

KARLSRUHER ENTWICKLERTAG / KARLSRUHE / 05.06.2024

Remote Mob Programming

Zuhause, aber nicht allein



JOSHUA TÖPFER
SENIOR CONSULTANT

**„Mob Programming
hat mein Leben
stark verändert“**

JOSHUA TÖPFER

Senior Consultant bei INNOQ

Joshua arbeitet seit mehr als 3 Jahren Vollzeit im Remote Mob. Er ist Maintainer von mob.sh und coacht Teams im Remote Mob Programming.



Wer hat schon mal Pair Programming gemacht?



Wer hat schon mal Mob Programming gemacht?



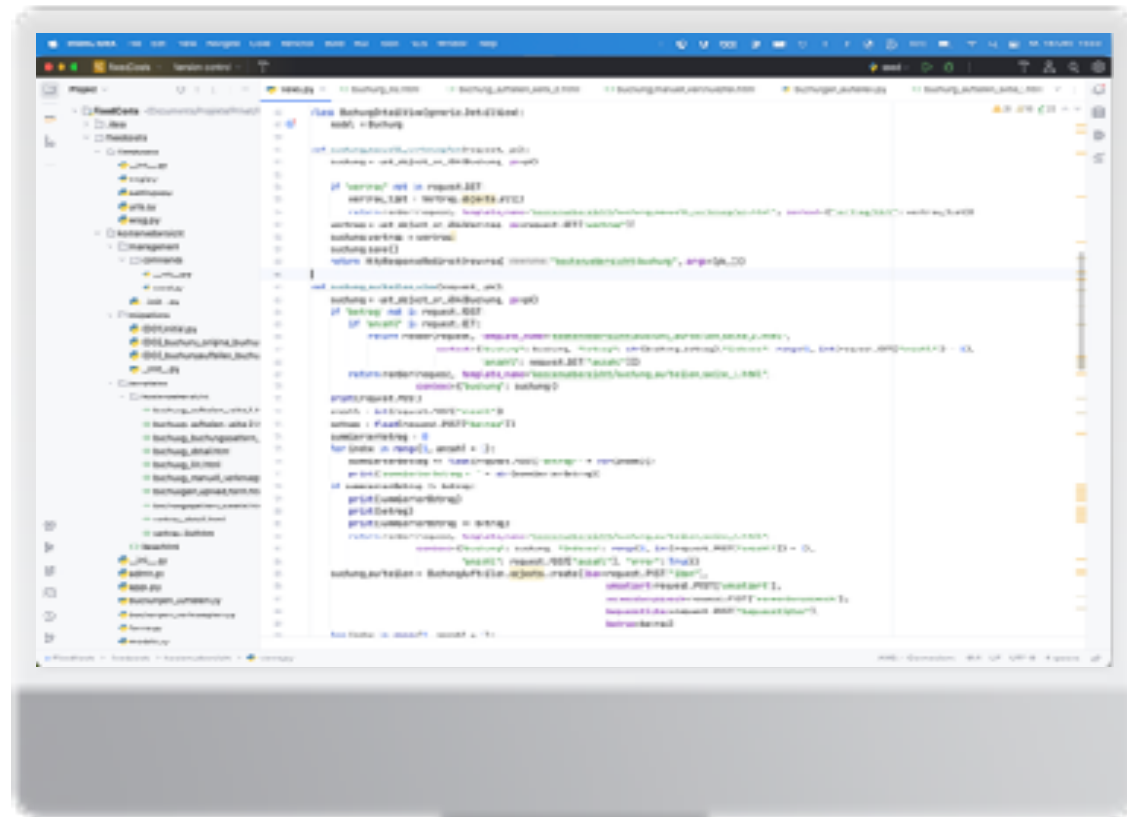
Wer hat schon mal Remote Mob Programming gemacht?



Was ist Mob Programming?

**All the brilliant minds working together
on the same thing, at the same time, in
the same space, and at the same
computer**

