

WE ARE DEVELOPERS WORLD CONGRESS / BERLIN

Remote Mob Programming

At home, but not alone



JOSHUA TÖPFER
SENIOR CONSULTANT

**„Mob Programming
has changed my life“**

JOSHUA TÖPFER

Senior Consultant at INNOQ

Joshua has been working full-time in remote mob for more than 4 years. He is the maintainer of mob.sh and coaches teams in remote mob programming.



Who has already done pair programming?



Who has already done mob programming?



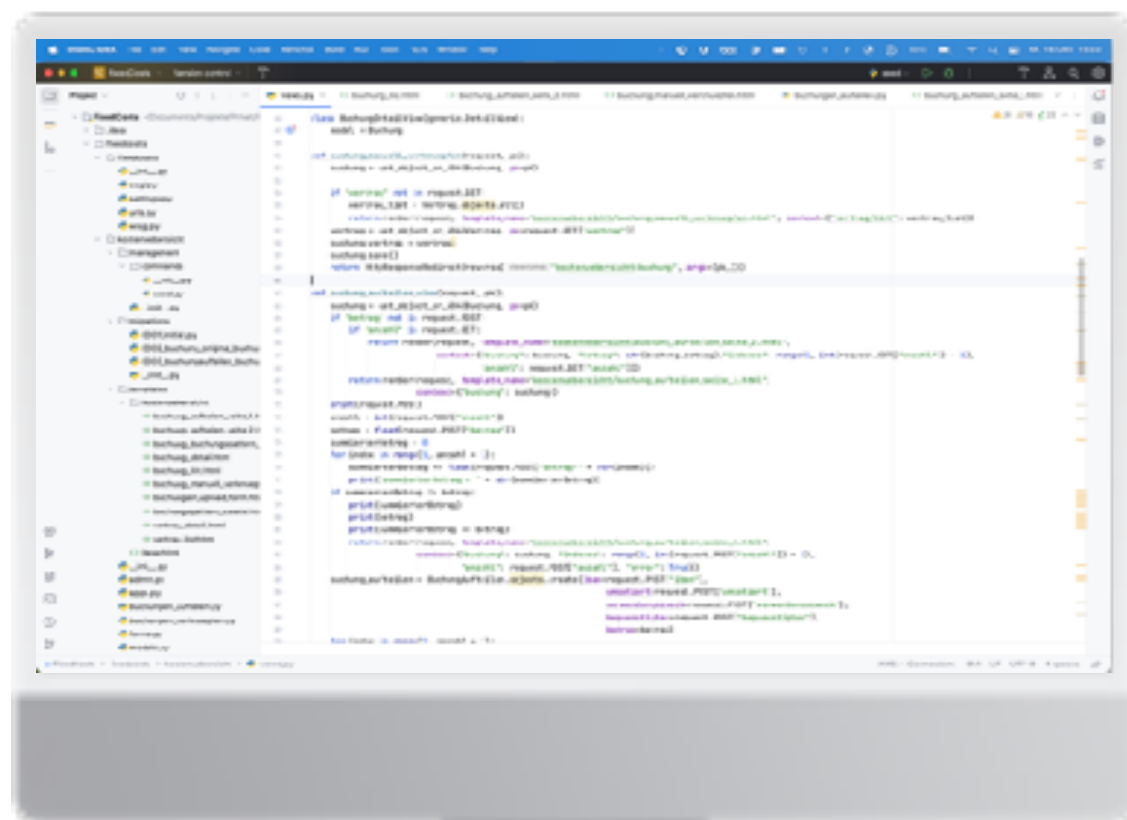
Who has already done remote mob programming?



