Team projects - best practices

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. Inital 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.

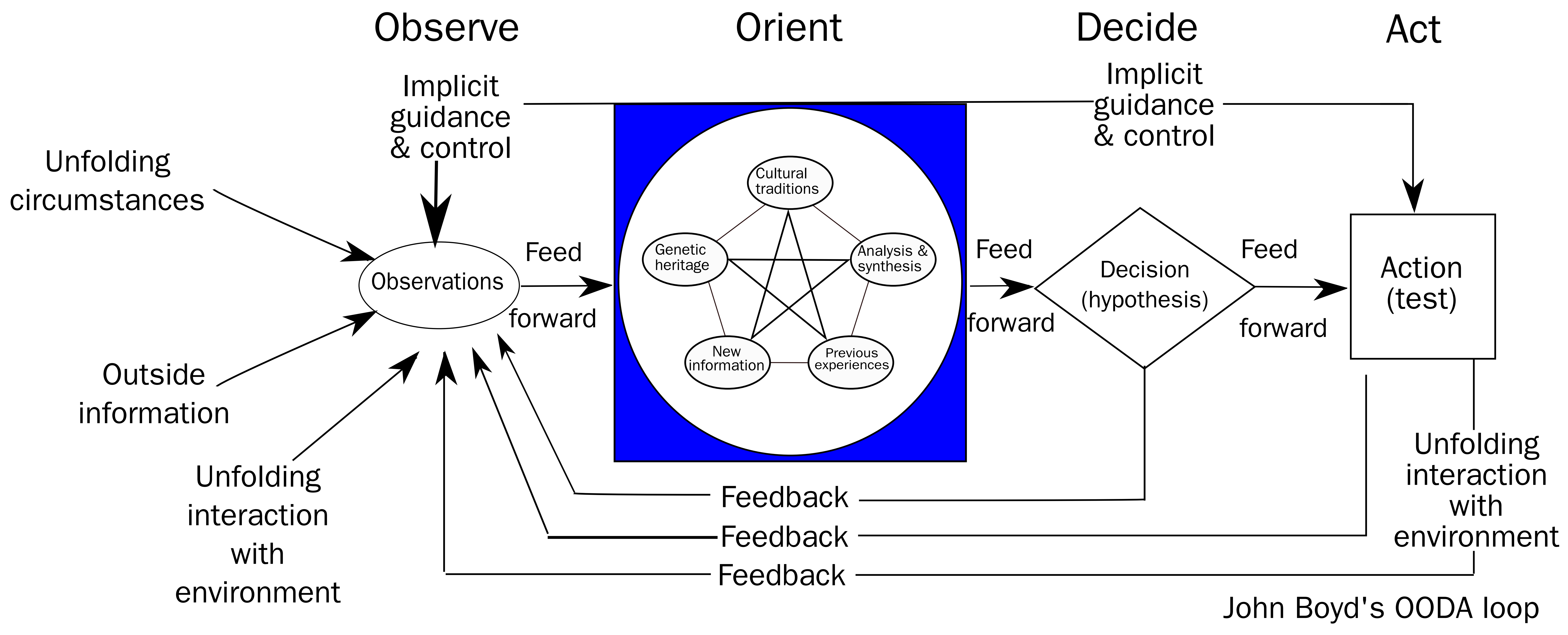
Everything is an experiment and iteration

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:



John Boyd's OODA loop

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 decions later.

2. UX/UI design

- for example, Figma,
- Good to setup what consitute a good UX feedback.

