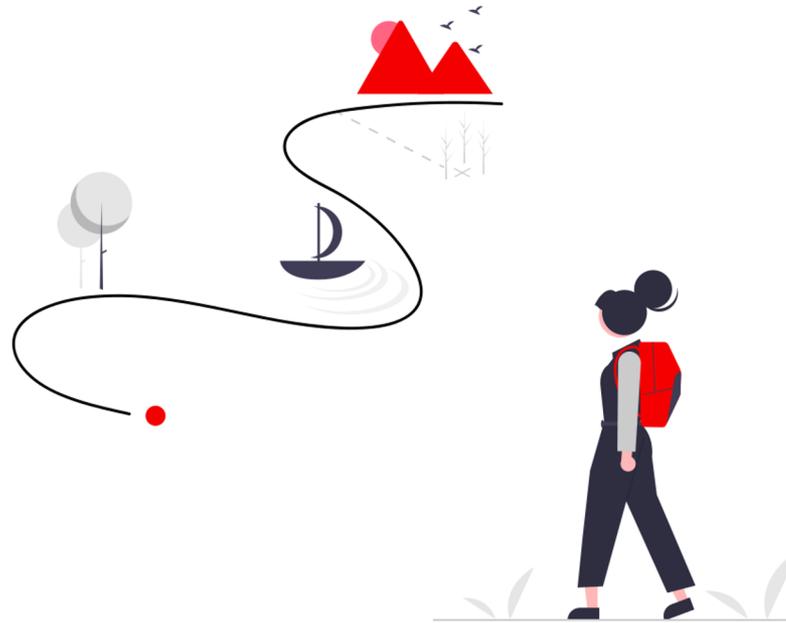


Chart 8



Feature Roadmap



Roadmap

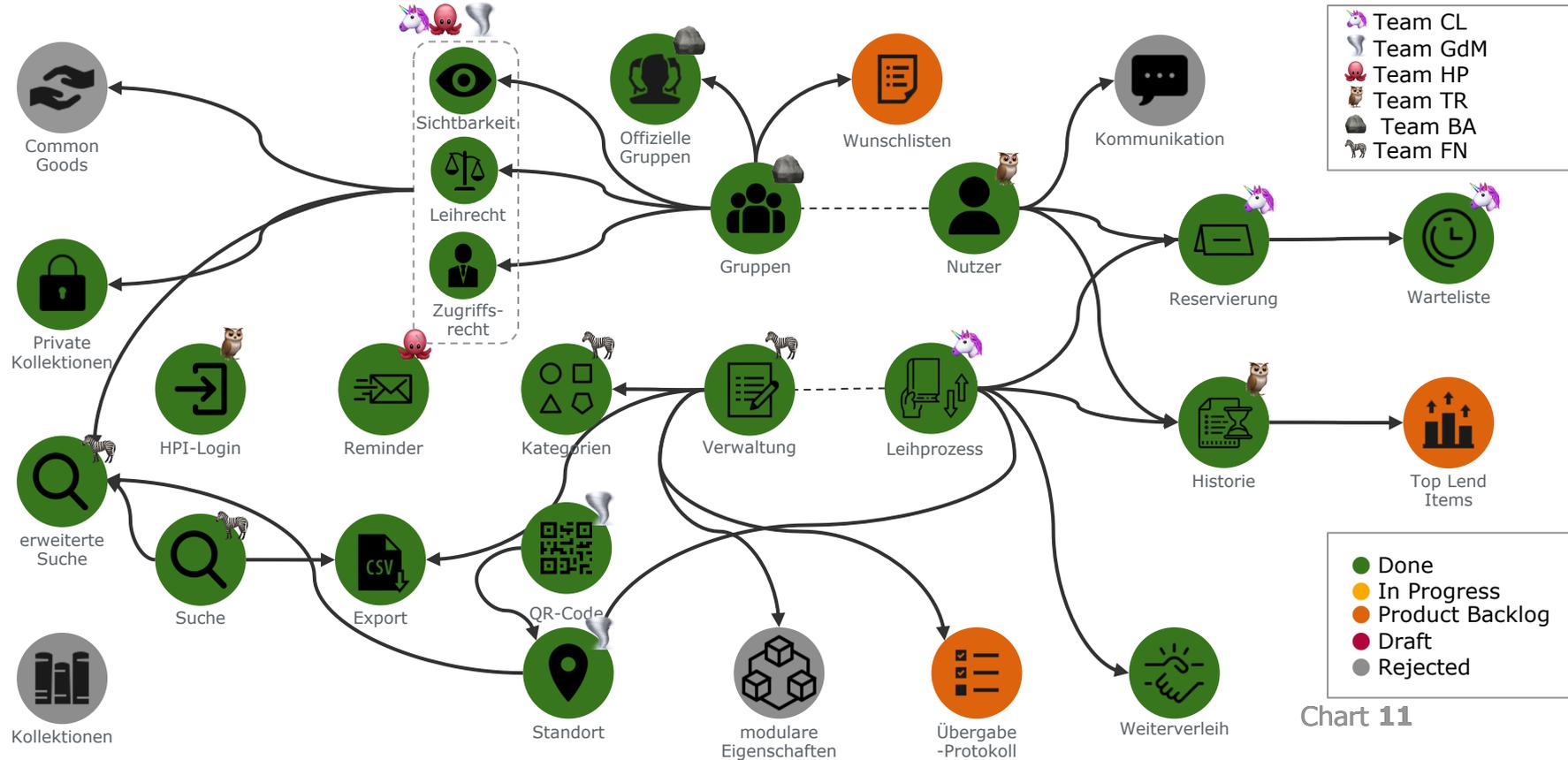
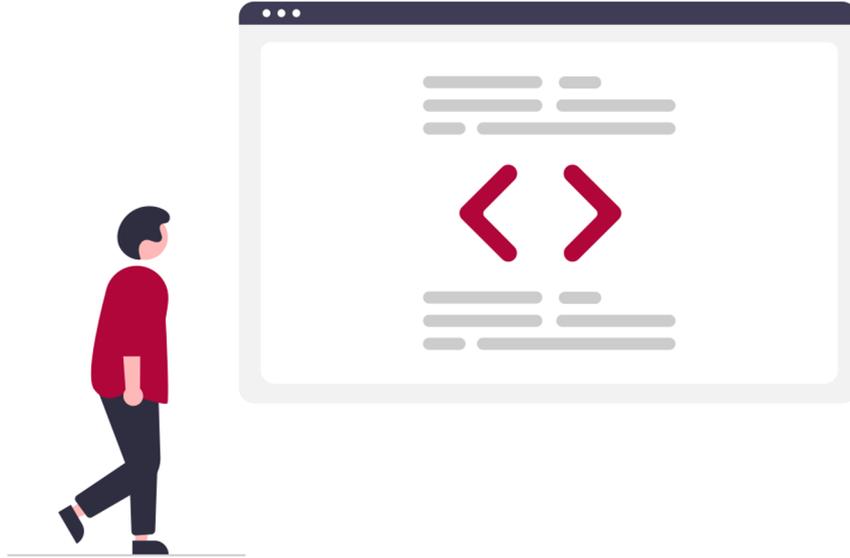


Chart 11

[Demo]



Technology Overview



UI Design & Concepts

UI Design

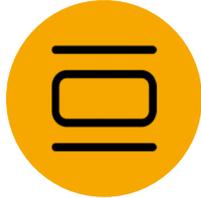
- Bilingualität für alle statischen Inhalte
- Responsive Webdesign (mittels Bootstrap Framework)
- Designorientierung an HPI-Farben
- Verwendung von **View Components**

User Experience

- Click-to-Target Rate: 6
- Responsive Webdesign (mittels Bootstrap Framework)
- QR Codes für verbesserte UX



View Components



View Components bieten wiederverwendbare, testbare und verschachtelbare UI Komponenten.

