



Projekt zespołowy

Wojciech Barczyński (Merito)

Wojciech Barczyński

- Tech Lead
- Software Developer -> System Engineer
- wojciech.barczynski@wroclaw.merito.pl

Best practices

1. Design docs,
2. UX/UI design,
3. Customer feedback,
4. Initial Planning and ETA,
5. Do work,
6. Working as a team,
7. Track progress + update every 1-2 weeks the plan,
8. Release checklists,
9. Alpha/Beta/RC/Release.

Best practices

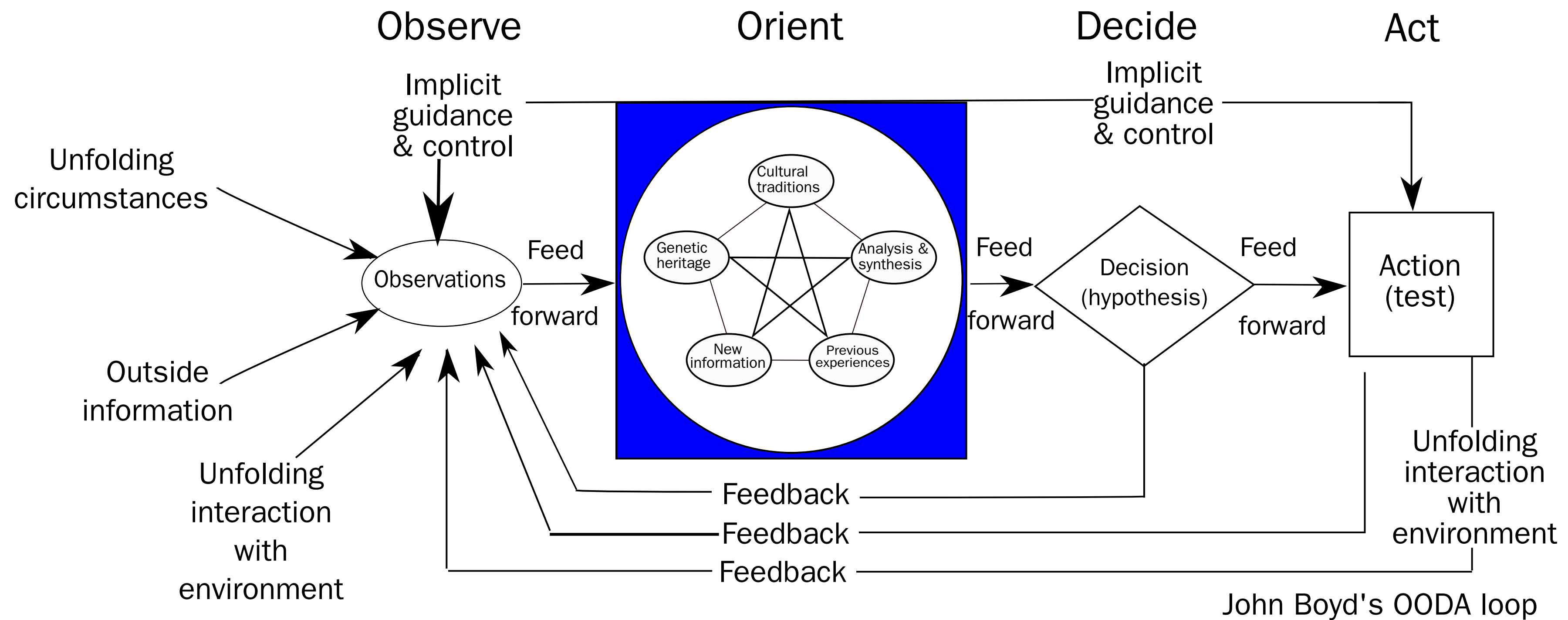
Everything is an experiment and iteration:

1. Pre-mortems,
2. Retrospective per project
(along with the team retros),
3. Keep and update project how-to / internal [wiki](#),
4. Post-mortems.

Later, with bigger team, see [svpg](#)

Best practices

OODA:



Engineering Culture

Strong foundation:

1. Safe environment (see [1](#) and [2](#)),
2. [Ownership](#),
3. Drum beat by the people who do the most of the work,
4. Pragmatism,
5. [Product teams](#)!

Engineering Culture

Quality of communication:

- 2x yes, and...;
- Strong opinion, weak held (read: [original post](#), not-a-silver-bullet);
- [number of f-given](#);
- trust / you do not need to be involved in all the discussions.