Woody Zuill



Typist



Typist



Typist



Typist



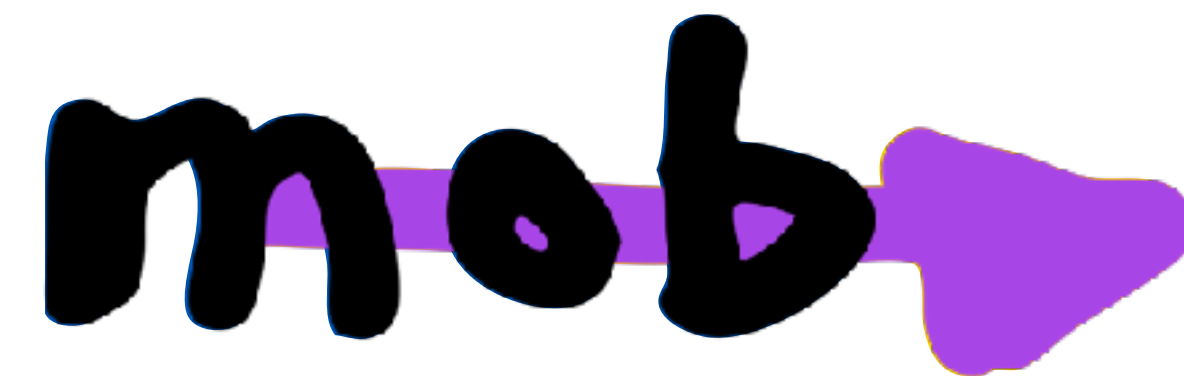
Typist

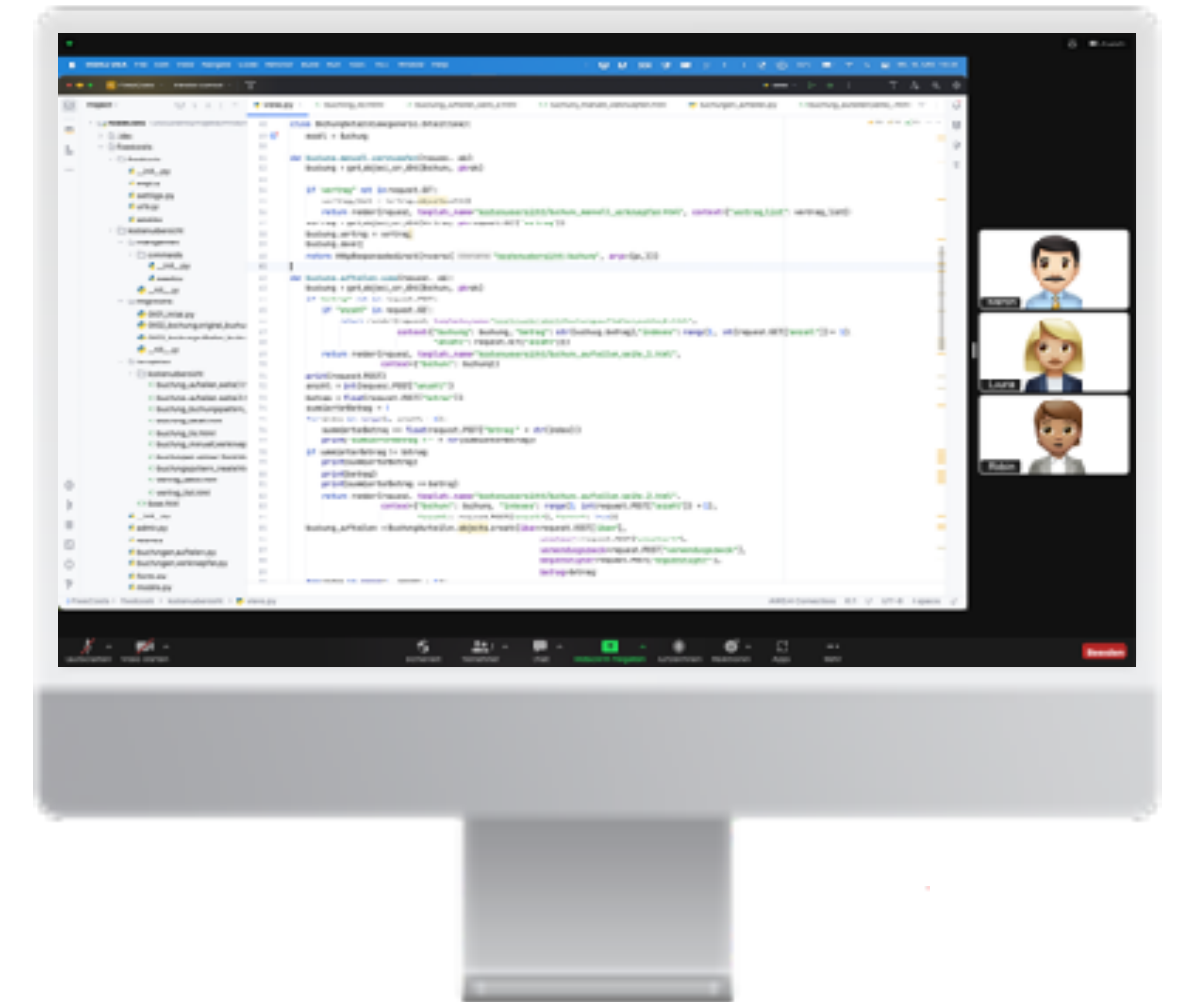
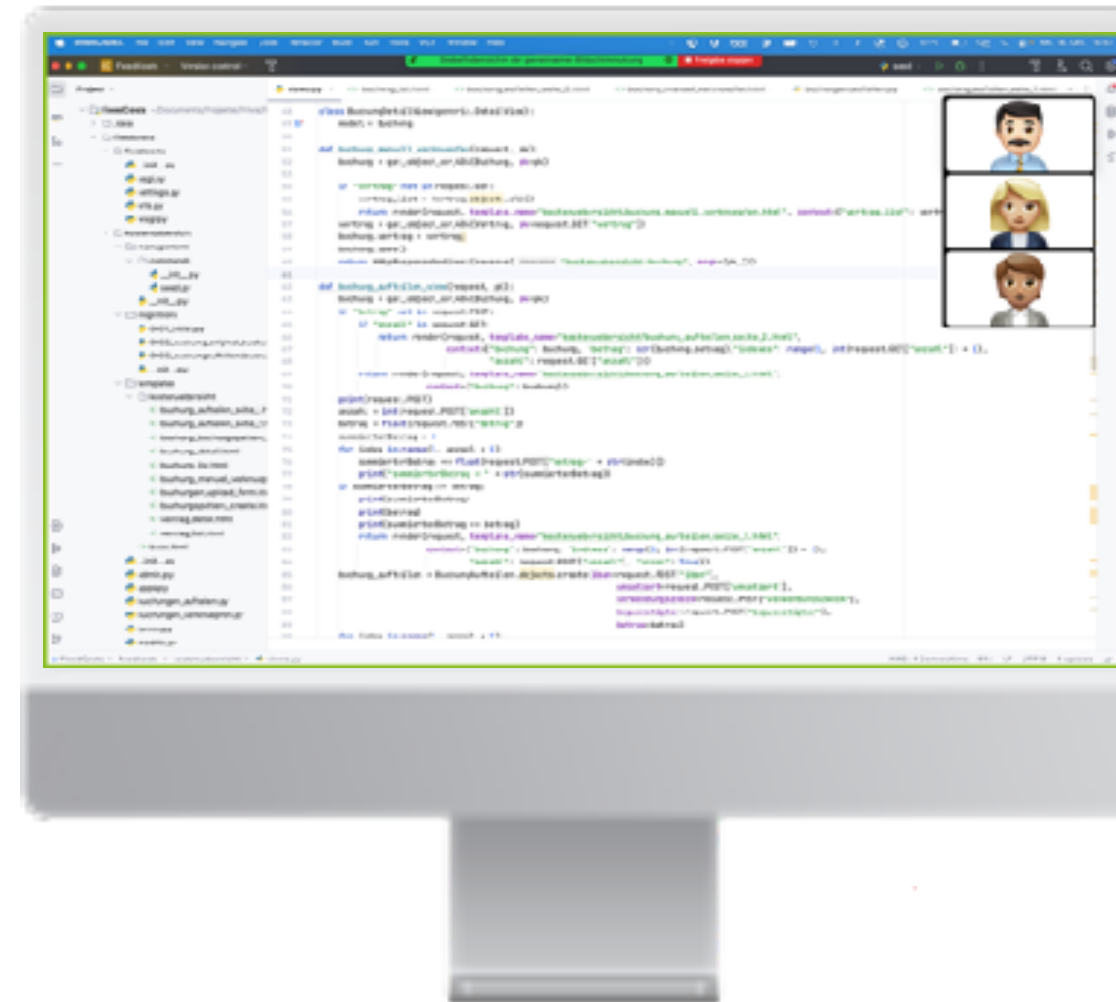
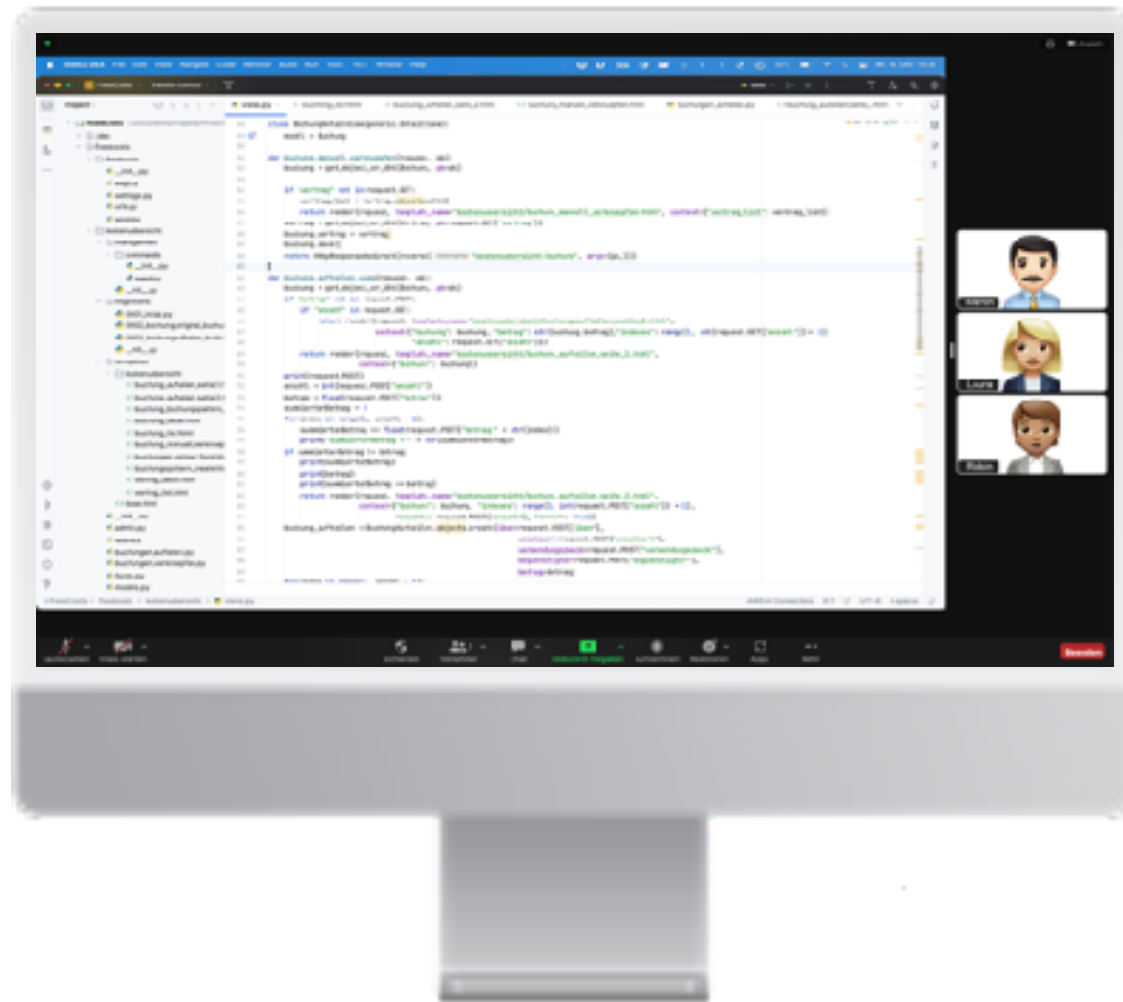


Was ist Remote Mob Programming?

Virtueller Teamraum

- Fester Ort
- Alle sehen und hören sich
- Alle sehen den gemeinsamen Bildschirm
- Tastatur weitergeben
- Whiteboard