Vorteile:

- Konsistenz in der UI
- Baustein-Prinzip
- Konsistenz in der Entwicklung

Verwendungsbeispiele

- Item Auflistung für Gruppen
- Gruppenname (Verlinkung und Verifizierungssymbol)
- Notification Alert-Card mit Header und Body
- Item Verfügbarkeits-Badge (inklusive der Zustände)
- Item Ausleih-Buttons (Leihen, Rückgabe, Reservierung, Warteliste)

View Components - Gruppenansicht

```
app\views\groups\index.html.erb
```

```
<% @groups.select{ |g| !g.personal_group? and !g.everyone_group?}.each
do |group| %>
<div class="container text-center mt-5 p-3">
  <div class="h3">
    <%= render GroupNameComponent.new(group: group) %>
    [..]
  </div>
  <div class="position-relative">
    <%= link_to "", group_path(group), class: "stretched-link" %>
    <%= render GroupItemListComponent.new(group: group) %>
  </div>
</div>
<% end %>
```

SSE Fellows

Item	Format	Genre	Language	Age Rating
Shrek - Der tollkühne Held	DVD	Animationsfilm	Deutsch	0
Für immer Shrek	DVD	Animationsfilm	Deutsch	6
Shrek der Dritte	DVD	Animationsfilm	Deutsch	6
Shrek 2 - Der tollkühne H...	DVD	Animationsfilm	Deutsch	0

Squeak Fanclub

Item	Author	Genre	Language
Design Patterns. Elements...	Erich Gamma, ...	Software Design	English
Refactoring: Improving th...	Martin Fowler, ...	Software Engi...	English
Refactoring to Patterns	Joshua Keriev...	Software Engi...	English
Pattern-Oriented Softwar...	Frank Buschm...	Software Desi...	English

Projektqualität



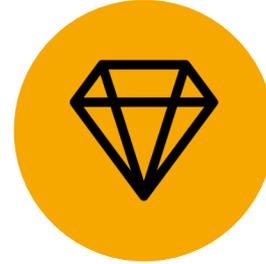
Sicherheit

- Dependency-Bots zum automatischen Ermitteln von notwendigen Security-updates
- Github Security Analyzer für Ruby Code
- HTTPS-Enabled (*über Heroku bereitgestellt*)



Tools

- Codequalitäts-Sicherung mit Rubocop
- **Dev-Deployment** + Main-Deployment über Heroku



Bibliotheken

- QR-Code-Erstellung (*rqrcode, ZXing*)
- OpenID Connect (*omniauth_openid_connect*)
- CSV-File Generierung (*csv*)
- Bootstrap Framework (UI)

Beispiel QR-Code

```
app\views\qr_reader\scan.html.erb
```

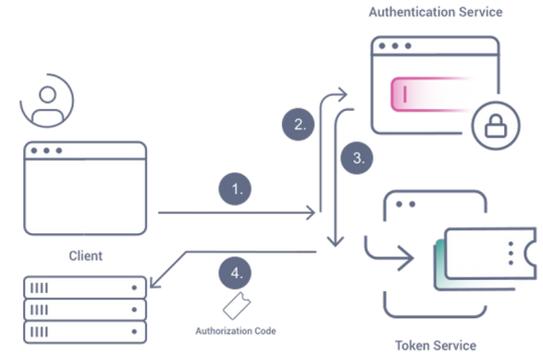
```
function openQrCodeReader() {  
  codeReader = new ZXing.BrowserQRCodeReader()  
  
  codeReader.decodeFromInputVideoDevice(undefined, "video").then((result) => {  
    document.location.href = result.text;  
  }).catch((err) => {  
    console.error(err)  
  })  
}
```

Beispiel Authentifizierung

Komplexer OIDC Flow konnte durch **Omniauth-Library** von Rails in **wenigen Zeilen Code** umgesetzt werden

```
app\controllers\users\omniauth_callbacks_controller.rb
```

```
def openid_connect
  @user = User.from_omniauth(request.env["omniauth.auth"])
  if @user.persisted?
    sign_in_and_redirect @user
    set_flash_message(:notice, :success, kind: "OpenID Connect") if
is_navigational_format?
  else
    failure
  end
end
```



Beispiel Localization

Teamweiter Standard war die Verwendung von **I18n** für die Implementierung von Zweisprachigkeit.