What is 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



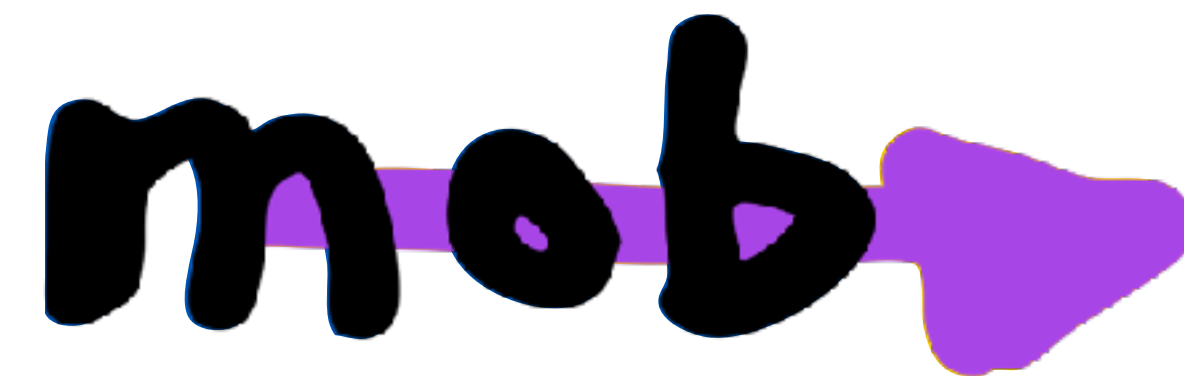
Typist

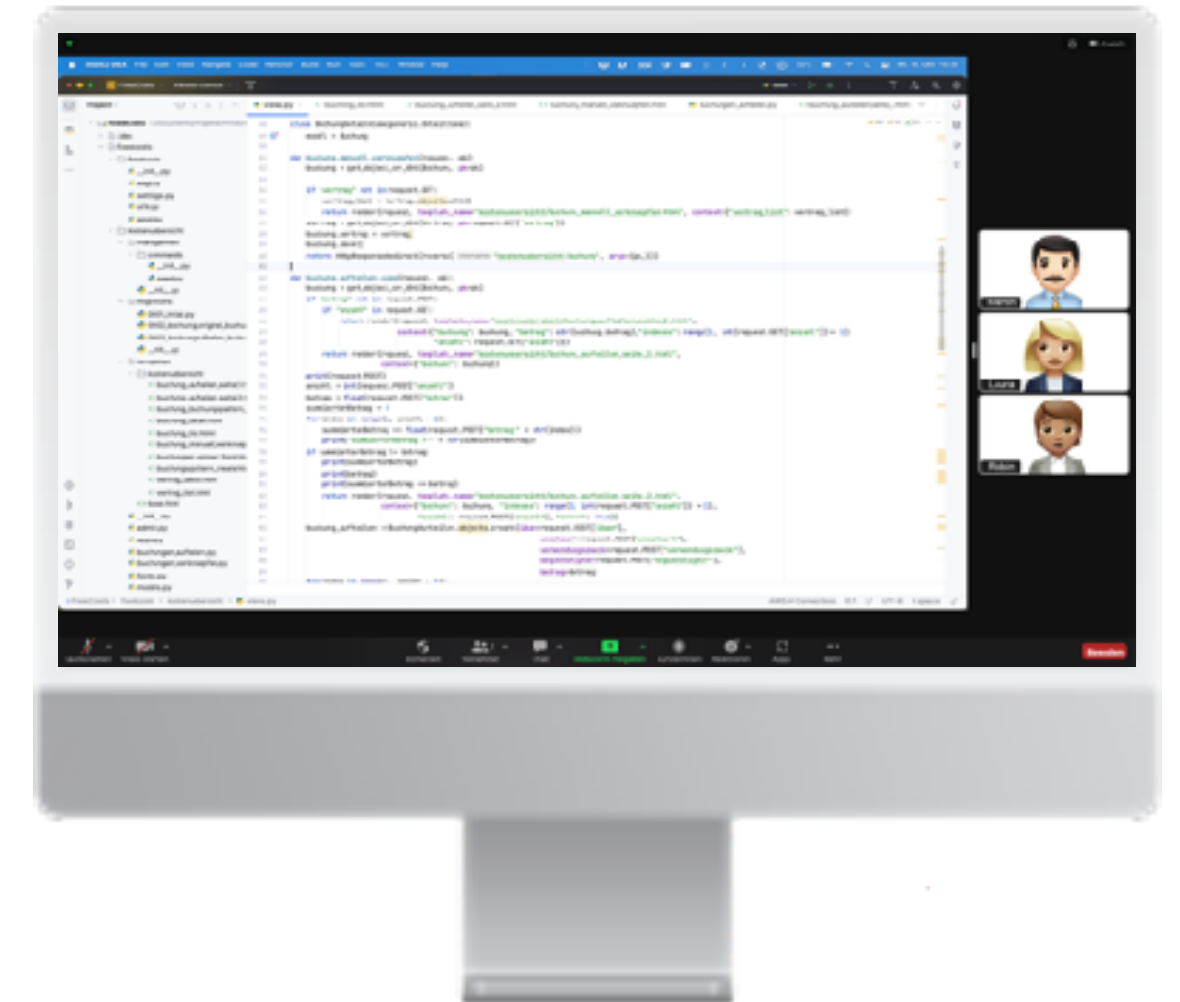
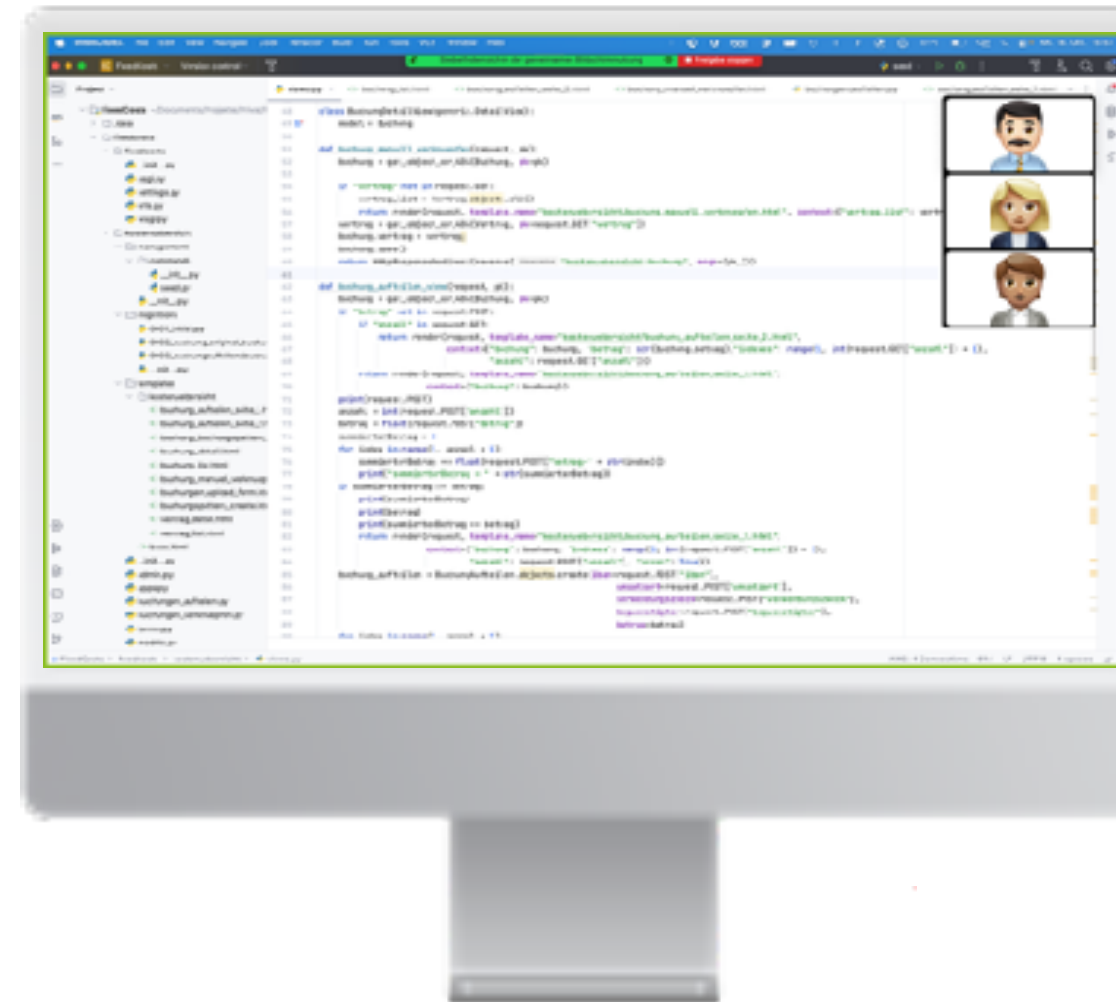
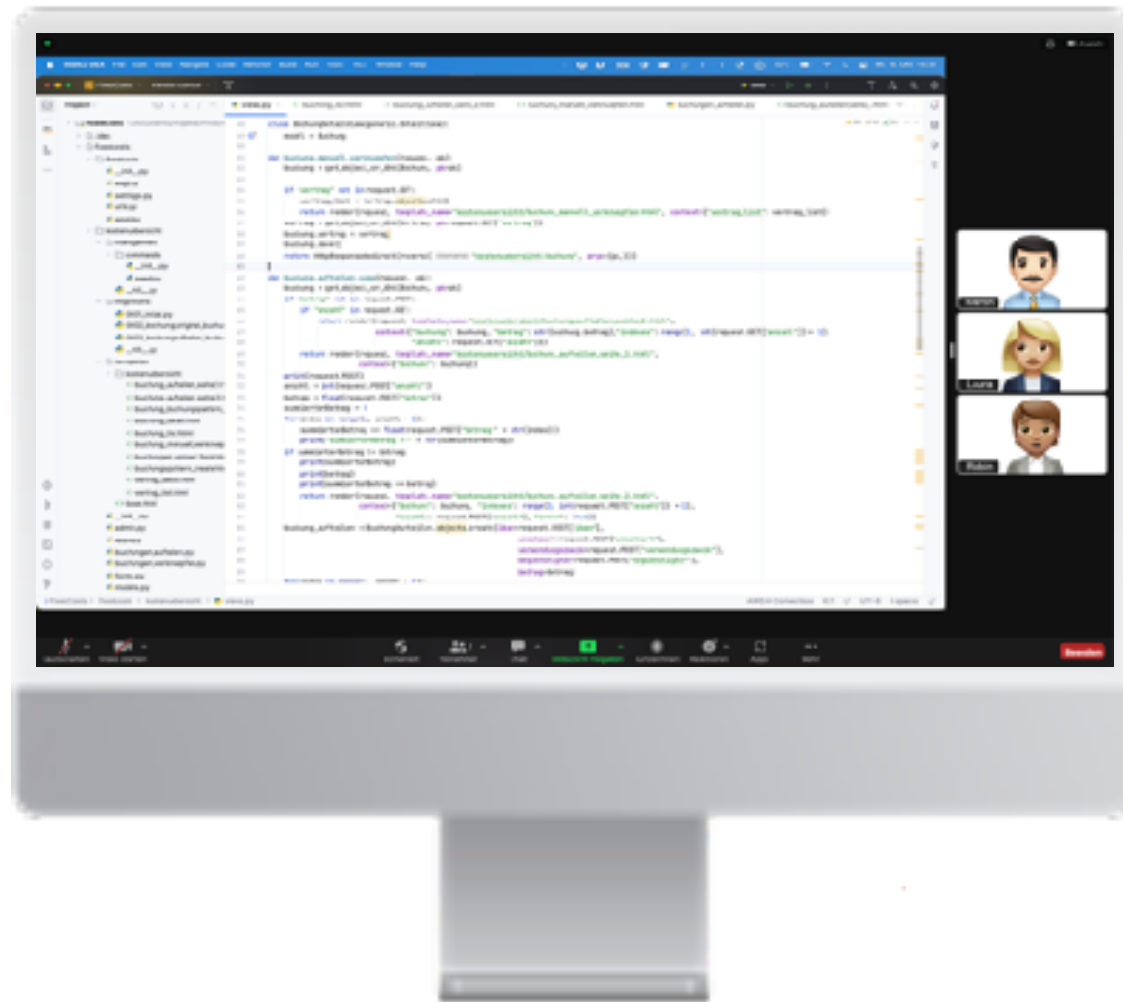


What is Remote Mob Programming?

Virtual team room

- Fixed location
- Everyone sees and hears each other
- Everyone sees the same screen
- Pass on keyboard
- Whiteboard