Consider

- Size of the team and the company,
- Experience of the engineers,
- Product maturity,
- etc.

0. Validate idea

Test/Validate the idea before writing code:

1. Mockups,
2. Code examples / API,
3. Share design.

You might also want to check Lean UX.

1. Design (engineering) docs

- With **DRI**,
- Thinking on paper,
- Opportunity to discuss and give feedback,
- **Example**,
- My favorite format: Google Docs and slack,
- A good place to track design decisions later.

tips: number of **f-given**, "yes, and"

2. UX/UI design

- for example, Figma,
- Good to setup what consitute a [good UX feedback](#).

3. Customer feedback

- Not only in the beginning, continuously reach out for feedback;
- Does not need to be a complex process (see [book](#));
- Design Doc, code example, ...,
- [Qualitative and Quantitative.](#)

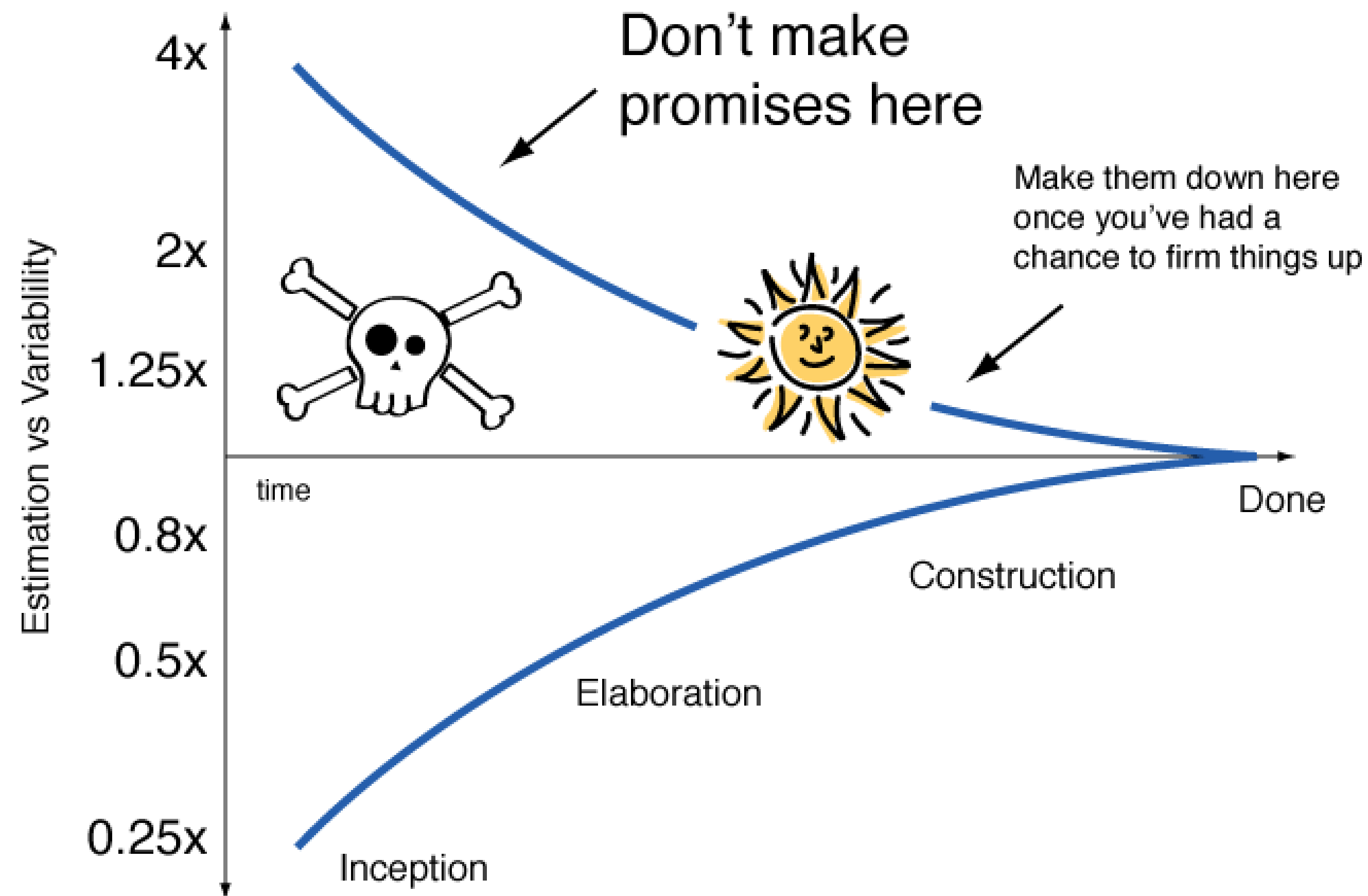
[a propos podejścia do klienta](#)