Übersetzung durch Devs; Textvorgaben durch POs waren nur einsprachig

```
app\models\item.rb
```

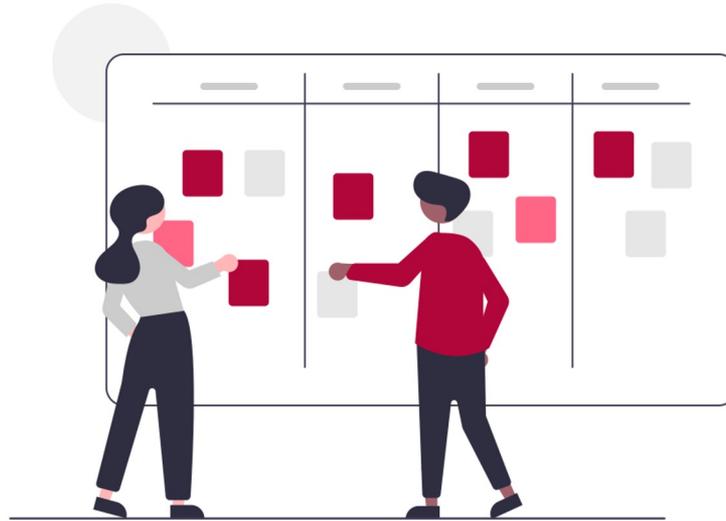
```
def status_text(user)
  return I18n.t("items.status_badge.no_access") unless user.can_borrow?(self)
  return I18n.t("items.status_badge.reserved_by_me") if reserved_by?(user)
  return I18n.t("items.status_badge.available") if borrowable_by?(user)
  return I18n.t("items.status_badge.overdue") if overdue_for?(user)
  return I18n.t("items.status_badge.borrowed_by_me") if borrowed_by?(user)

  I18n.t("items.status_badge.not_available")
end
```

Beispiel Export

```
app\models\csv_export.rb
```

```
def self.to_csv(items)
  csv_string = []
  items.group_by(&:item_type).each do |type, items_of_type|
    csv_data = CSV.generate(headers: true) do |csv|
      add_entries_to_csv(csv, type, items_of_type)
    end
    csv_string << csv_data
  end
  csv_string.join
end
```



Kanban vs. Scrum



Kanban - Our new Best Friend

Wie haben wir es in **Team Rot** umgesetzt?



Global Backlog



Eigenes **Kanban Board** pro Team



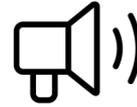
Kanban **Refinement Meetings**

Kanban - What We Noticed in Comparison with Scrum



Motivation

- **Mehr Selbstverantwortung**, weniger Zwang
- Teams sind **flexibler**
- Klare **Deadlines** fehlen



Kommunikation

- Kommunikation ist **schwieriger**
 - Teams sind asynchron und weniger zentral gesteuert
- **Fehlende zentrale Vorstellung** der Issues



Und wat nu?



Statistiken & Feedback



Daten pro Developer

Wie zufrieden sind die Devs mit ihrem **eigenen Zeitmanagement** über die Phasen hinweg gewesen?

bezüglich humaner Uhrzeiten und Arbeitslänge

- + **Zufriedenheit** hat sich zur **Kanban Phase** hin **verbessert**
- zum **3. Sprint** hat sich die **Zufriedenheit** jedoch **verschlechtert**

In der Zwischenstandspräsentation haben wir angekündigt der Unzufriedenheit auf den Grund zu geben. Das waren die Ergebnisse...