3. Customer feedback

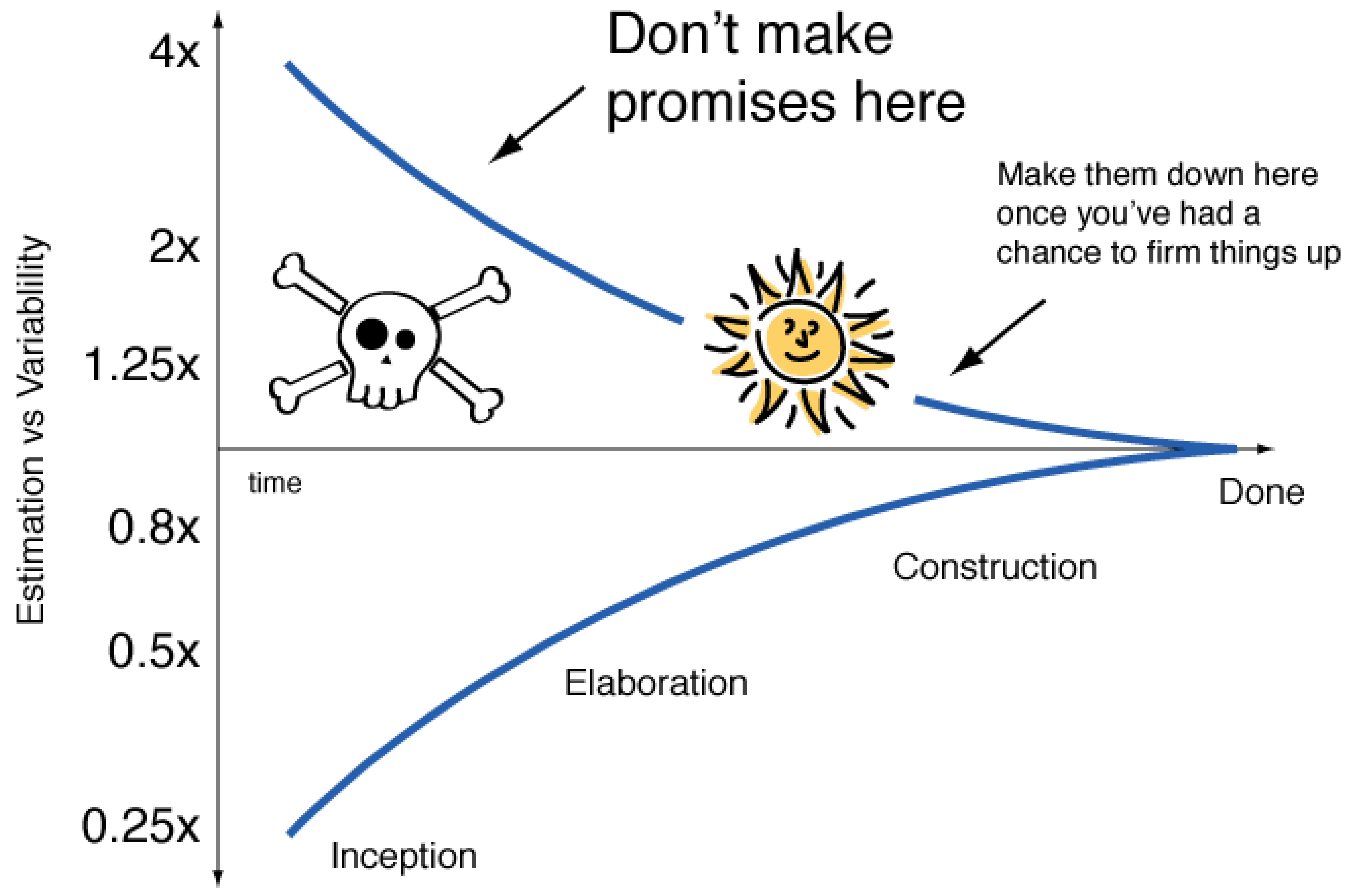
- Not only in the beginning, continuously reach out for feedback;
- Does not need to be a complex process (see book);
- Design Doc, code example, ...;
- Qualitative and Quantitative;
- Check tipics from YC.

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:



The Quirks of Software Effort Estimation

4. Initial plan

- Lay few alternatives scenarios out - completness/quality/**speed**;
- Deliver -> patch/patch/patch / Tracer Bullet Dev;
- v1/v2.

4. Initial plan

- what we need first to learn, risk to address, key value to deliver;
- MVP?
- Uncharted territory? Spike or 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

1. Daily syncs (driven by the team);
2. Working in pairs+ or pair programming;
3. Swarming;
4. Do-ocracy / owning inputs.

6. Work as a team

- Blocked or struggle -> let the team know.

7. Track progress & update the plan

Every 2 weeks:

1. Demo session that every project stream presents what they done,
2. Update the plan.

7. Track progress & update the plan

Demo:

- the best way to present the progress;
- Sync;
- Drum-beat;
- everything is demoable!