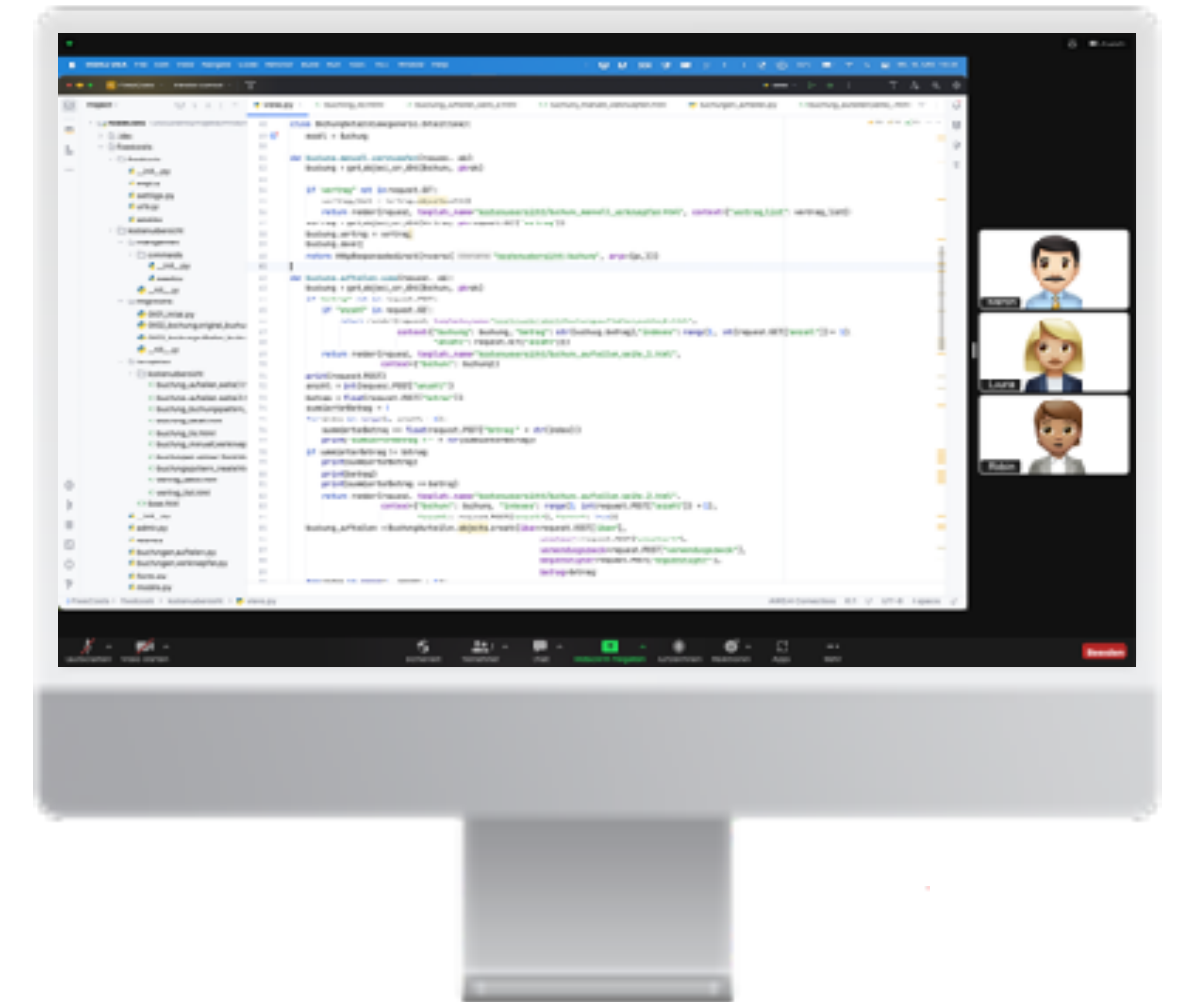
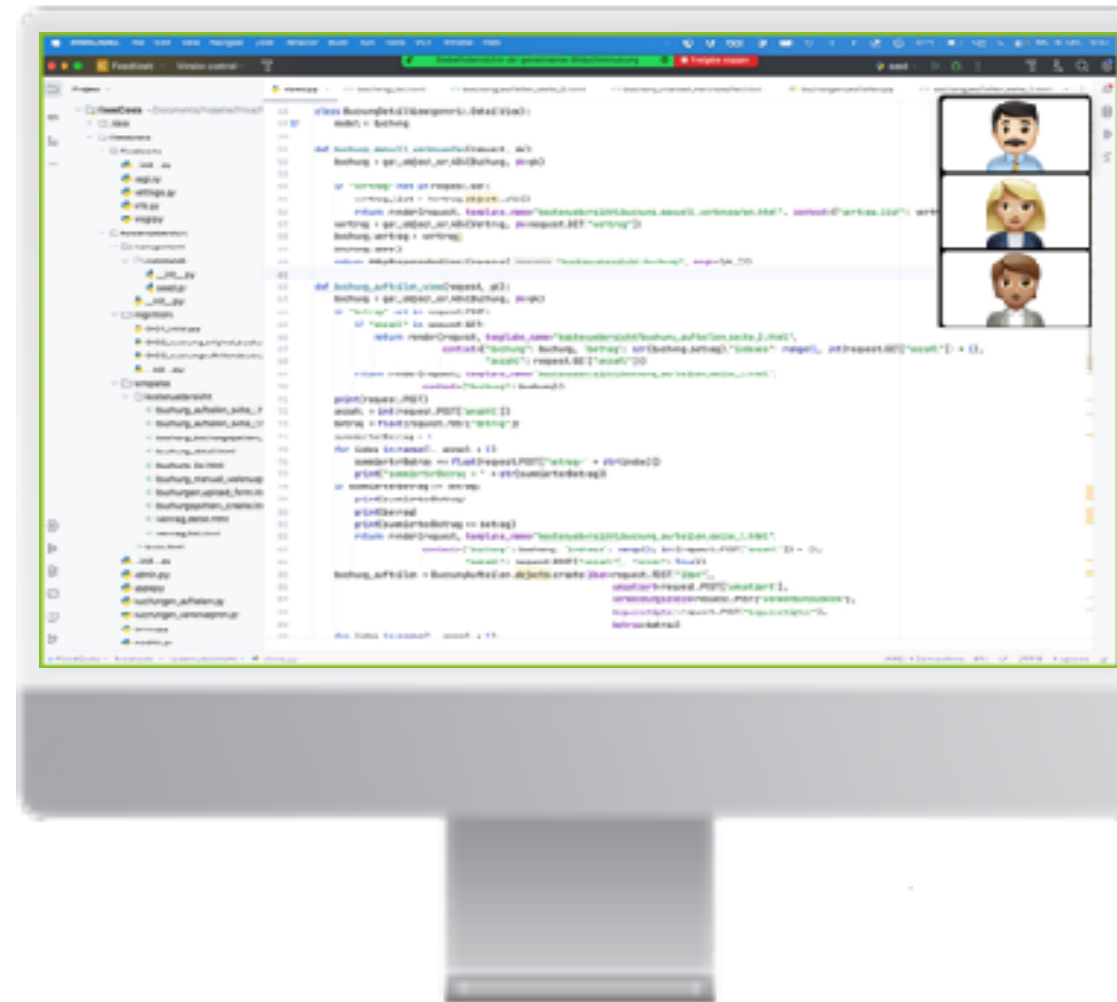
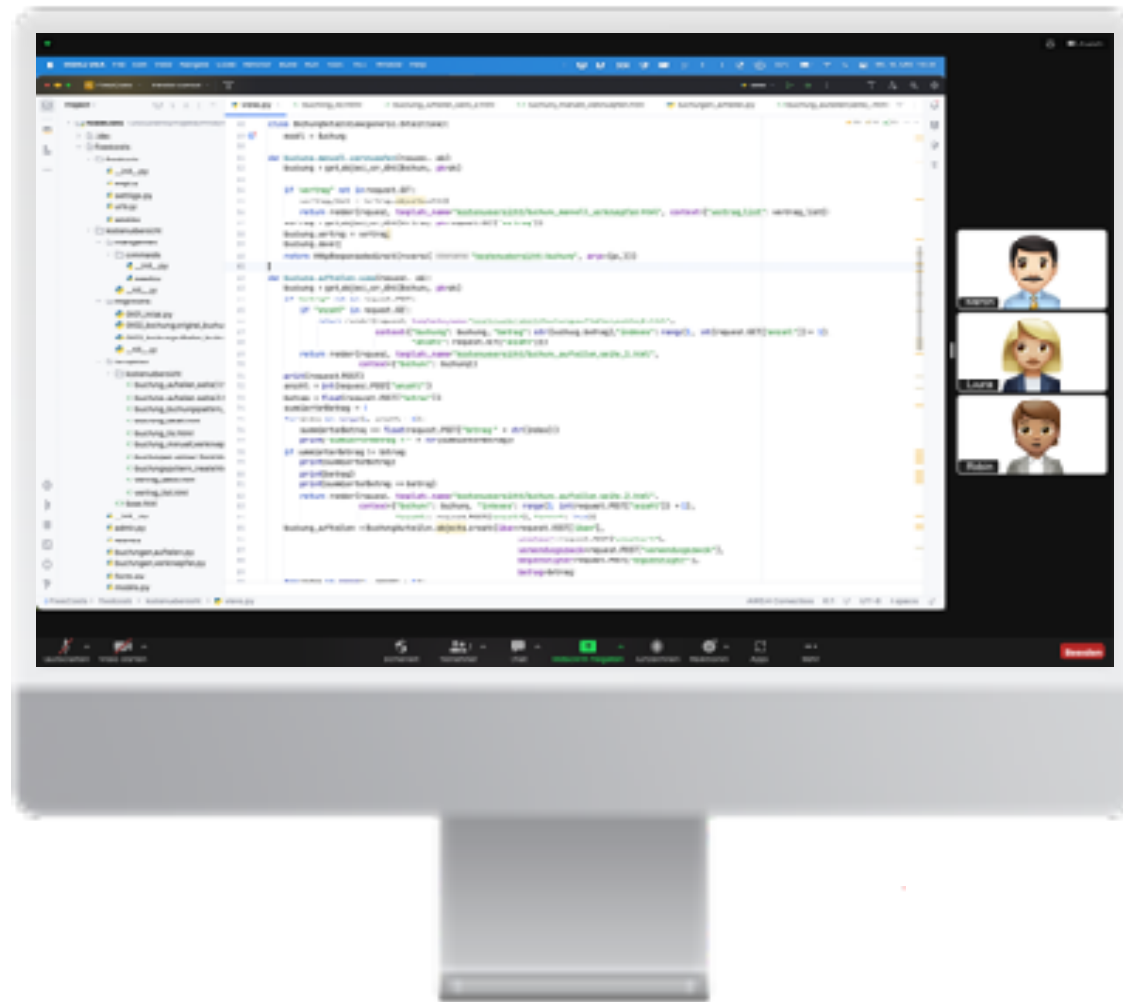