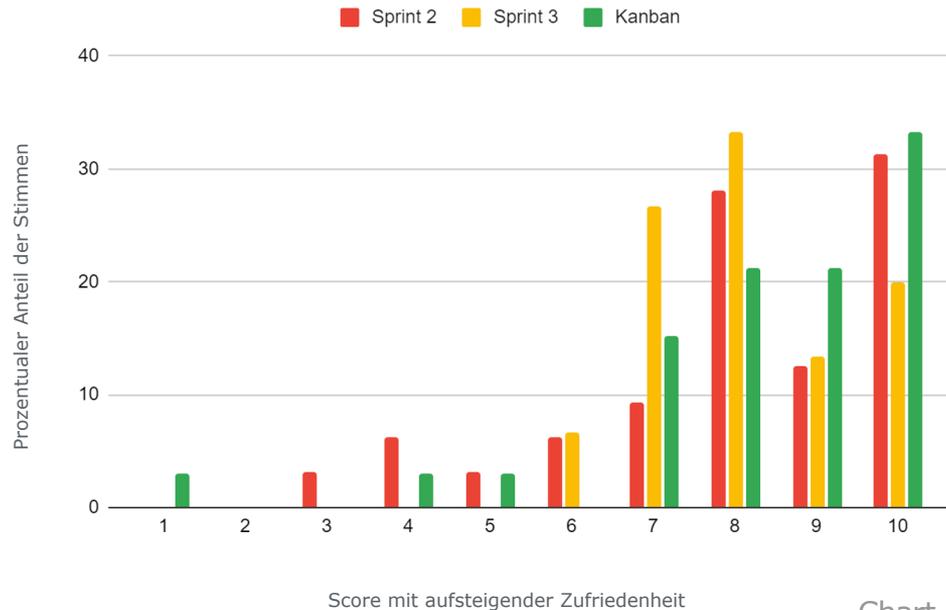


Chart 27

Daten pro Developer

Wie zufrieden sind die Devs mit ihrem **eigenen Zeitmanagement** über die Phasen hinweg gewesen?

bezüglich humaner Uhrzeiten und Arbeitslänge

- + **Zufriedenheit** hat sich zur **Kanban Phase** hin **verbessert**
- zum **3. Sprint** hat sich die **Zufriedenheit** jedoch **verschlechtert**

In der Zwischenstandspräsentation haben wir angekündigt der Unzufriedenheit auf den Grund zu geben. Das waren die Ergebnisse...