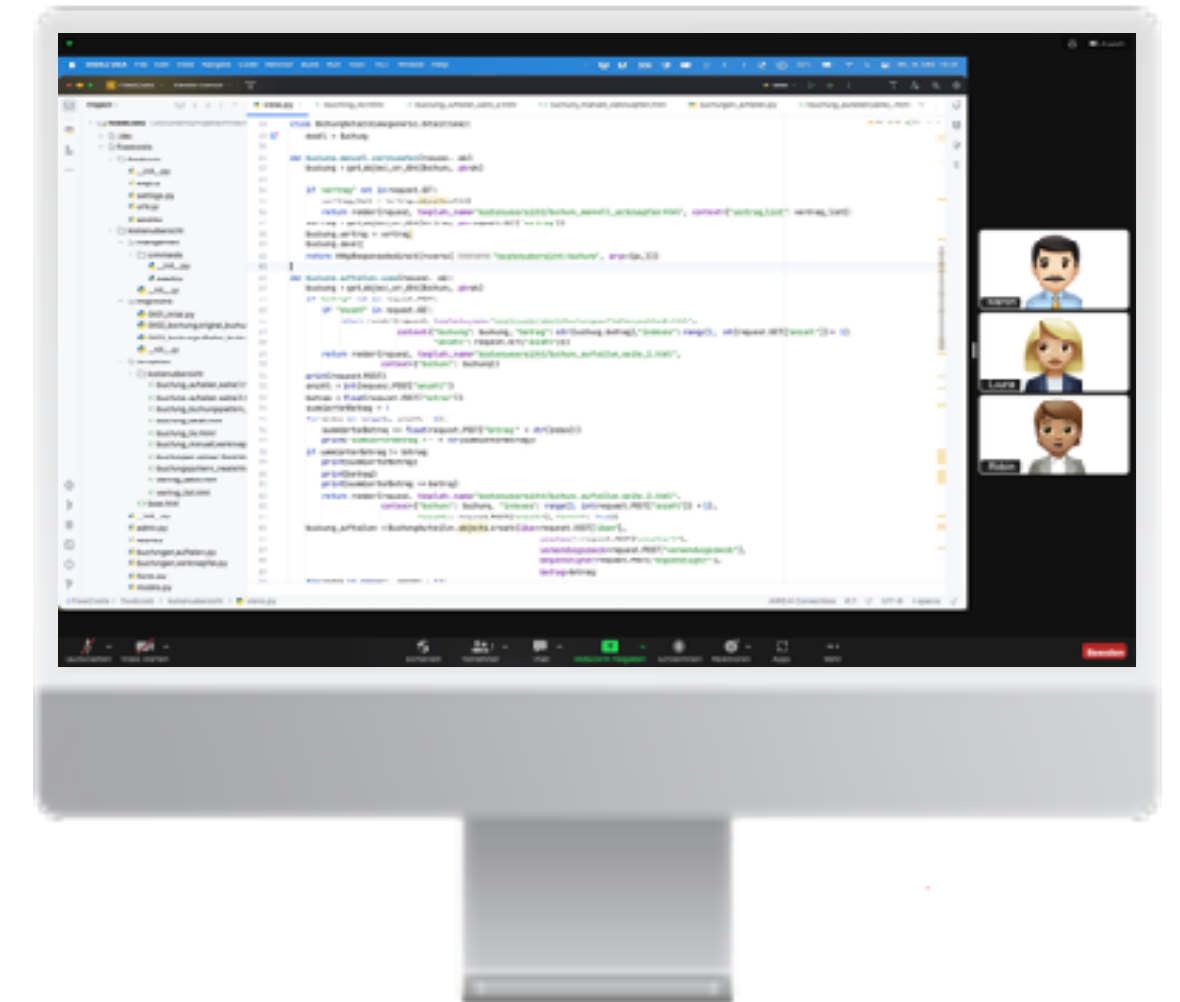
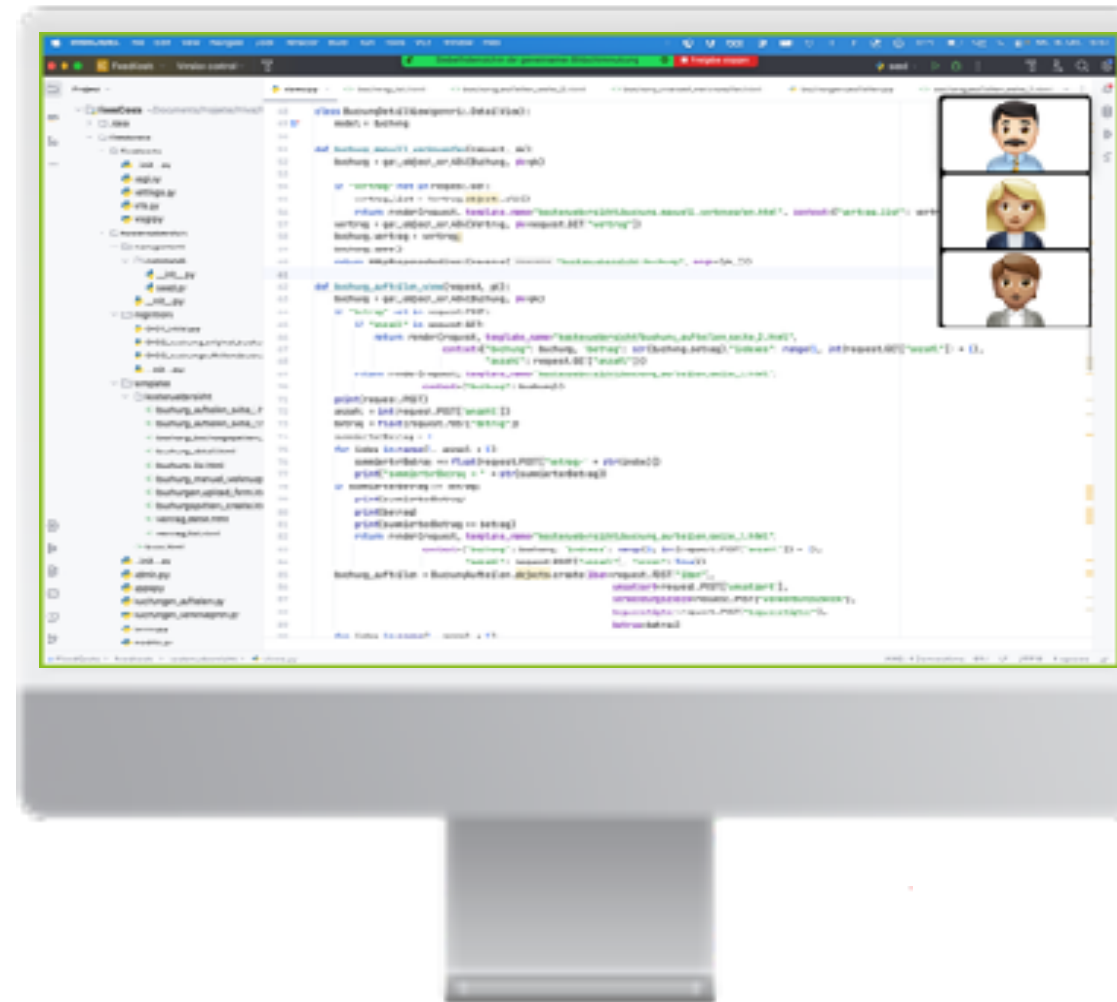
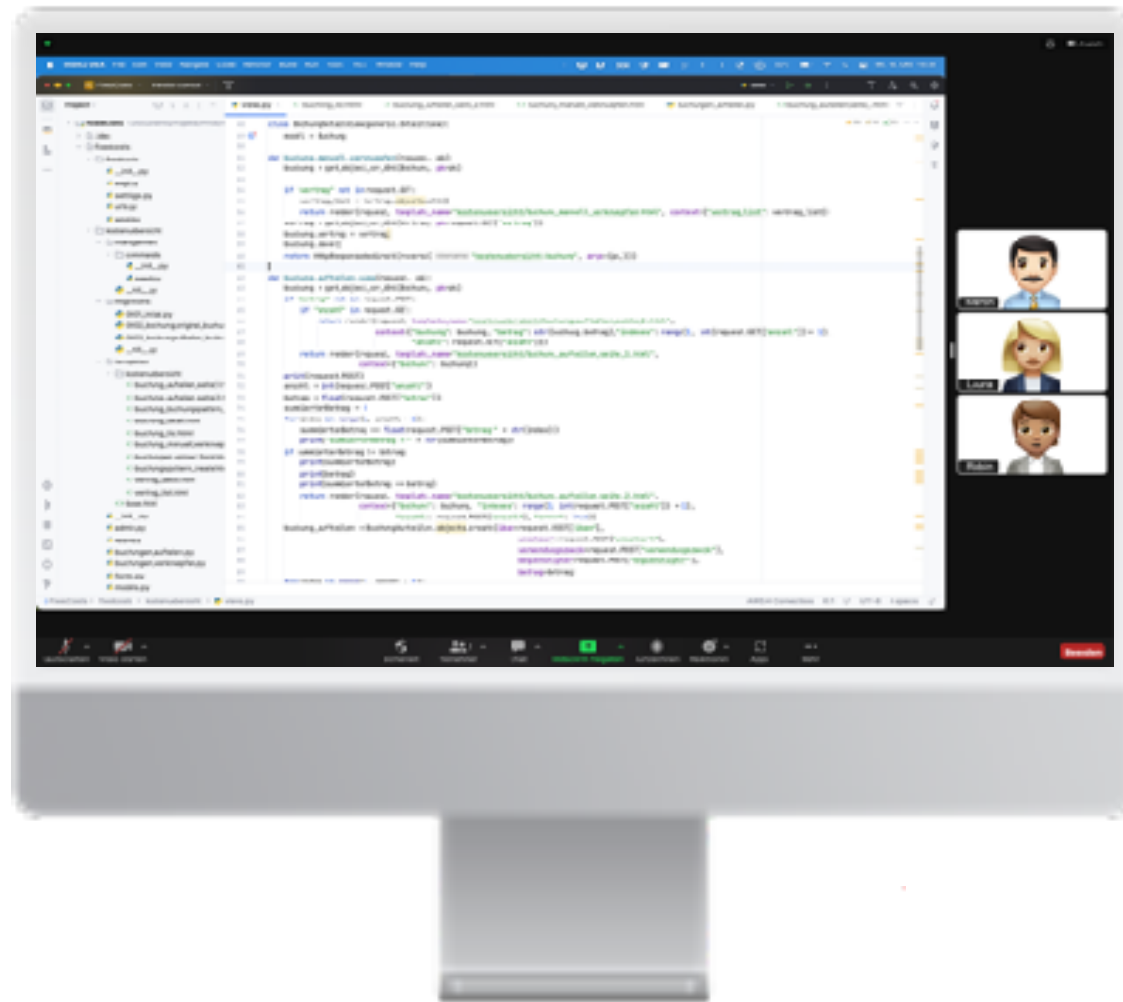