Typist

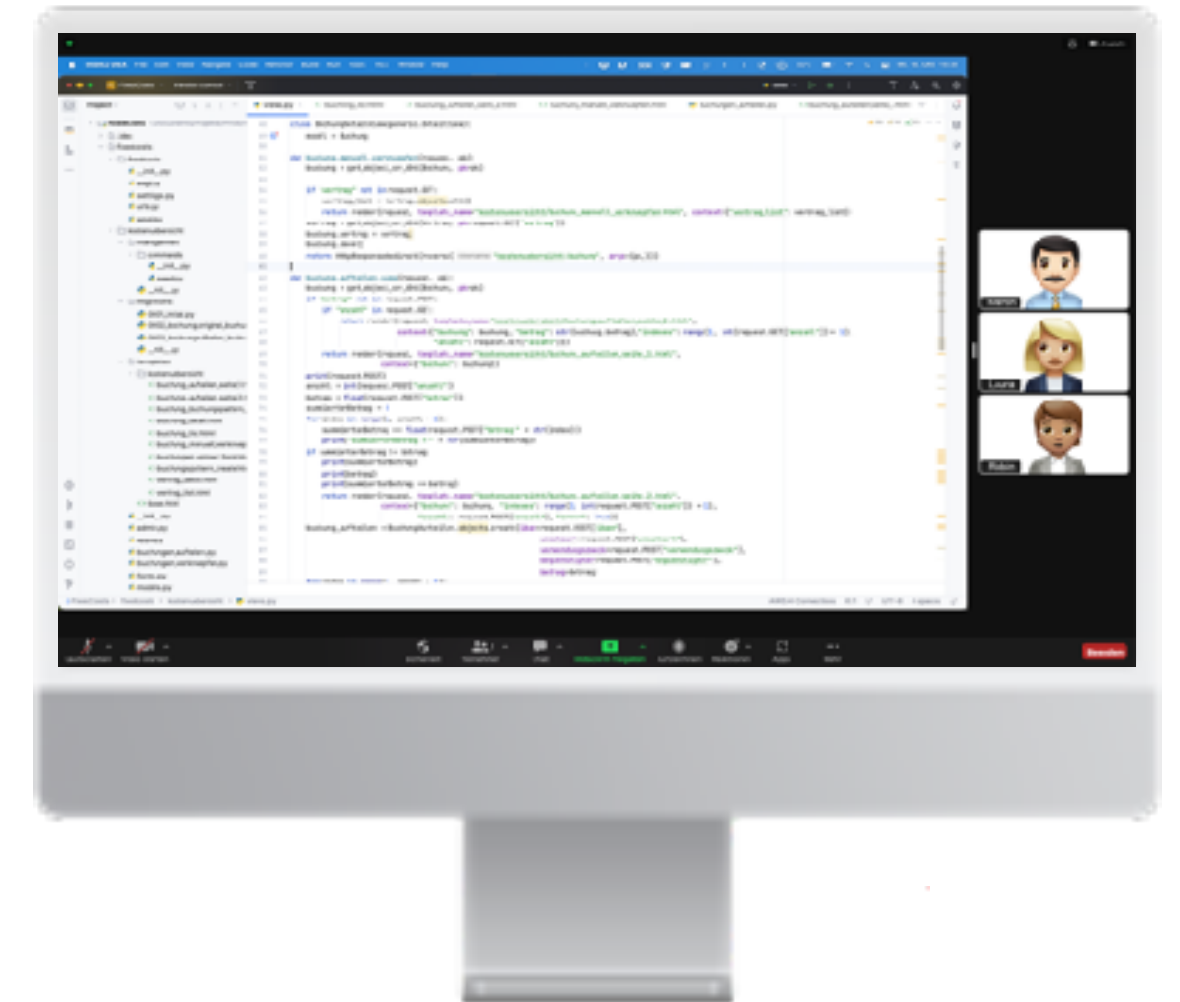
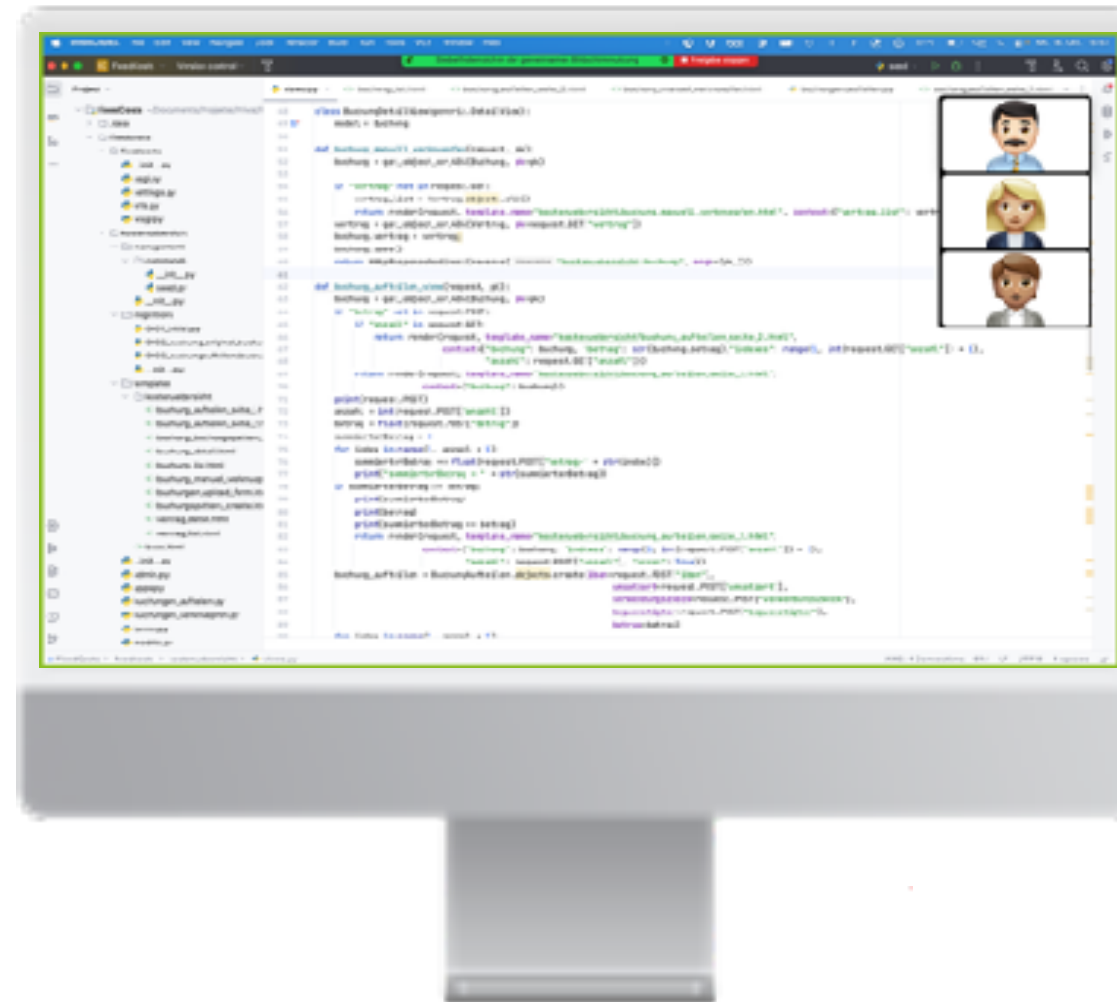
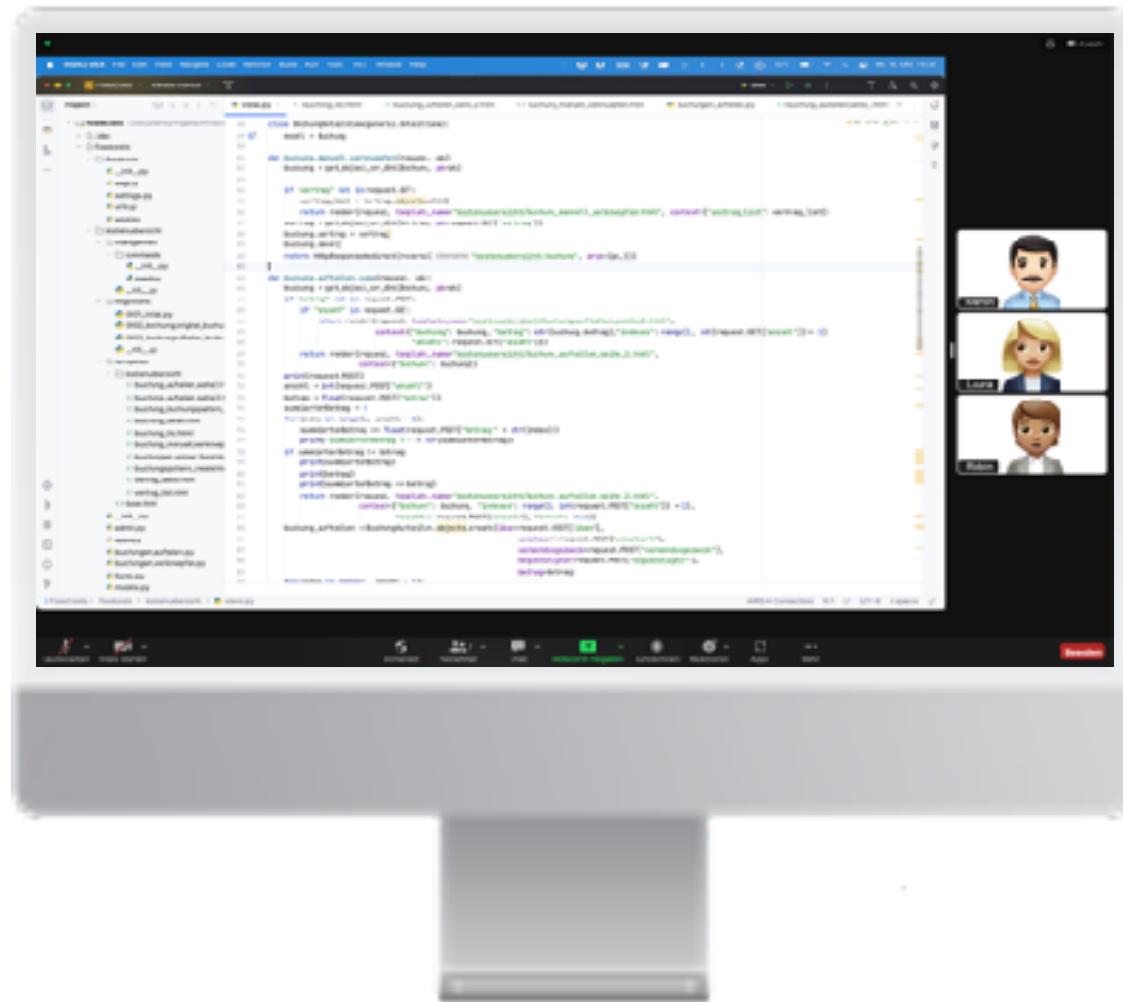


testrepo — joshuatopfer@Laptop-von-Joshua — ..NNOQ/testrepo — -zsh — 76x12

```
→ testrepo git:(main) mob start
  git fetch origin --prune
  git merge FETCH_HEAD --ff-only
> starting new session from origin/main
  git checkout -B mob/main origin/main
  git commit --allow-empty -m mob start [ci-skip] [ci skip] [skip ci]
  git push --no-verify --set-upstream origin mob/main:mob/main
> you are on wip branch 'mob/main' (base branch 'main')
5b5ca6f 2 seconds ago <Joshua Töpfer>
> PUT https://timer.mob.sh/test-room {"timer":1,"user":"Joshua Töpfer"}
> It's now 17:42. 1 min timer ends at approx. 17:43. Happy collaborating! :)
→ testrepo git:(mob/main) █
```