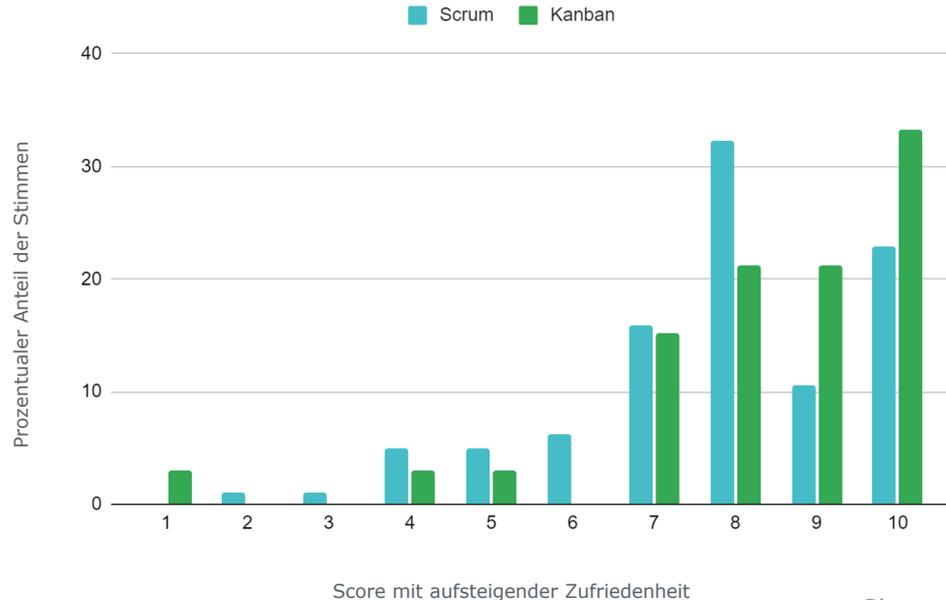


Chart 28

Daten pro Developer

Wir haben das Thema in den Meetings im Team angesprochen und kamen zu folgendem Ergebnis:

- Problem war eher die Arbeitslänge als die Arbeitsuhrzeit
- die Devs haben sich durch die Dependencies zu anderen Teams oft dazu genötigt gefühlt, Tickets trotz Unterschätzung zu beenden
- dabei haben sich die Devs selten aktiv von anderen Teams gedrängt gefühlt, sondern eher eine intrinsische Verantwortung

- + das Thema wurde in den Meetings angesprochen und die Devs wurden motiviert sich in solchen Momenten an den SM zu wenden um eine bessere Lösung zu finden
- + durch den selbstbestimmten Pull hat sich die Überarbeitung in der Kanban Phase enorm verbessert

Rückblick: Verbesserungen seit Interim-Presentation



Rückblick: Verbesserungen

- POs behalten Überblick, erstellen **Tickets** & weisen Devs darauf hin → es gab **keine Unklarheiten bei der Ticketverteilung** mehr
- **Bessere Absprache** der POs
- **Teamübergreifende Reviews?** → es war den **Teams freigestellt, teamübergreifende Reviews** zu machen

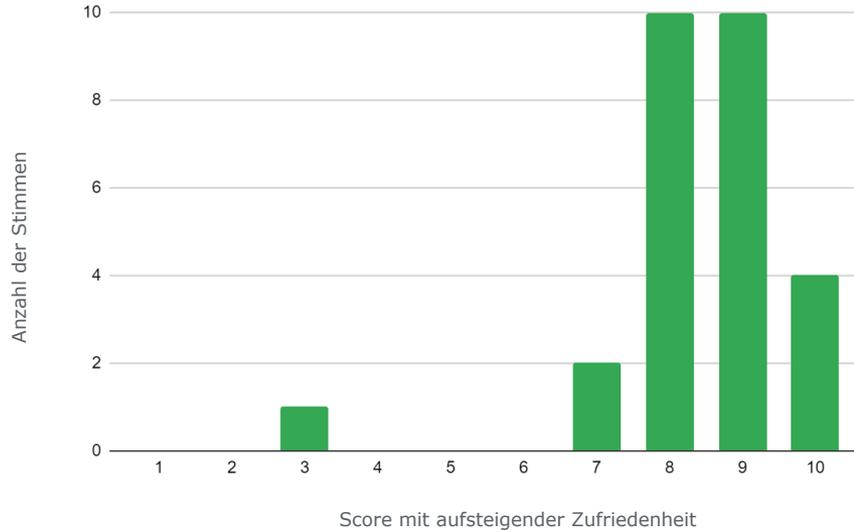
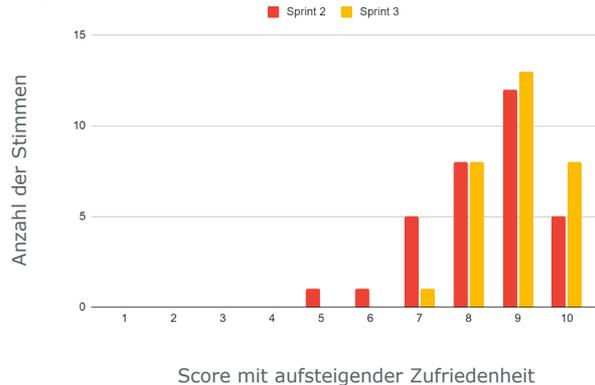
Zufriedenheit Teamarbeit

Wie zufrieden sind die Devs mit der Teamarbeit im eigenen Team während der Kanban Phase gewesen?