Typist

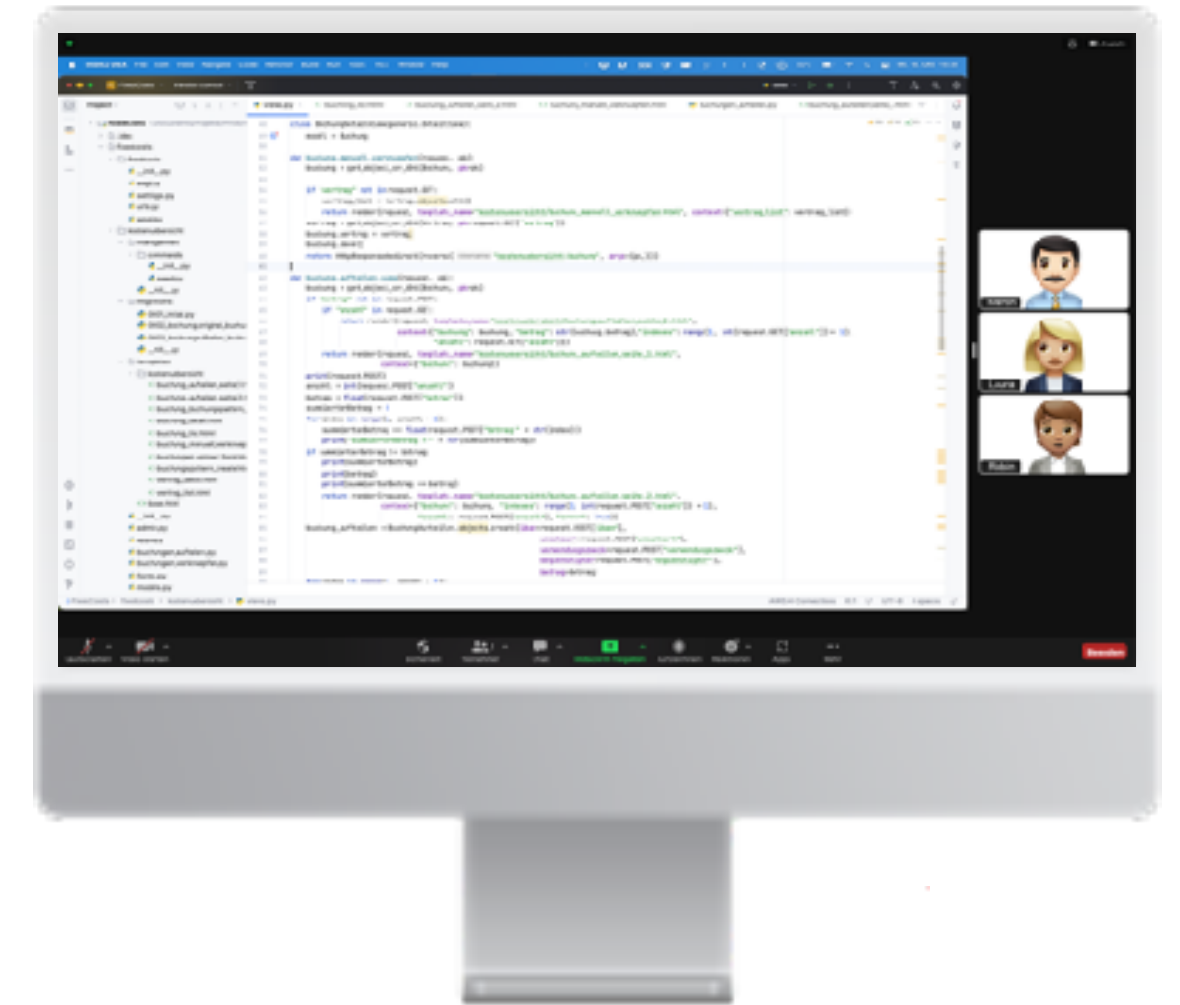
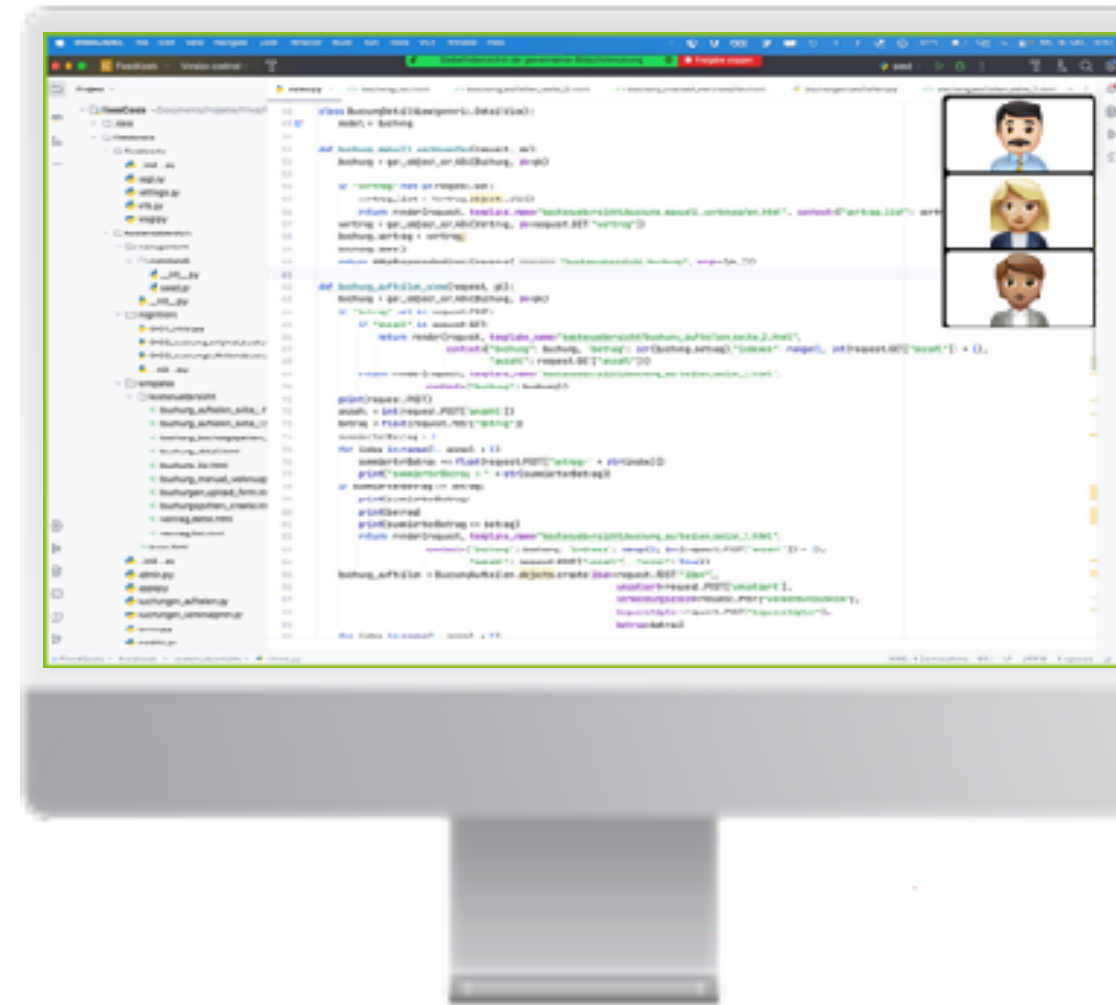
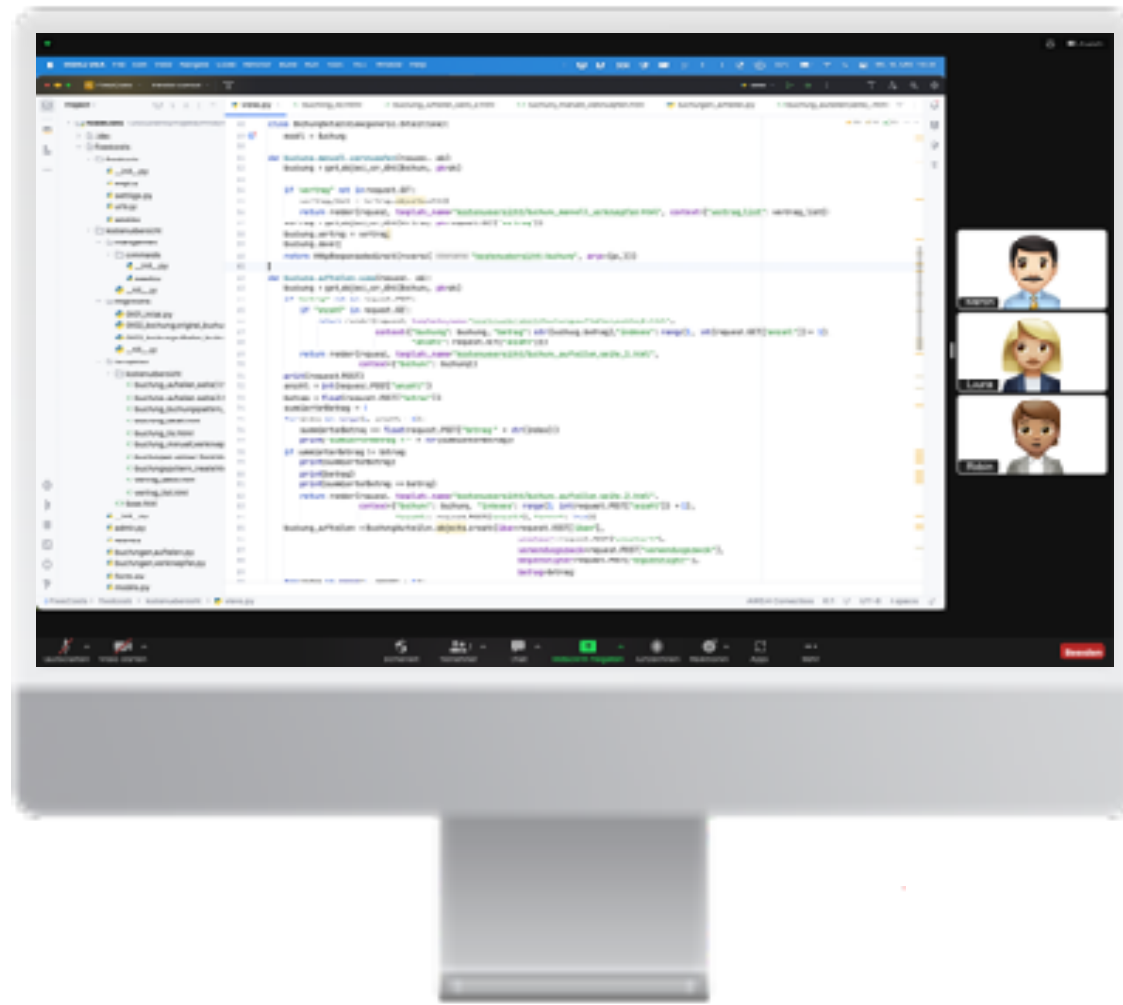