4. Initial plan

- Planning is everything;
- The first iteration is a sketch;
- ETA;
- The further into the future, the larger the unit of measurement.

4. Initial plan

Be aware / communicate:



4. Initial plan

- kilka scenariuszy kompletnosc/jakosc/szybkosc,
- Deliver -> patch/patch/patch / [Tracer Bullet Dev](#),
- [v1/v2](#).
- co musimy zwalidować najpierw / [MVP](#)
- Invest first: Spike albo PoC.

5. Do work

- 1 week perspective,
- Focus and cutting Work-In-Progress,
- Important project/feature -> more focus time + take the best engineers.

Kanban, Scrum

6. Work as a team

- Working pairs/programming,
- Swarming,
- Daily syncs,
- **Do-ocracy** / owning inputs.

7. Track progress update the plan

Every 2 weeks:

- demo session that every project stream presents what they done,
- update the plan.

7. Track progress update the plan

Every 2 weeks;

- Add new tickets/tasks;
- Update plan and the ETA for v1, v2...;
- Be honest with yourself.

7. Track progress update the plan

Live demo!

8. Release checklist

- getting the business on the same page;
- testing;
- pre-mortem if the rollout is complex;
- release details;
- metrics.

7. Release

- Feature flags,
- Alpha, (Closed) Beta, RC.

Misc

- komunikacja na publicznych kanałach (slack),
- dokumenty we wspólnych GDrives,
- wcześnie otwieranie PRa w draft.

Metodyka?

- Kanban z tygodniowymi cyklami według prokanban.org - mój domyślny tryb pracy,
- **SCRUM** z 1 tyg lub 2 tyg cyklami.

Ciekawym jest również koncept [flight levels](#)

QUESTIONS?



Nasz projekt

- Wstęp wstępem, co z projektem
- ... i dobrą ocenę.

Wasz projekt

Cel

1. Zabawa,
2. Praca w zespole,
3. Eksperymentowanie,
4. Szlifowanie swoich umiejętności.

Temat

1. Złożona aplikacja

2. ...

3. ...

Następne kroki

1. Zespół 2 do 3 osób,
2. Utworzyć repozytorium na githubie,
3. `README.md`,
4. `docs/`.

README.md

- Cel projektu, **why, what, how**
- (to jest pierwsza iteracja)

Następne kroki

Repozytorium:

- `docs` - all docs (Markdown) and diagrams
- `docs/plan.md` (`example`) - project tracker and updates

Następne kroki

Praca w zespole:

- discord / slack / teams;
- syncs (done, todo, blockers? ideas?);
- jeśli chcecie można się pokusić o github project czy ClickUp.

Ważne

Każdy z członków zespołów:

- mieć commity w repozytorium;
- być autorem PRów.

Ocena

- Czy eksperymentowaliście, bawiliście się konwencją;
- Złożoność aplikacji;
- Repozytorium z kontrybucjami;
- Live demo na ostatnim spotkaniu.

Pierwsze zadanie

Przestać emaila na

`wojciech.barczynski@wroclaw.merito.pl`

z:

1. Tytuł: **nazwa grupy**: projekt zespołowy` ,
2. Link to Repozytorium githuba (z `README.md`),
3. Członkowie zespołu.

Najlepiej dzisiaj. Jeśli repo jest prywatne, proszę
zaprosić `wojciech11`.

Rekomendacje

- git workflow oparty o *master/main* (aka github workflow),
- krótko żyjące PRy.

Patrz: [praca w zespole z gitem](#)

Narzędzia

Diagramy:

- draw.io / [yEd](#)
- [excalidraw](#) or [miro](#) for discussions

Thank you

Questions?

Backup

MISC

Facilitate growth:

- Ed Batista - the art of self coaching,
- yes, and...,
- masterclass,
- or other initiatives.