+ **bis auf eine Ausnahme ist die Zufriedenheit** in der Kanban Phase hoch geblieben

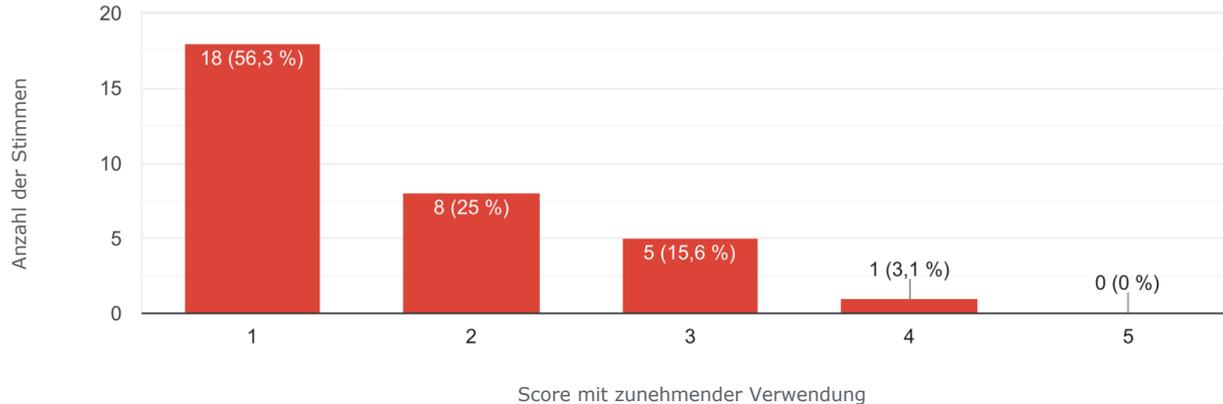
Zum Vergleich hier Sprint 2 und

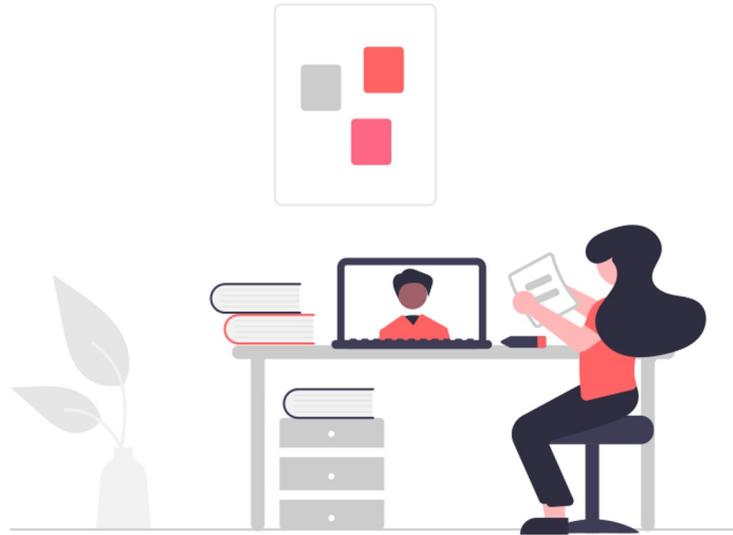
3:



Wie viel hast du **ChatGPT** zum coden bei SSE genutzt?