testrepo — joshuatopfer@Laptop-von-Joshua — ..NNOQ/testrepo — -zsh — 76x12

```
→ 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

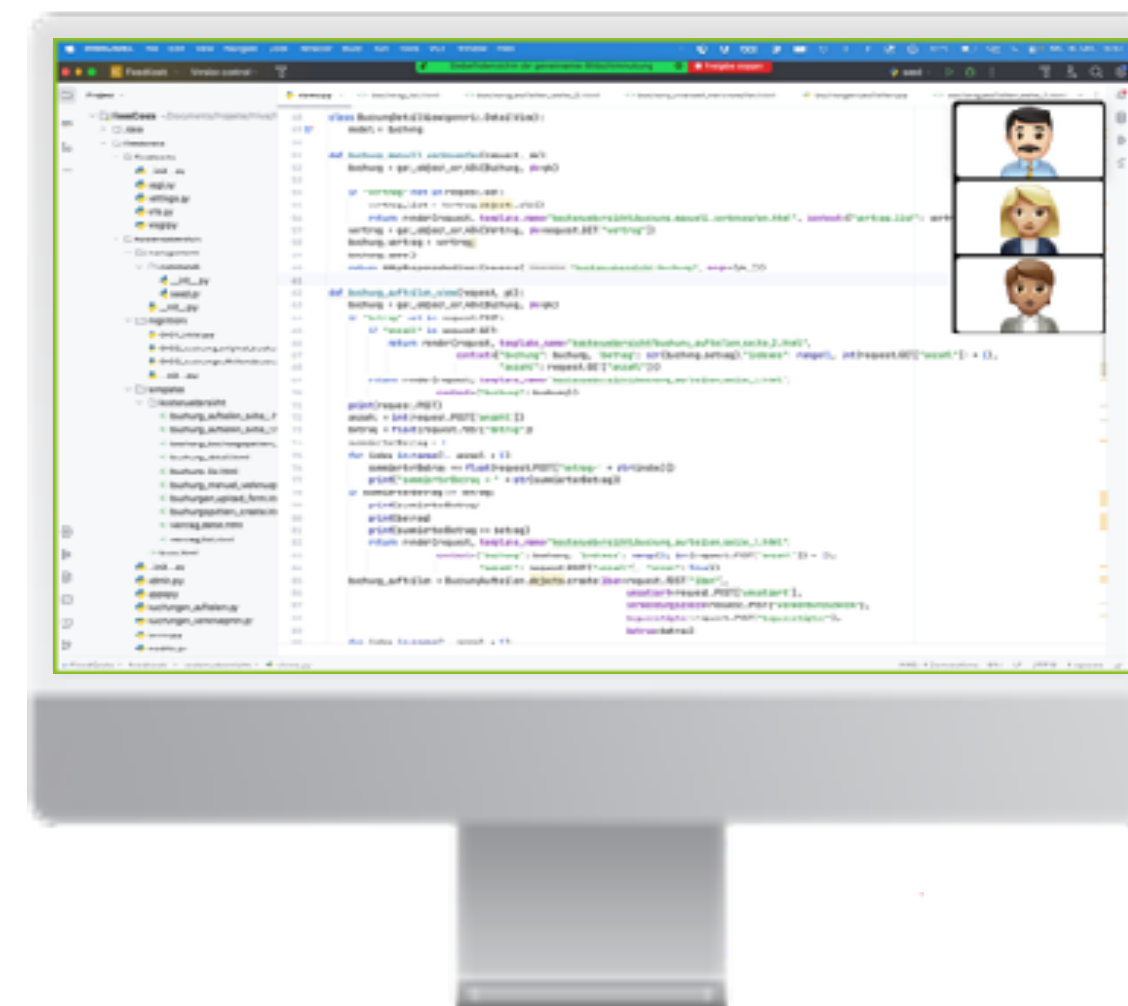
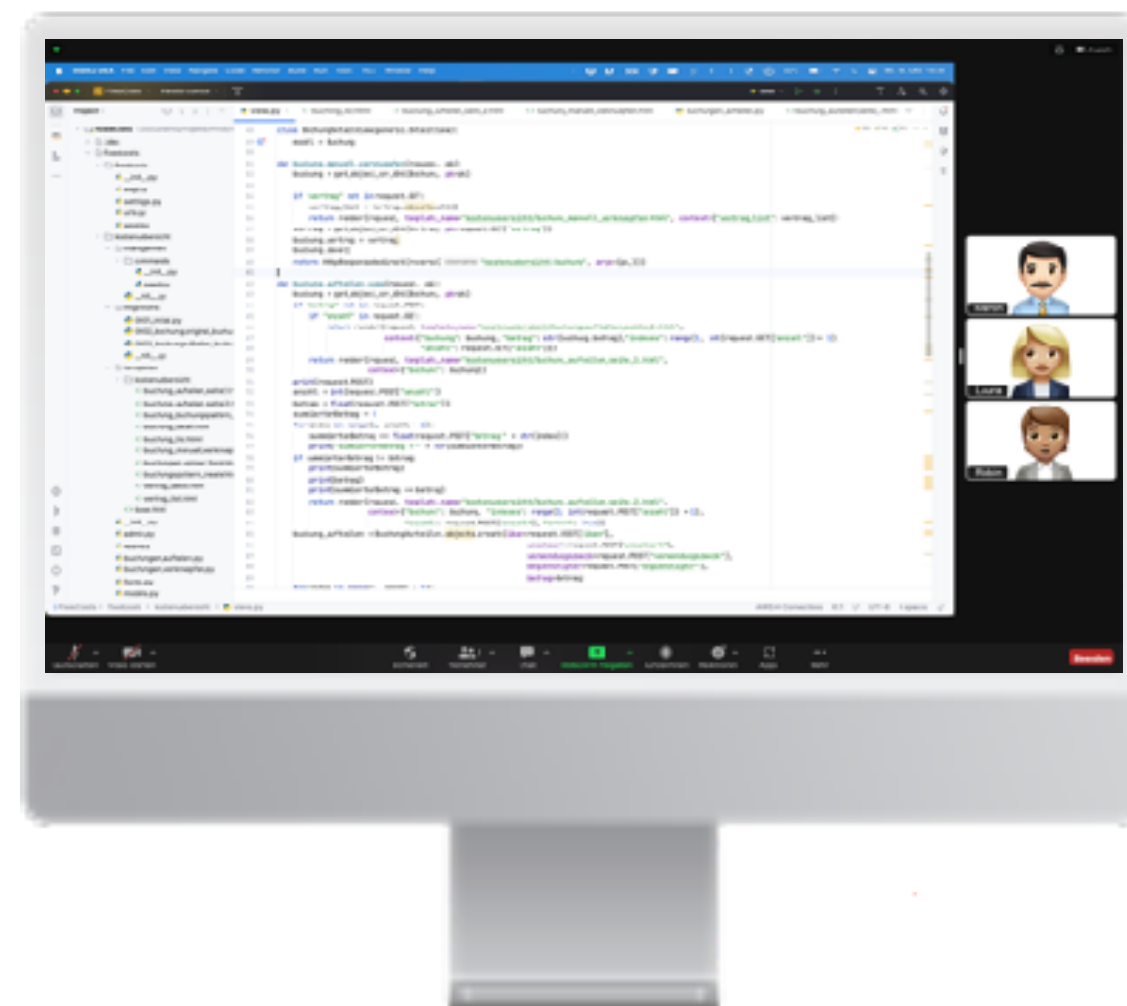
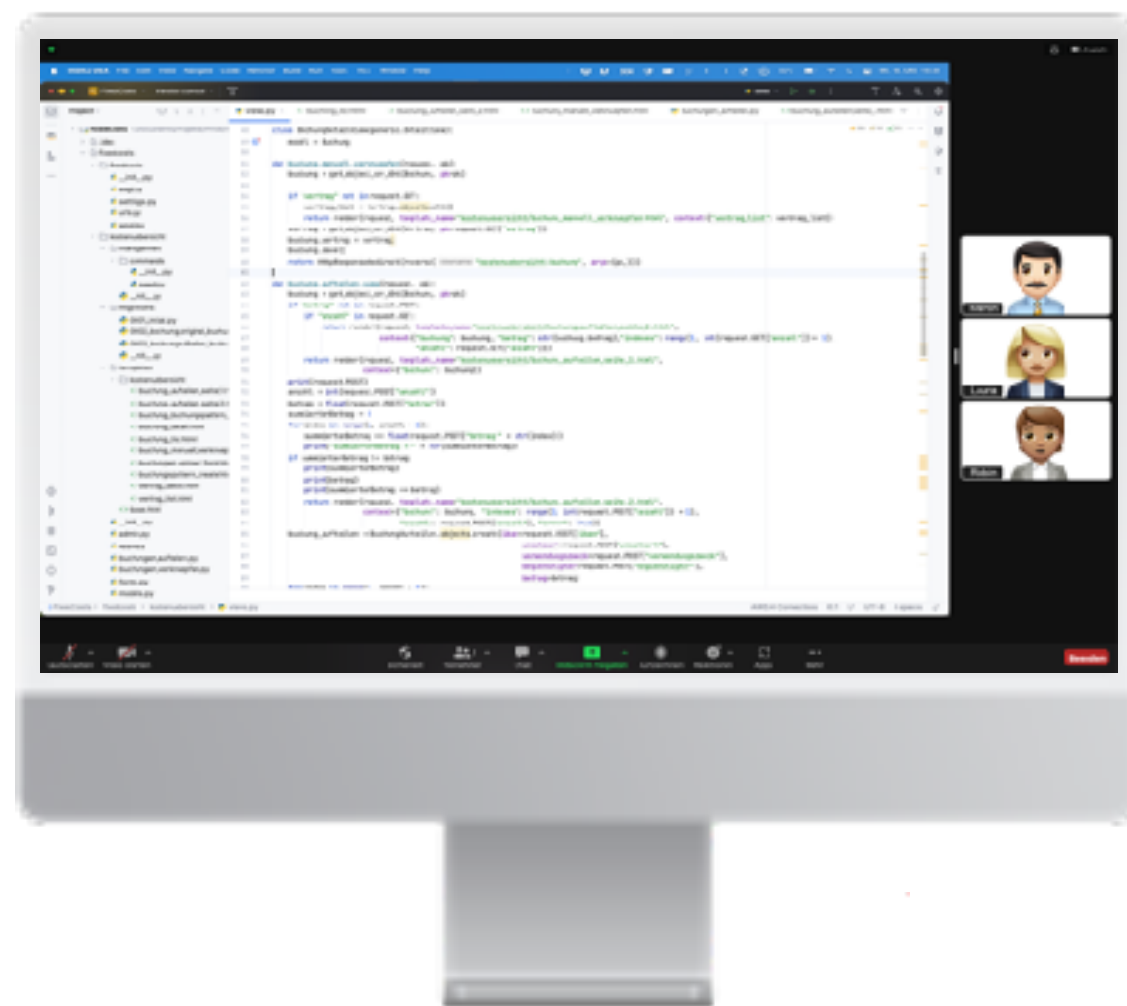


Typist

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

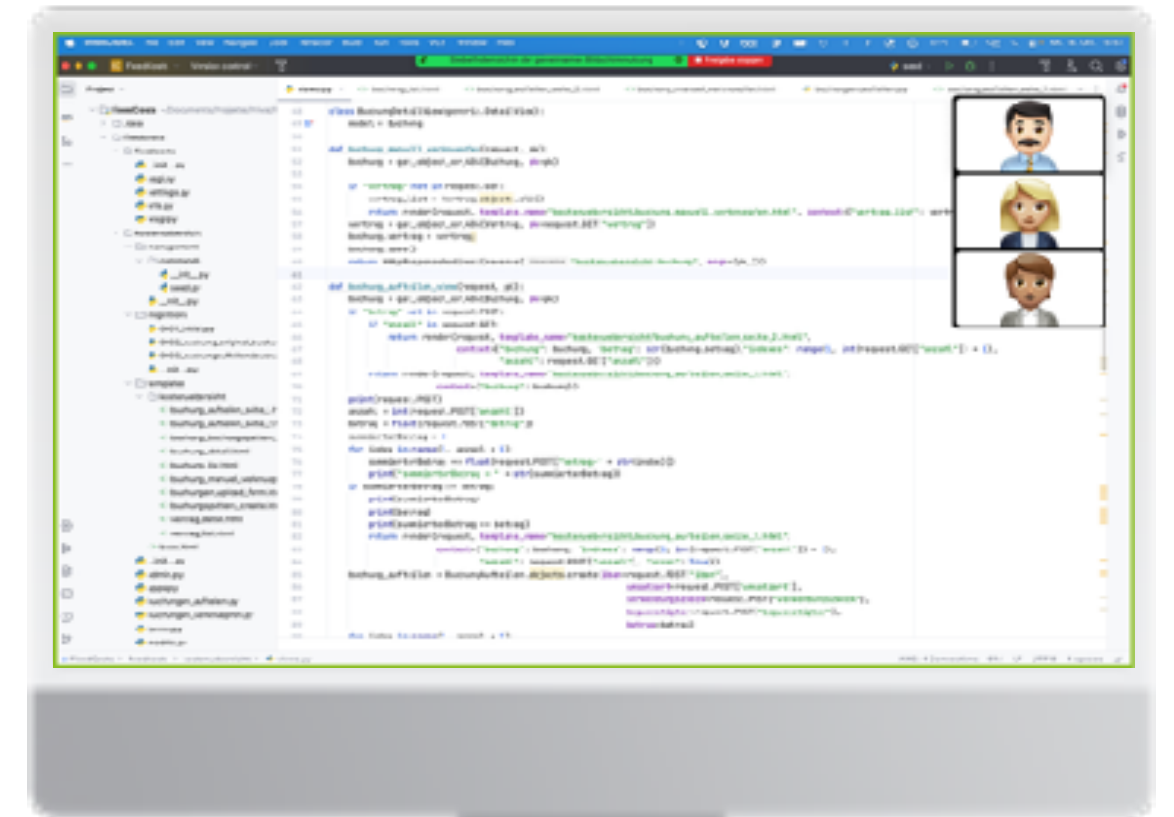
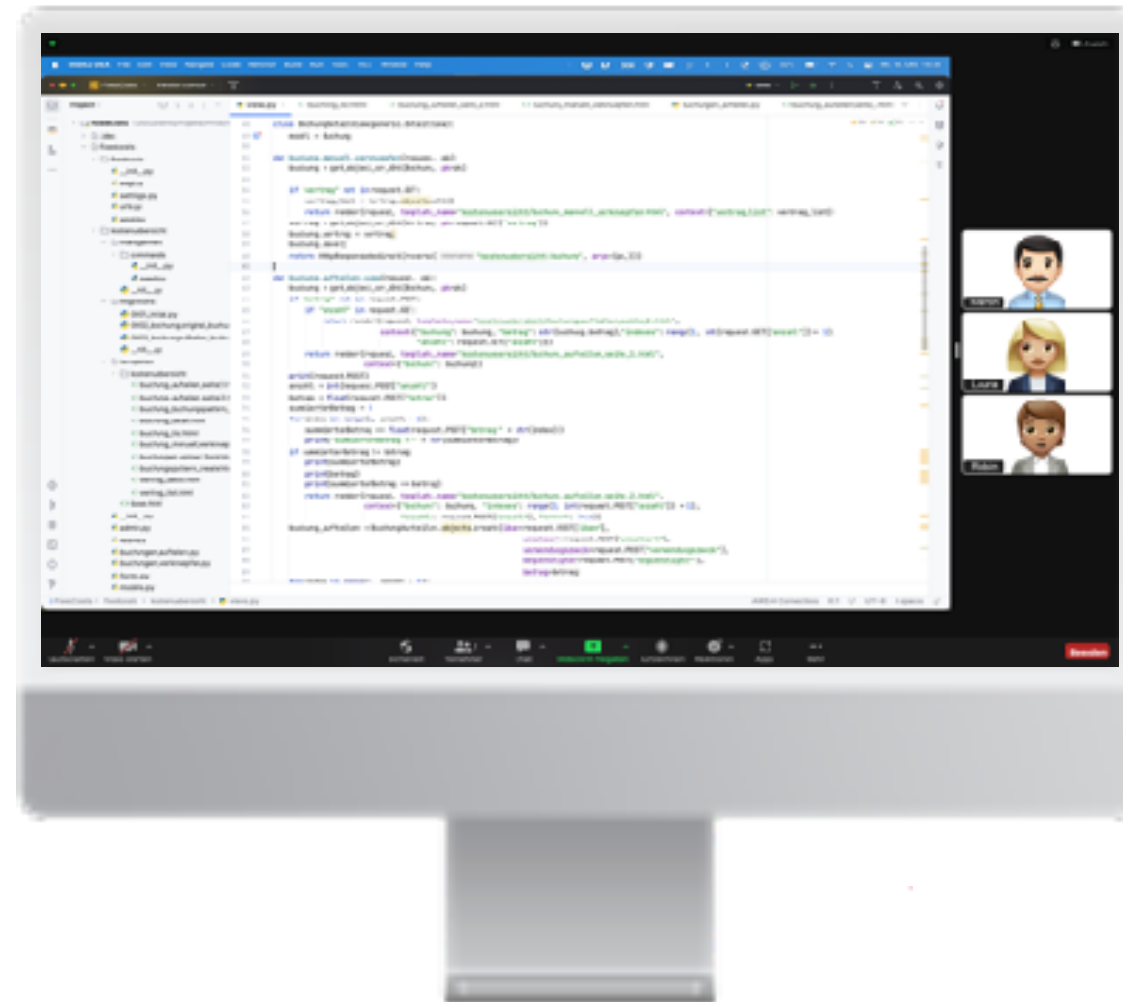
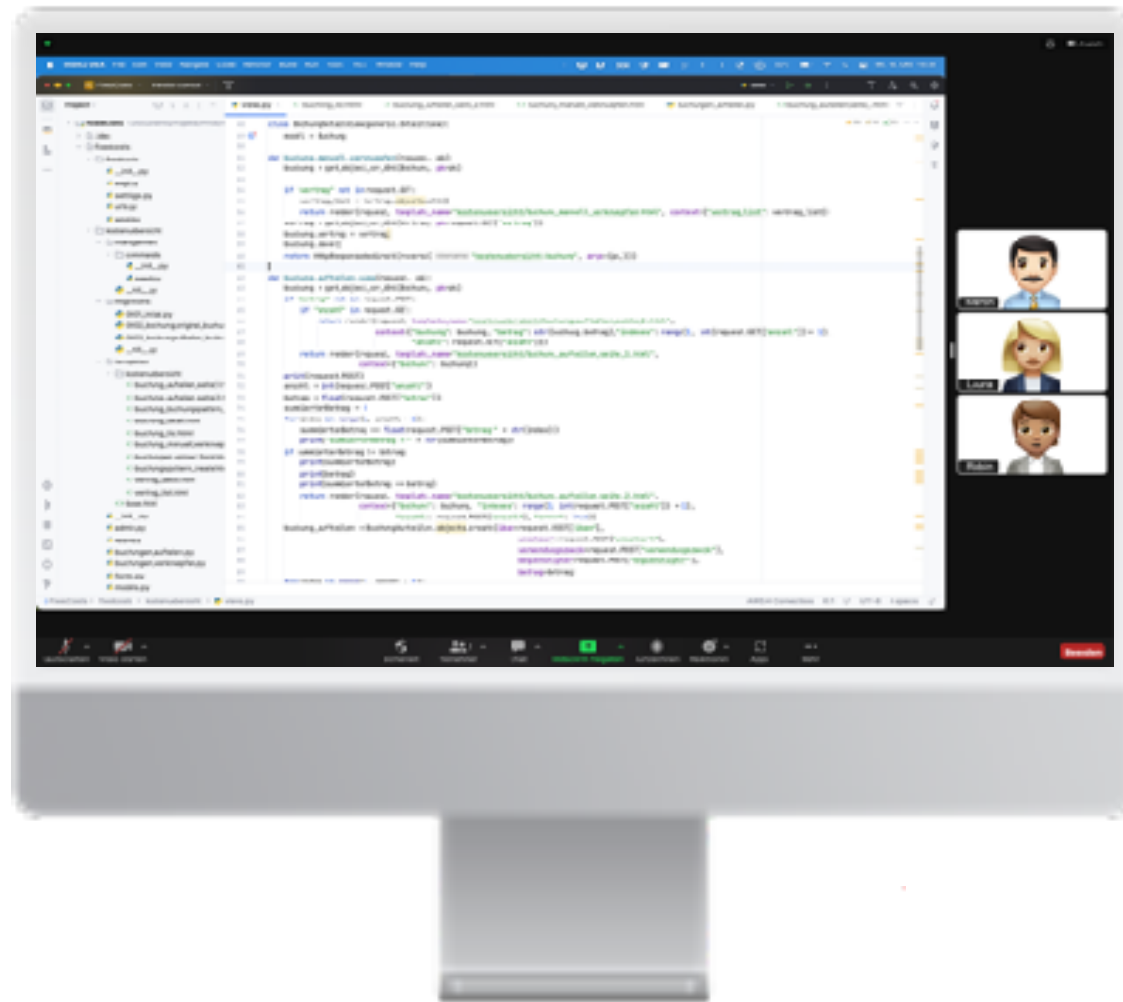
```
→ testrepo git:(mob/main) ✕ mob next
git add --all
git commit --message mob next [ci-skip] [ci skip]
[skip ci]