Typist

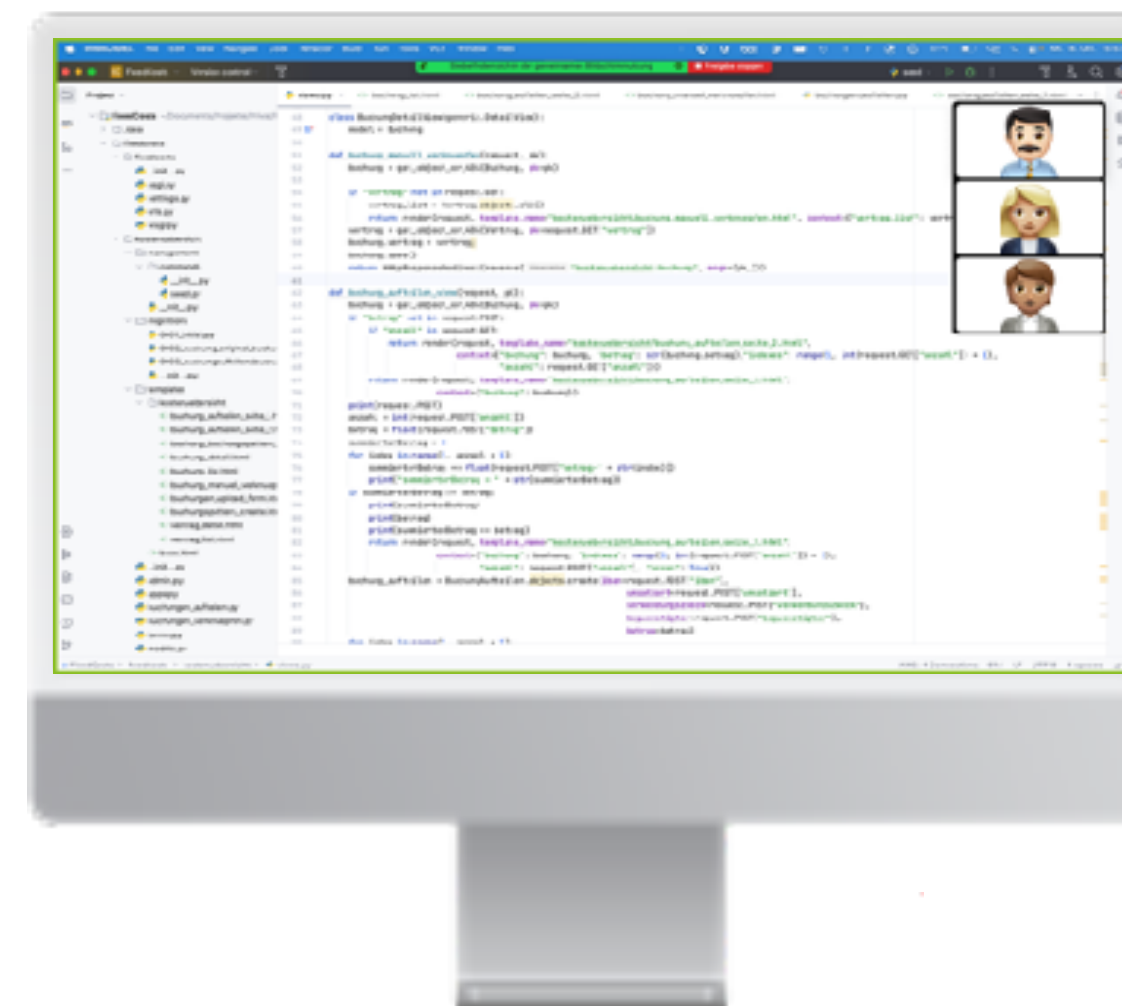
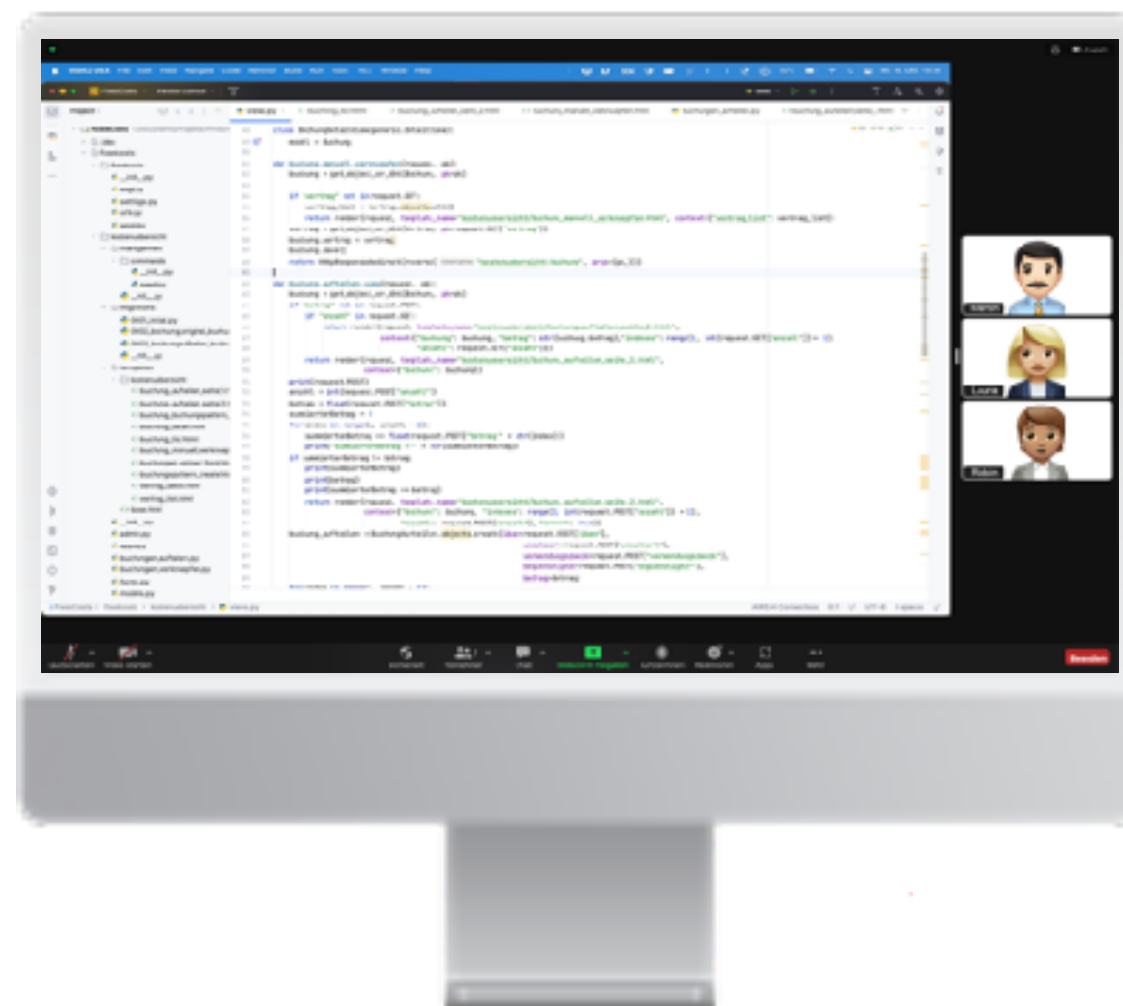
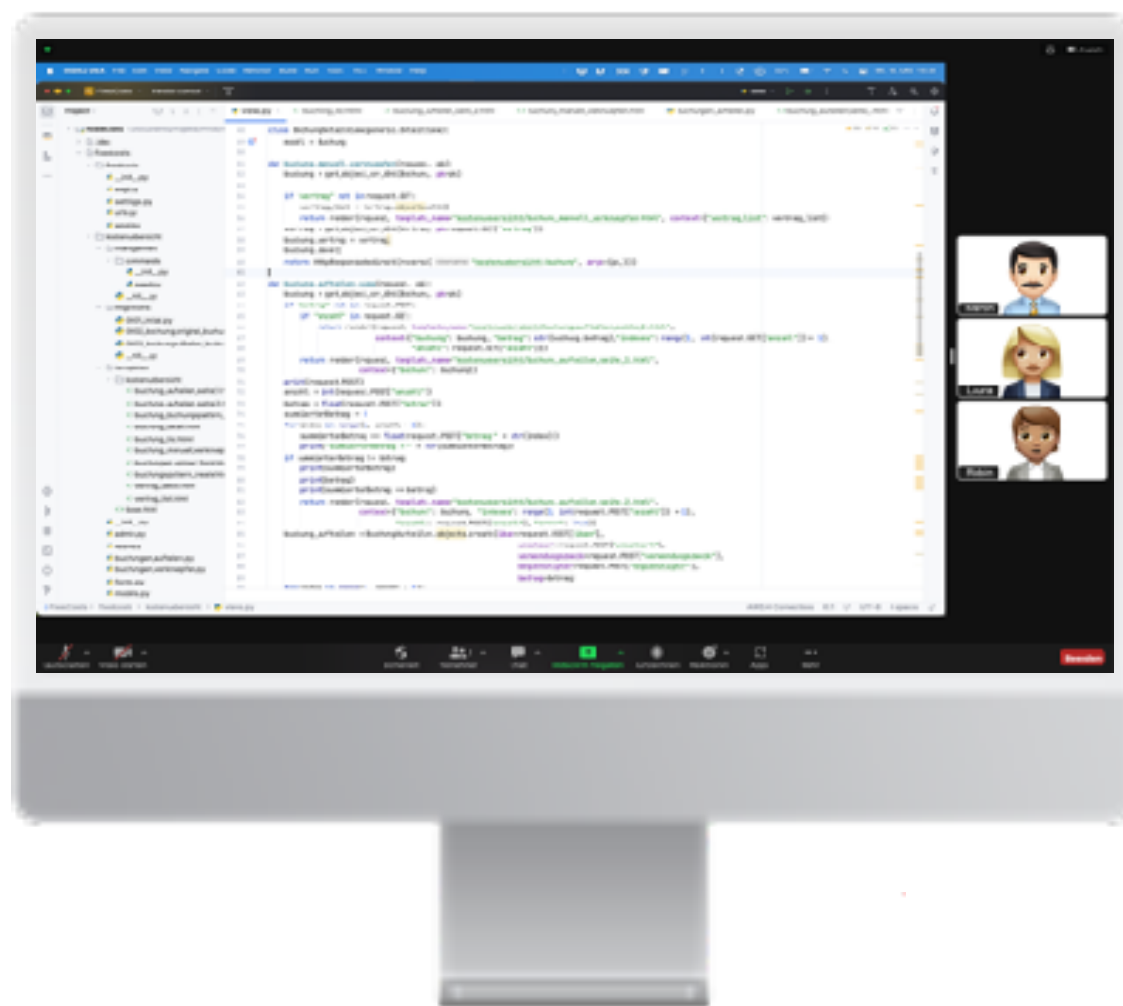