Von "gar nicht" bis "ich habe gar keinen Code selbst geschrieben"





Lessons Learned



“**Pair programming** lohnt sich” “**Rede öfter** mit dem PO”

“**Kernarbeitszeit** als Team, was einen immer wieder zum Projekt holt”

“Die anderen sind **genauso planlos** wie du, deshalb ist **fragen** nicht schlimm”

“Mit **Schätzen** von Stories erst gar **nicht anfangen**”

“Mehrwert vom Schätzen in Scrum nicht gesehen”

“**Arbeitszeiten** fest einplanen” “Frag früher nach **Hilfe**”

“‘Main wird dem Kunden gezeigt’ ist ein **Anreiz**, Dinge fertig zu machen” “**Kommunikation** ist ein riesiger Overhead”

“Dem **Prozess** zu folgen ist **lehrreicher** als Features zu schaffen”

“**Datenmodell** am Anfang **schnellstmöglich**
aufsetzen”

“Bessere **Zeitaufteilung** für Retros”

“Lieber Dinge einmal zu **viel kommunizieren**, als das es hinten runter fällt.” Chart **34**

Lessons Learned



Zeit Management

- **Kernarbeitszeit** als Team & **Arbeitszeiten** fest einplanen
- Bessere **Zeiteinteilung** für Retros
- **Datenmodell schnell** aufsetzen
- 'Main wird dem Kunden gezeigt' ist ein **Anreiz**, Dinge fertig zu machen

- **Stories schätzen** gar nicht anfangen, der **Mehrwert ist gering**



Kommunikation

- Sehr großer **Overhead**
- Lieber einmal zu **viel kommunizieren**, als dass etwas hinten runter fällt
- Frag früher nach **Hilfe**
- Die anderen sind **genauso planlos** wie du, **fragen** ist nicht schlimm

- **Pair programming** lohnt sich
- **Rede öfter** mit dem PO

Lessons Learned



Pair Programming macht mehr Spaß, sondern hilft auch bei Produktivität und erhöht den Lerneffekt.



Kommunikations-Overhead muss berücksichtigt werden, nimmt mehr Zeit in Anspruch als zu erwarten war



Arbeitszeit sollte sinnvoll eingeteilt werden.



Zusammenarbeit ist ein wichtiger Faktor für Projekte.



Verantwortlichkeiten sollten eindeutig und möglichst schnell festgelegt werden.

Was lief richtig toll?

“Team Kommunikation war **sehr gut**”

“Team Arbeit hat **Spaß** gemacht -Weekly meetings war **immer hilfreich**”

“Wir haben ein **funktionierendes** Produkt. 🤗”

“Die **Zusammenarbeit** in meiner Gruppe. Niemand hat sich treiben lassen und keiner ist vorausgeeilt. Alle haben ihren Teil dem Projekt **beigetragen**”

“**Team**arbeit/**Team**zusammenhalt”

“Dinge **ausprobieren**, gegen die Wand fahren, daraus lernen und **besser machen**”

“**Hilfsbereitschaft** im Team”

“**Kommunikation** im Team”

“Die Arbeit in **Team** 🦄”

“**sinnvolle** Aufgabenverteilung”

“**Schnelle** Reaktion von den Managing Rollen”

“Alles in allem haben Teams das gemacht, worauf sie **committed** haben.”

Was lief richtig toll?



Teamarbeit

- **Kommunikation** im Team
- **Zusammenarbeit** im Team
- **Hilfsbereitschaft** im Team
- **Zusammenhalt** im Team



“Wir haben ein **funktionierendes** Endprodukt.”

- “**Schnelle** Reaktion von den Managing Rollen”
- “**Sinnvolle** Aufgabenverteilung”
- “Die **Zusammenarbeit** in meiner Gruppe. Niemand ließ sich treiben & keiner ist vorgeeilt. Alle haben ihren Teil dem Projekt **beigetragen**”
- “Teamarbeit hat **Spaß** gemacht”
- “Dinge **ausprobieren**, gegen die Wand fahren, daraus lernen und **besser machen**”
- “Alle haben ihren Teil dem Projekt **beigetragen**”
- “Teams machten, wozu sie **committed** haben.”





Thank you!
Final Presentation - Team Rot

SSE 2022
Hasso-Plattner-Institut