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) █
```

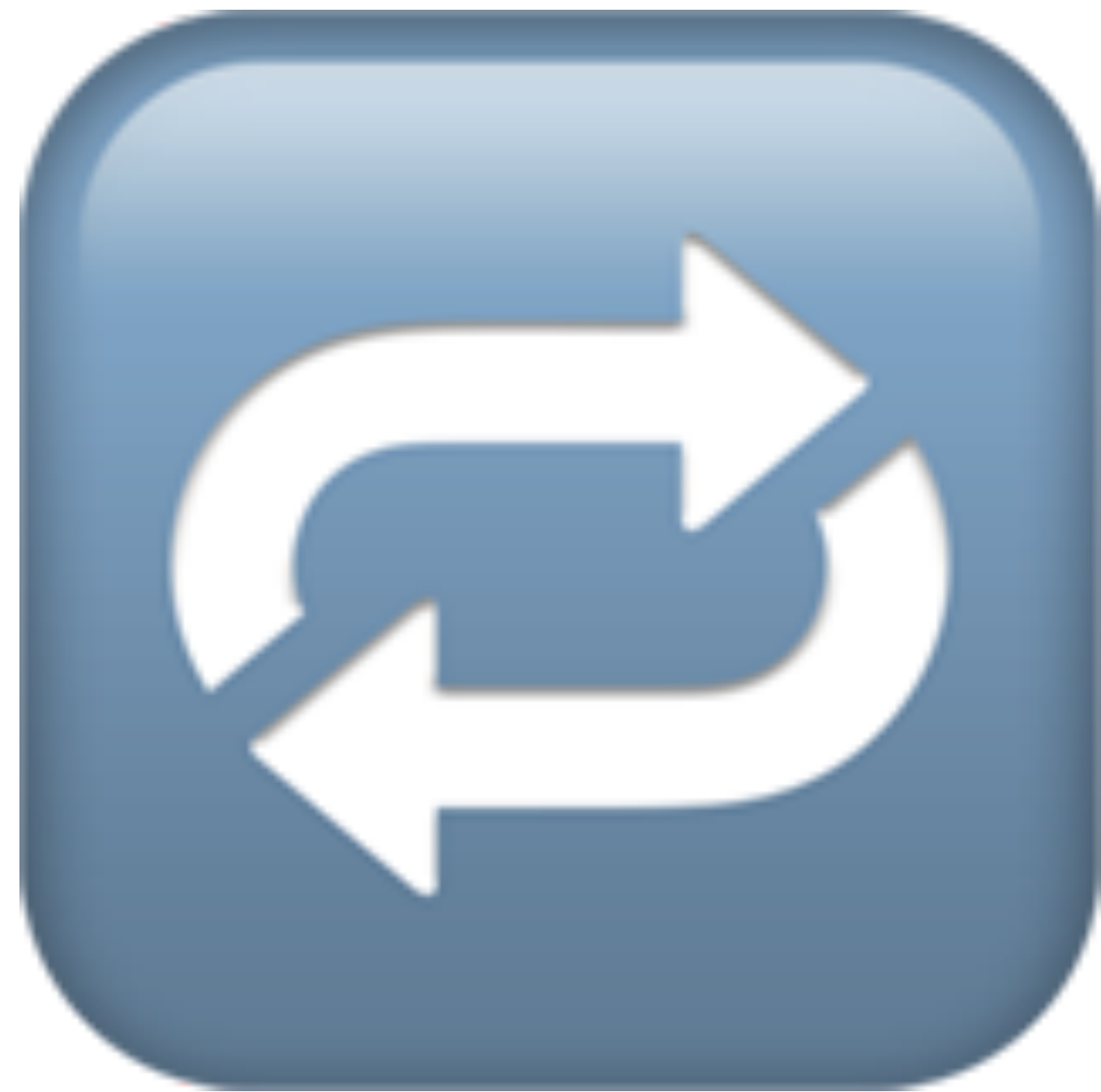



Typist