Typist

testrepo — joshuatopfer@Laptop-von-Joshua — ..NNOQ/testrepo — -zsh — 51x11

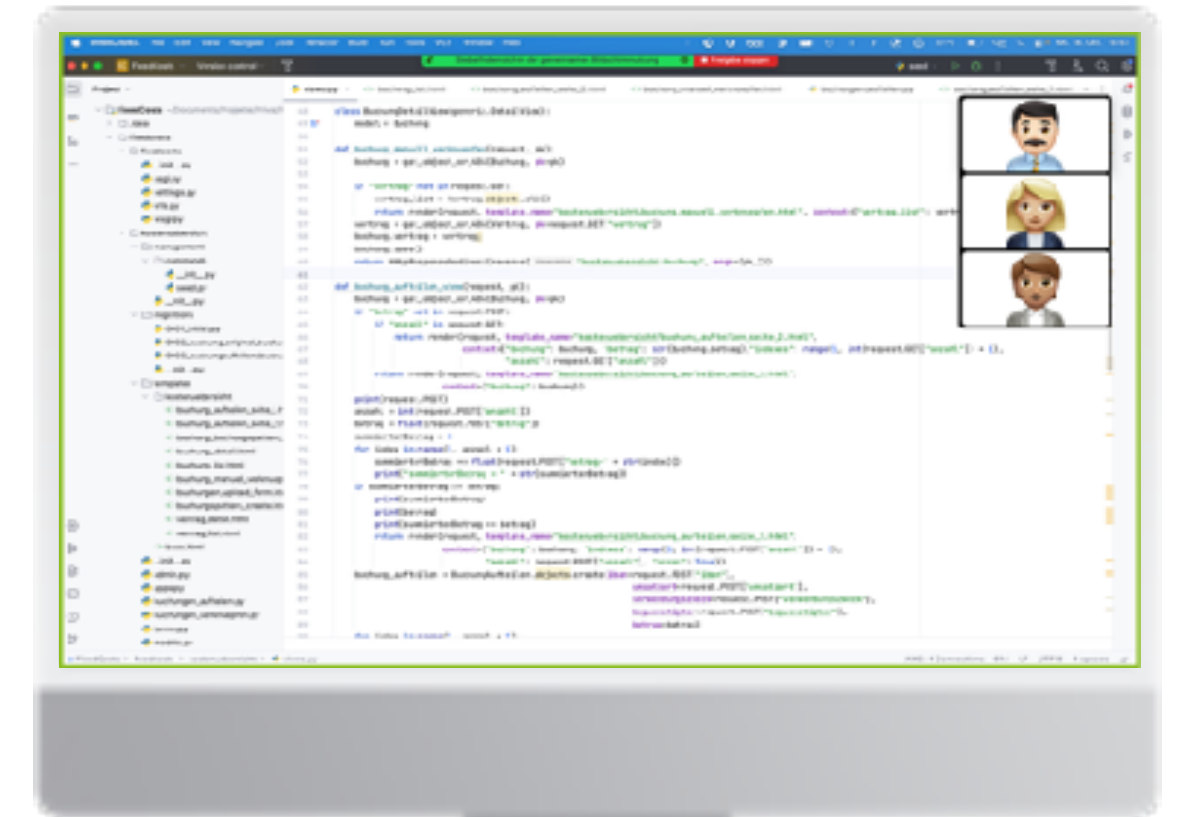
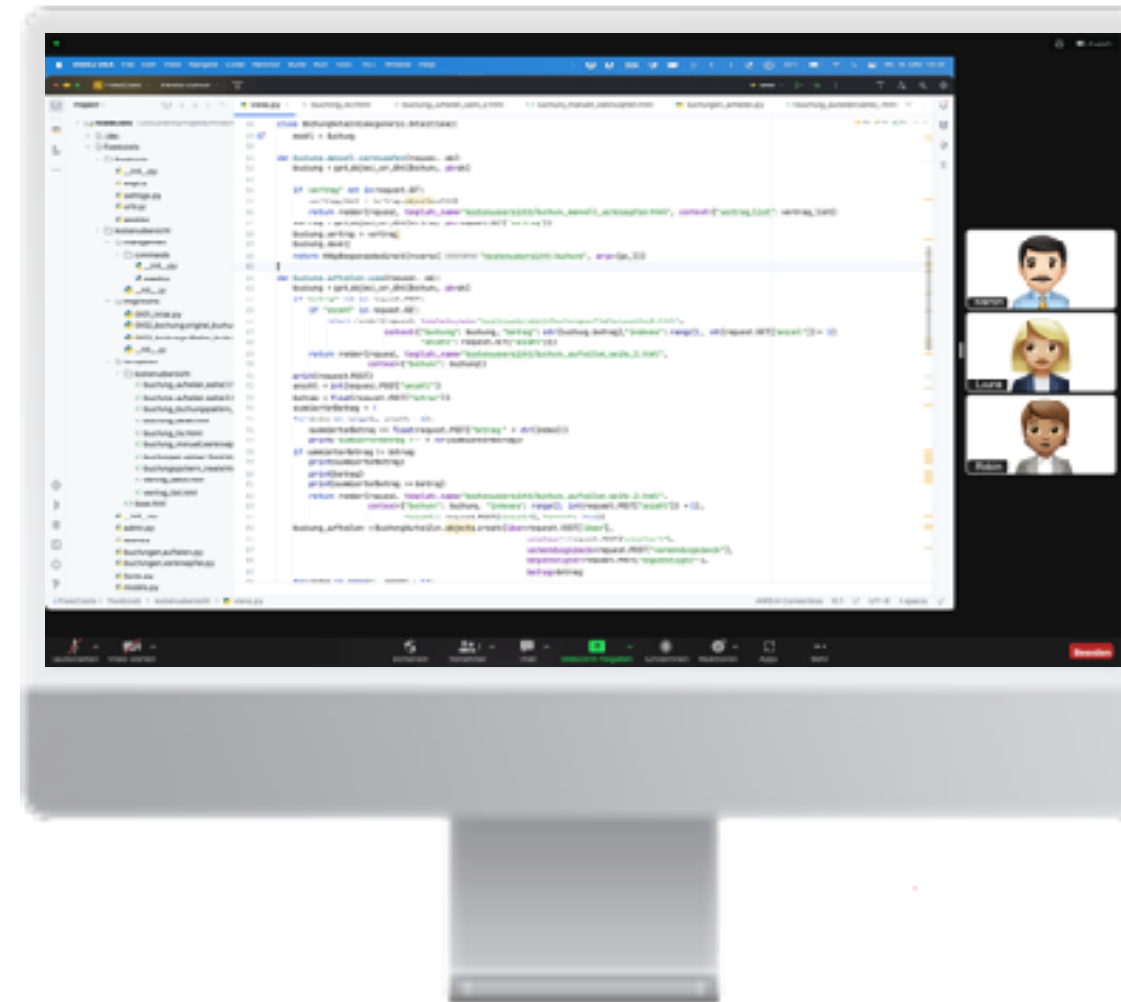
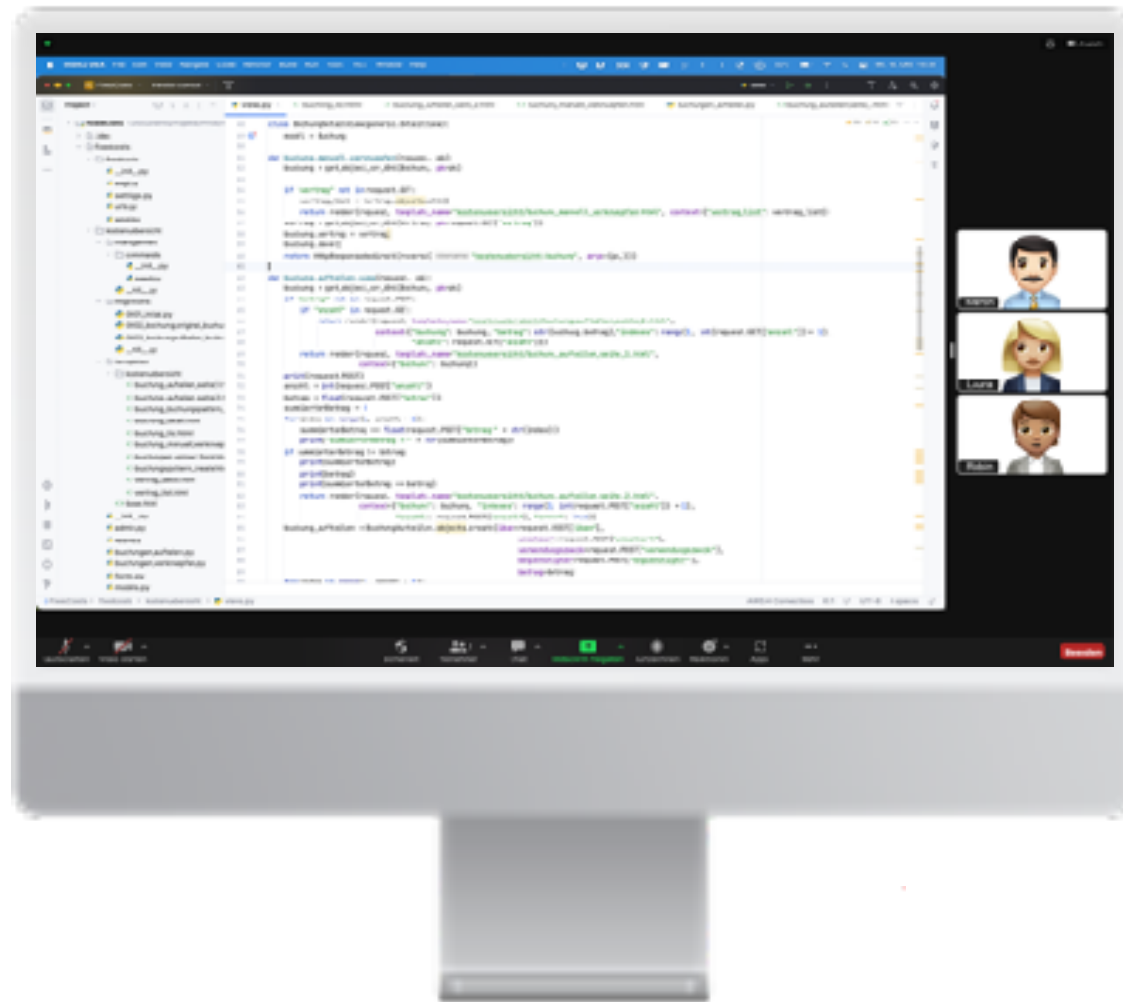
```
→ testrepo git:(mob/main) ✕ mob next
git add --all
git commit --message mob next [ci-skip] [ci skip]
[skip ci]

lastFile:text.txt --no-verify
text.txt | 0
1 file changed, 0 insertions(+), 0 deletions(-)
1465761cbc5963998e18a2ac61412ddc6f309eff
git push --no-verify origin mob/main
→ testrepo git:(mob/main) █
```



Typist

```
testrepo — joshuatopfer@Laptop-von-Joshua — ..NNOQ/testrepo — -zsh — 76x11
→ testrepo git:(mob/main) mob start
  git fetch origin --prune
> joining existing session from origin/mob/main
  git checkout -B mob/main origin/mob/main
  git branch --set-upstream-to=origin/mob/main mob/main
> you are on wip branch 'mob/main' (base branch 'main')
5b5ca6f 8 minutes ago <Joshua Töpfer>
31d2d5e 21 seconds ago <Joshua Töpfer>
> PUT https://timer.mob.sh/test-room {"timer":1,"user":"Joshua Töpfer"}
> It's now 17:50. 1 min timer ends at approx. 17:51. Happy collaborating! :)
→ testrepo git:(mob/main) █
```



Typist



→ testrepo git:(mob/main) mob done

```
git fetch origin --prune
```

```
git push --no-verify origin mob/main
```

```
git checkout main
```

```
git merge origin/main --ff-only
```

```
git merge --squash --ff mob/main
```

```
git branch -D mob/main
```

```
git push --no-verify origin --delete mob/main
```

```
  tesfile1.txt | 1 +
```

```
  testfile2.txt | 0
```

```
  2 files changed, 1 insertion(+)
```

👉 To finish, use

```
git commit
```

→ testrepo git:(main) ✕

Warum?

Time-to-Market
anstatt Ressourcen-Effizienz

Konstanter Fortschritt

- Fokus auf das wichtigste Ticket
- Kontextwechsel & Wartezeiten vermeiden
- Keine Code Reviews



Termine

- Team tritt geschlossen auf
- Unnötige Termine beseitigen
- Botschafter Prinzip
- Regeltermine wie Daily Scrum sind überflüssig



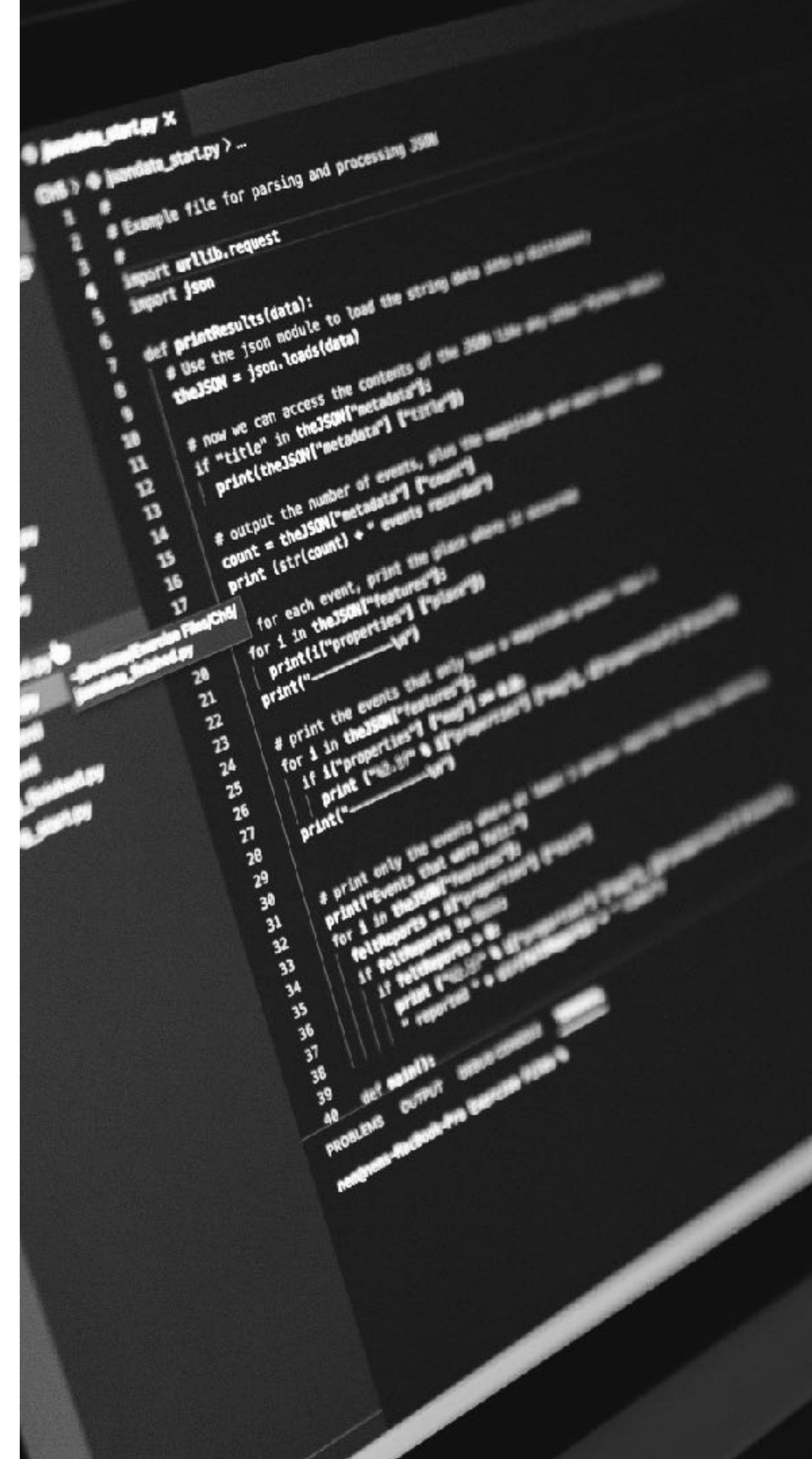
Architekturarbeit

- Entscheidungen explizit machen
- Team übernimmt die Rolle des Architekten wenn erforderlich
- Architecture Decision Records zusammen erarbeiten und diskutieren
- Konsent statt Konsens



Codequalität

- Viele Augen sehen viel
- Weniger Bugs
- Verzicht auf asynchrone Code Reviews
- Gemeinsames Refactoring



Lernen von den Anderen

- Kontinuierliches Lernen
- Busfaktor reduzieren
- Onboarding ohne Aufwand



Zuhause aber nicht allein

- Mittagessen mit der Familie
- Kontakt zu Kollegen
- Nicht ins Büro fahren müssen

Foto von Vlada Karpovich: <https://www.pexels.com/de-de/foto/dammerung-fashion-frau-entspannung-4050387/>



Die *Silver* Bullet?

Verschiedene Zeitzonen

- Bei kleiner Verschiebung:
 - Gemeinsame Zeiten schwieriger zu finden
 - Möglichkeit der Kernzeit im Mob
 - Abstimmungen werden wieder benötigt
- Bei großen Verschiebungen:
 - Nur kurze Mob Sessions
 - 24/5 Mob um die Welt



Konflikte im Team

- Konflikte kommen schnell an die Oberfläche
- Einführung unbedingt durch Coach begleiten
- Zu Beginn viele kleine Retrospektiven