```
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) ✕

Why?

Time-to-market
instead of resource efficiency

Constant momentum

- Focus on the most important task
- Reduce context switching & waiting times
- No code reviews



Meetings

- Attending as whole team
- Eliminate unnecessary meetings
- If not possible just one team member attends
- Regular meetings like daily scrum are not necessary anymore



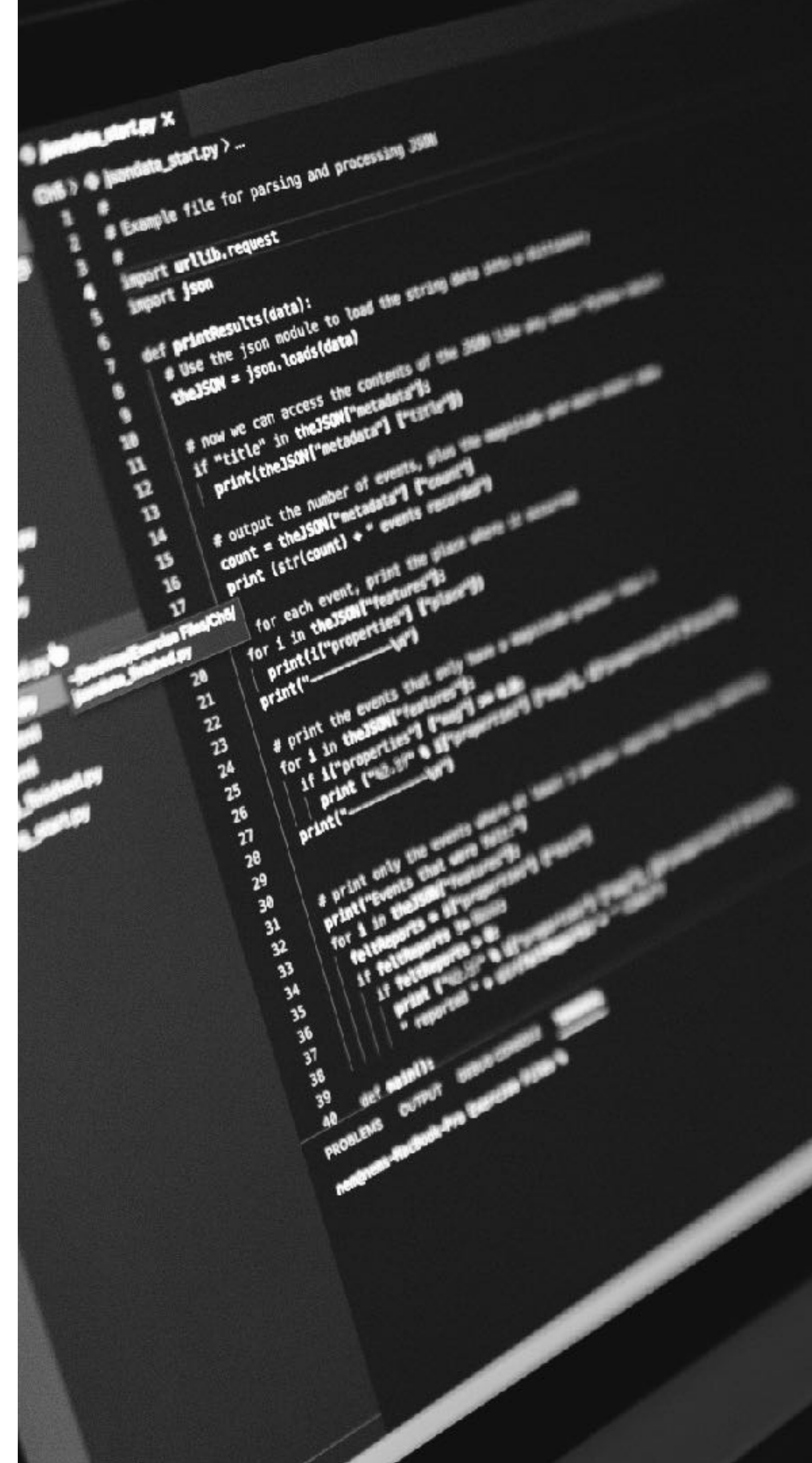
Architectural decisions

- Make architectural decisions explicit
- Team takes on the role of architect if required
- Create and discuss architecture decision records
- Consent instead of consensus



Code quality

- Many eyes
- Less bugs
- No asynchronous code reviews
- Refactor together



Learning from others

- Continuous learning
- Reduce bus factor
- Onboarding with no effort



At home, but not alone

- Lunch with the family
- Contact to colleagues
- No need to drive to the office

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



The *silver* bullet?

Different timezones

- Small time offset:
 - Hard to find common time slot
 - Core hours
 - Need for coordination
- Bigger time offset:
 - Just short mobbing session
 - 24/5 mob around the word



Conflicts within the team

- Conflicts quickly come to the surface
- Introduction must be accompanied by a coach
- Many small retrospectives at the beginning



Dependencies

- Waiting times
- Context switching
- Reduce dependencies
- Teams should have end to end ownership



Break down silos

- Share general knowledge within the team
- Very in-depth knowledge is difficult to share
- Bus factor is reduced, but not completely mitigated



Not suitable for everyone

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

Photo by Karolina Grabowska: <https://www.pexels.com/de-de/foto/frau-emotionen-tisch-ausbrennen-7272596/>



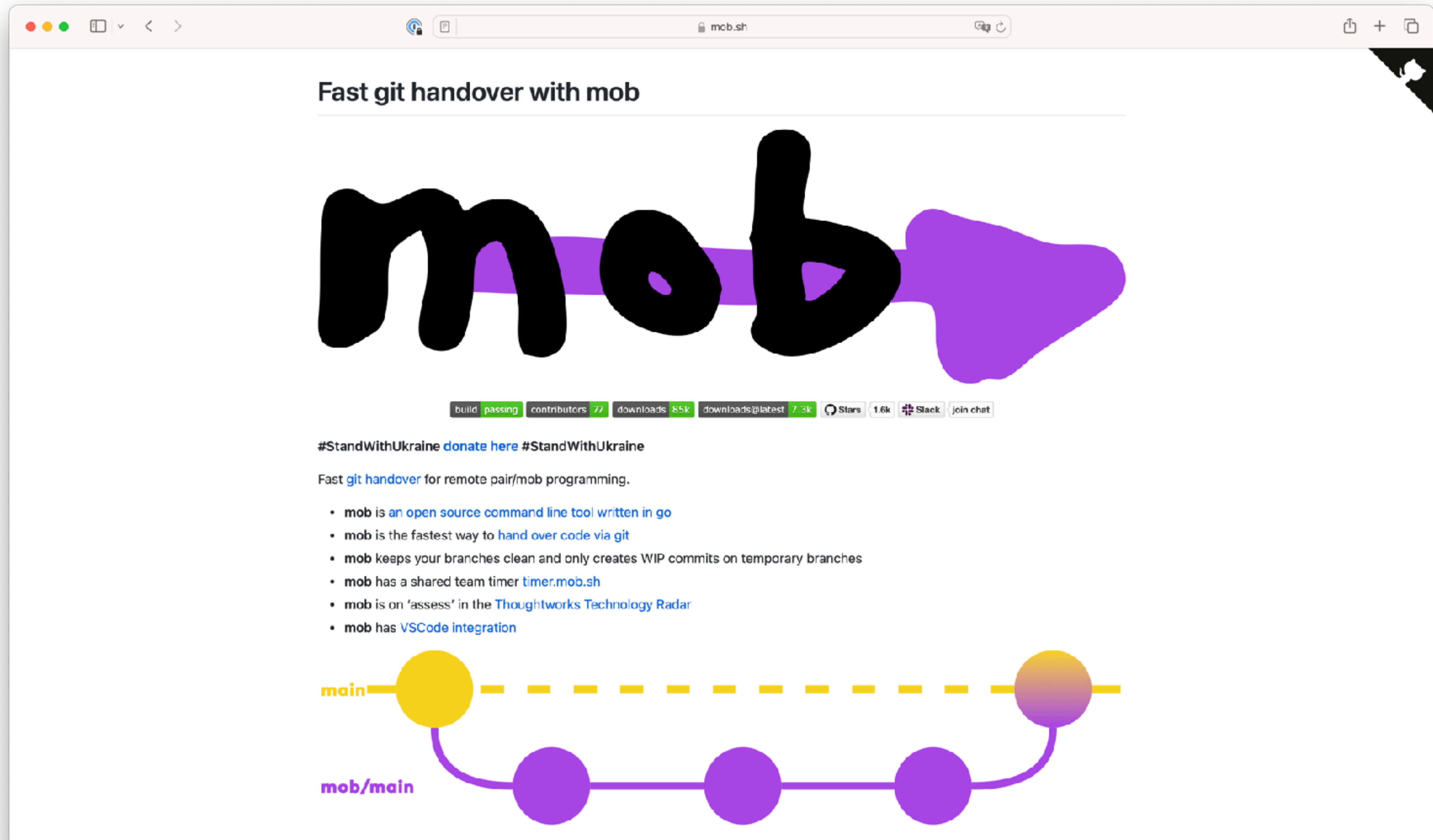
How to *start*?

Do an experiment

- Just try out a sprint full time
- Many retrospectives
- Accompany with a coach (I am happy to support)
- 1-day workshop for your team at [Socreatory](#) or try it together with strangers at mobusoperandi.com



mob.sh



The screenshot shows the mob.sh website in a browser window. The page has a light pink header with navigation icons and a GitHub logo in the top right. The main content area is white and features the title "Fast git handover with mob" in a bold, dark font. Below the title is a large, stylized logo for "mob" in black, with a purple arrow pointing to the right. Underneath the logo is a row of status badges: "build passing", "contributors 77", "downloads 85k", "downloads@latest 7.3k", "Stars 1.6k", "Slack", and "join chat". Below these badges is a line of text: "#StandWithUkraine [donate here](\"#\") #StandWithUkraine". This is followed by a paragraph: "Fast [git handover](\"#\") for remote pair/mob programming." and a bulleted list of features. At the bottom of the page is a diagram illustrating the workflow between a "main" branch (yellow circle) and a "mob/main" branch (purple circle). A dashed yellow line connects the two main circles, and a solid purple line connects the two mob/main circles. The diagram shows a sequence of commits on the mob/main branch, with the final commit being merged back into the main branch.

Fast git handover with mob

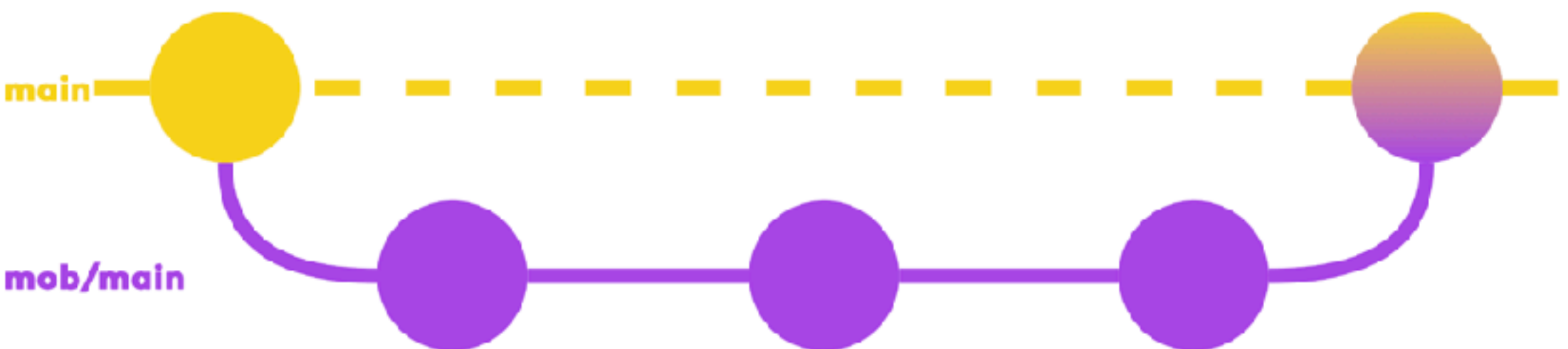
mob

build passing contributors 77 downloads 85k downloads@latest 7.3k Stars 1.6k Slack join chat

#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](#)



The diagram illustrates the mob workflow. It shows a horizontal sequence of circles. The first circle is yellow and labeled "main". A dashed yellow line connects it to a second yellow circle on the right. Below the "main" circle is a purple circle labeled "mob/main". A solid purple line connects the "main" circle to the "mob/main" circle. This line then continues through three more purple circles, representing a sequence of commits on the mob/main branch. Finally, a curved purple line connects the last purple circle back to the "mob/main" circle, indicating a merge or handover back to the main branch.

Our Primer



www.innoq.com/en/primer

Download for free

Jochen Christ
Simon Harrer
Martin Huber

Remote Mob Programming

At home, but not alone

INNOQ

Foreword by Mark Pearl

remotemobprogramming.org

Remote Mob Programming

New: [How to act as a typist](#) and a [rotation timer](#)

Camera Always On

Screen Sharing

Git Handover

10 Minutes Intervals

Small Team

Dine with your Family

Typist and the Rest of the Mob

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.

Conclusion

- More focus
- Learn continuously
- High performing team
- Colleagues became friends
- More time for the family



Thank you! Any questions?



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

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