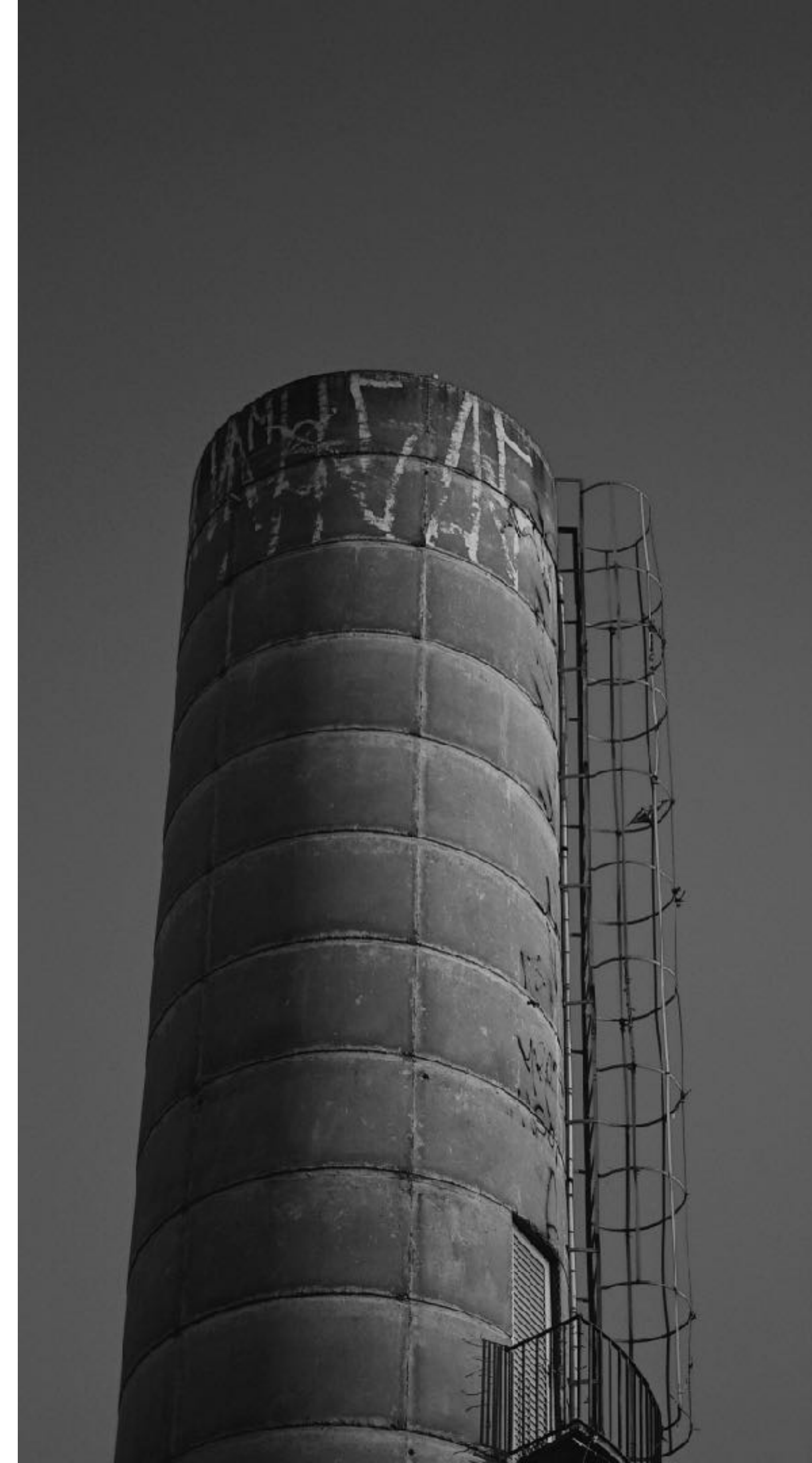
Abhängigkeiten

- Wartezeiten
- Kontextwechsel
- Sollten aufgebrochen werden
- Im besten Fall hat das Team End-to-End Verantwortung



Silos aufbrechen

- Wissen wird im Team verteilt
- Sehr tiefes Wissen wird nur schwierig verteilt
- Busfaktor wird reduziert, aber nicht komplett mitigiert



Passt nicht für jeden

- Social Fatigue
- Social Anxiety
- Over Stimulation
- Etc.



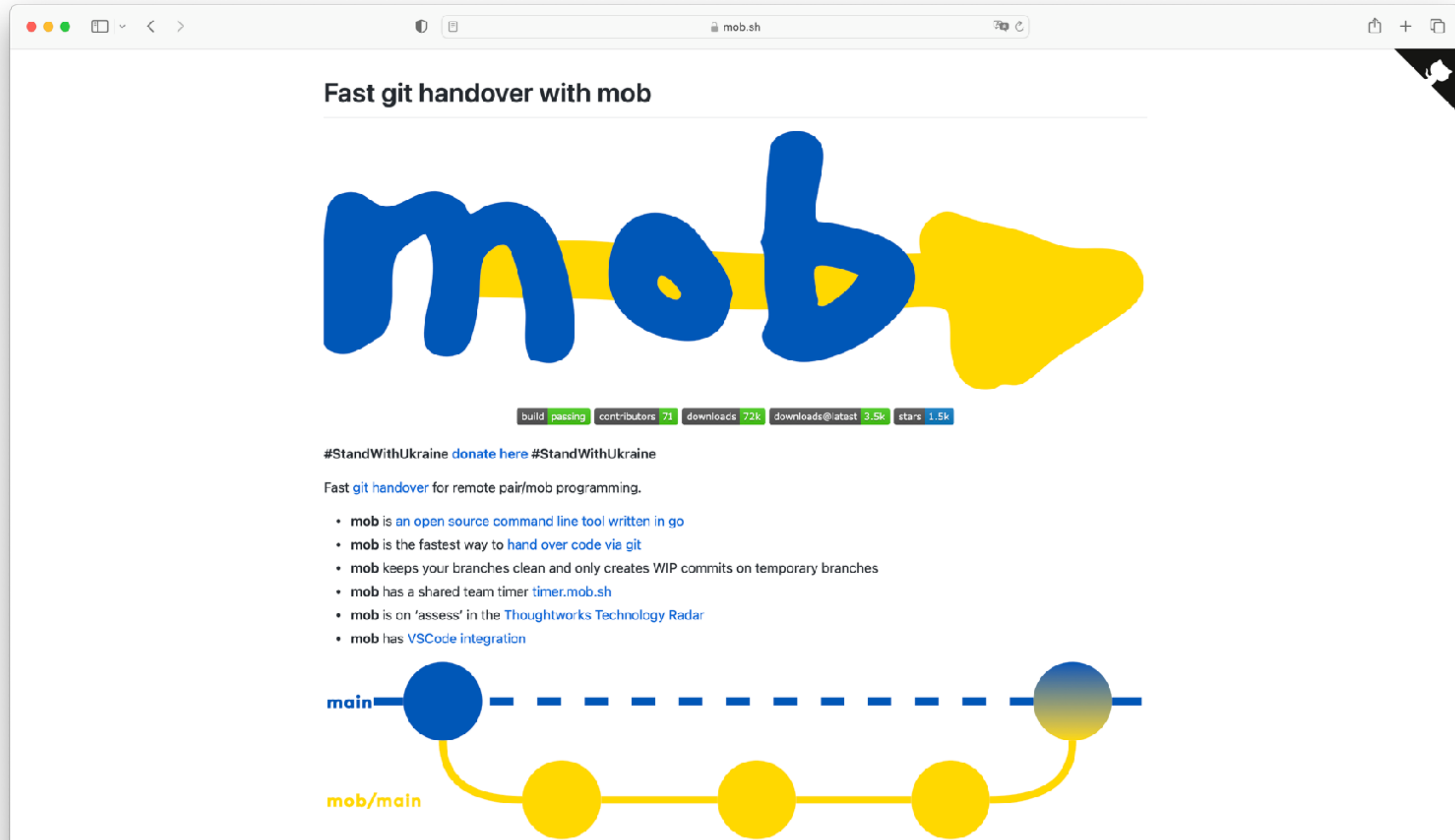
Wie anfangen?

Ein Experiment

- Ein Sprint einfach mal Vollzeit ausprobieren
- Viele Retrospektiven
- Mit Coach begleiten (Ich unterstütze gerne 😊)
- 1 Tages Workshop für das Team von Socreatory oder einzeln ausprobieren bei mobusoperandi.com



mob.sh



The screenshot shows the mob.sh website in a browser window. The page has a clean, modern design with a white background. At the top, the title "Fast git handover with mob" is displayed. Below the title is a large, stylized logo for "mob" in blue, followed by a yellow arrow pointing to the right. Underneath the logo is a row of status indicators: "build passing", "contributors 71", "downloads 72k", "downloads@latest 3.5k", and "stars 1.5k". Below these indicators is a line of text: "#StandWithUkraine [donate here](#) #StandWithUkraine". The main content area features a paragraph: "Fast [git handover](#) for remote pair/mob programming." followed by a bulleted list of features. At the bottom, there is a diagram illustrating the workflow. It shows a horizontal line with a blue circle on the left labeled "main" and a blue-to-yellow gradient circle on the right. A dashed blue line connects them. Below this, a yellow line with four yellow circles represents the "mob/main" workflow, with a yellow label "mob/main" at the start.

Fast git handover with mob

mob

build passing contributors 71 downloads 72k downloads@latest 3.5k stars 1.5k

#StandWithUkraine [donate here](#) #StandWithUkraine

Fast [git handover](#) for remote pair/mob programming.

- **mob** is an [open source command line tool written in go](#)
- **mob** is the fastest way to [hand over code via git](#)
- **mob** keeps your branches clean and only creates WIP commits on temporary branches
- **mob** has a shared team timer [timer.mob.sh](#)
- **mob** is on 'assess' in the [Thoughtworks Technology Radar](#)
- **mob** has [VSCode integration](#)

main

mob/main

Unser Primer



www.innoq.com/primer

Kostenlos herunterladen

Jochen Christ
Simon Harrer
Martin Huber

Remote Mob Programming

At home, but not alone

INNOQ

Foreword by Mark Pearl

remotemobprogramming.org

The screenshot shows the homepage of remotemobprogramming.org. The page has a clean, minimalist design with a white background. At the top, the site's name is in a large, dark teal font. Below the header, there's a section titled "Remote Mob Programming" in a bold, dark font. Underneath this, a line of text reads "New: [How to act as a typist](#) and a [rotation timer](#)".

The central focus of the page is a laptop displaying a code editor with a dark theme and a video call window on the right side showing four participants. Surrounding the laptop are several handwritten annotations in black ink, connected to the laptop by curved arrows:

- Camera Always On**: An arrow points from this text to a camera icon above the laptop screen.
- Screen Sharing**: An arrow points from this text to a screen sharing icon (a monitor with an upward arrow) above the laptop screen.
- Git Handover**: An arrow points from this text to a hand holding a card with "GIT" on it, located to the right of the laptop.
- 10 Minutes Intervals**: An arrow points from this text to a clock icon to the left of the laptop.
- Small Team**: An arrow points from this text to a group of three people icon to the right of the laptop.
- Dine with your Family**: An arrow points from this text to a fork and knife icon at the bottom right of the laptop.
- Typist and the Rest of the Mob**: An arrow points from this text to a keyboard icon at the bottom left of the laptop.

Below the laptop image, there is a paragraph of text explaining the concept of Remote Mob Programming:

Remote Mob Programming combines two ways of working: Mob Programming and working as a distributed team. Woody Zuill describes [Mob Programming](#) as creating the "same thing, at the same time, in the same space, and on the same computer". Working in the same space clashes with working as a distributed team at first glance, but actually, it goes together really well. With Remote Mob Programming, we collaborate closely in the same virtual space. But Remote Mob Programming is more than that.

Mein Fazit

- Mehr Fokus
- Kontinuierliches Lernen
- High performing Team
- Teamkollegen sind Freunde
- Mehr Zeit für die Familie



Danke! Fragen?



Joshua Töpfer
joshua.toepfer@innoq.com
+49 172 3666344
@joshuatoepfer

Feedback & Anregungen



innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim
+49 2173 3366-0

Ohlauer Str. 43
10999 Berlin

Ludwigstr. 180E
63067 Offenbach

Kreuzstr. 16
80331 München

Wendenstraße 130
20537 Hamburg

Spichernstraße 44
50672 Köln

Königstorgraben 11
90402 Nürnberg