7. Track progress update the plan

Every 2 weeks;

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

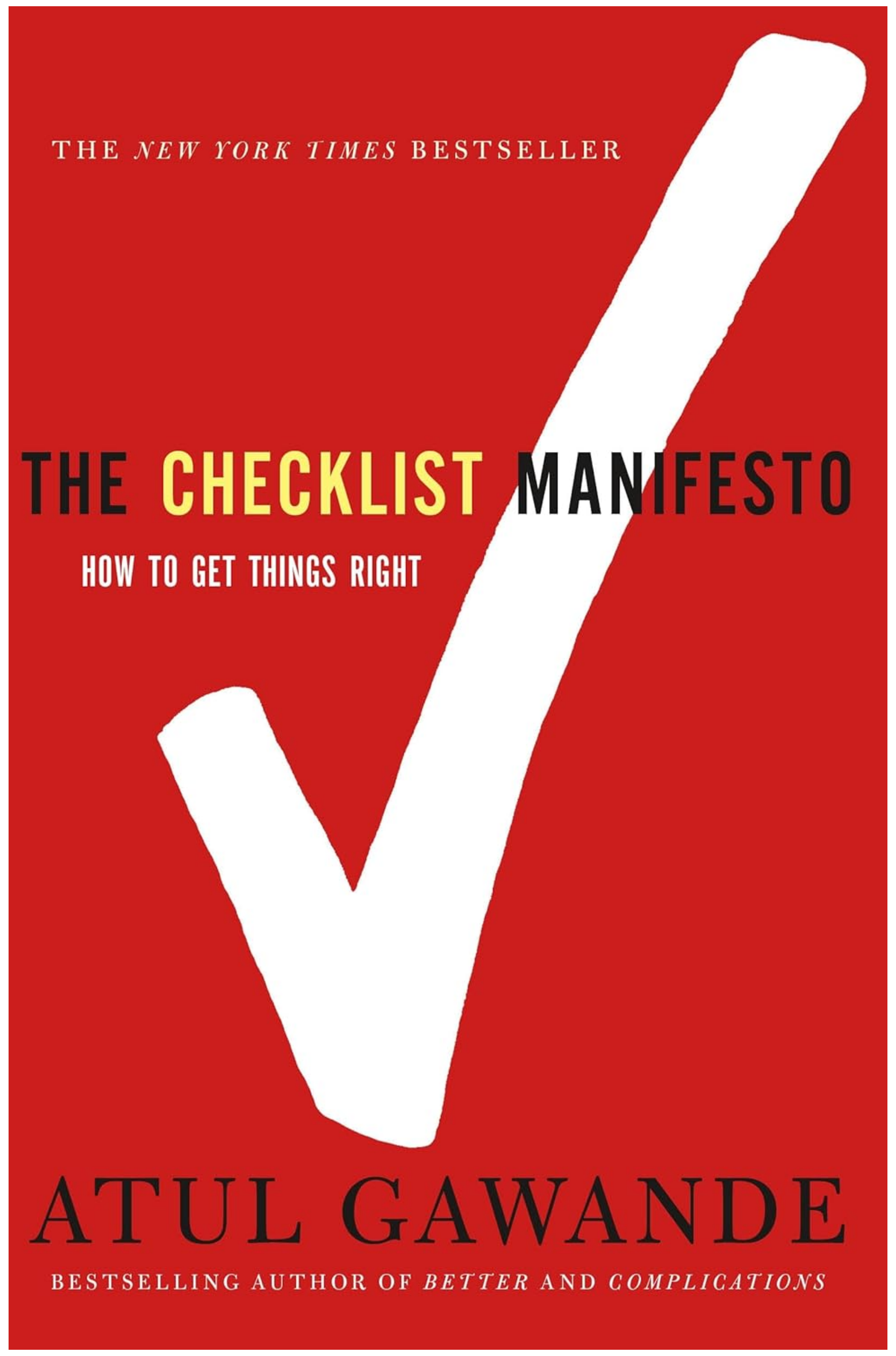
7. Track progress update the plan

Live demo!

8. Release checklist

- Getting the business on the same page;
- Testing;
- Pre-mortem if the rollout is complex;
- Release details;
- Metrics.

8. Release checklists


THE NEW YORK TIMES BESTSELLER
THE CHECKLIST MANIFESTO
HOW TO GET THINGS RIGHT
ATUL GAWANDE
BESTSELLING AUTHOR OF BETTER AND COMPLICATIONS

- Recording know-how;
- Keep quality;
- Do not make unneccessary mistakes..
- ...

7. Release

- Feature flags (recommended),

7. Release

Remember, you have many options to release your software to manage customers' expectations:

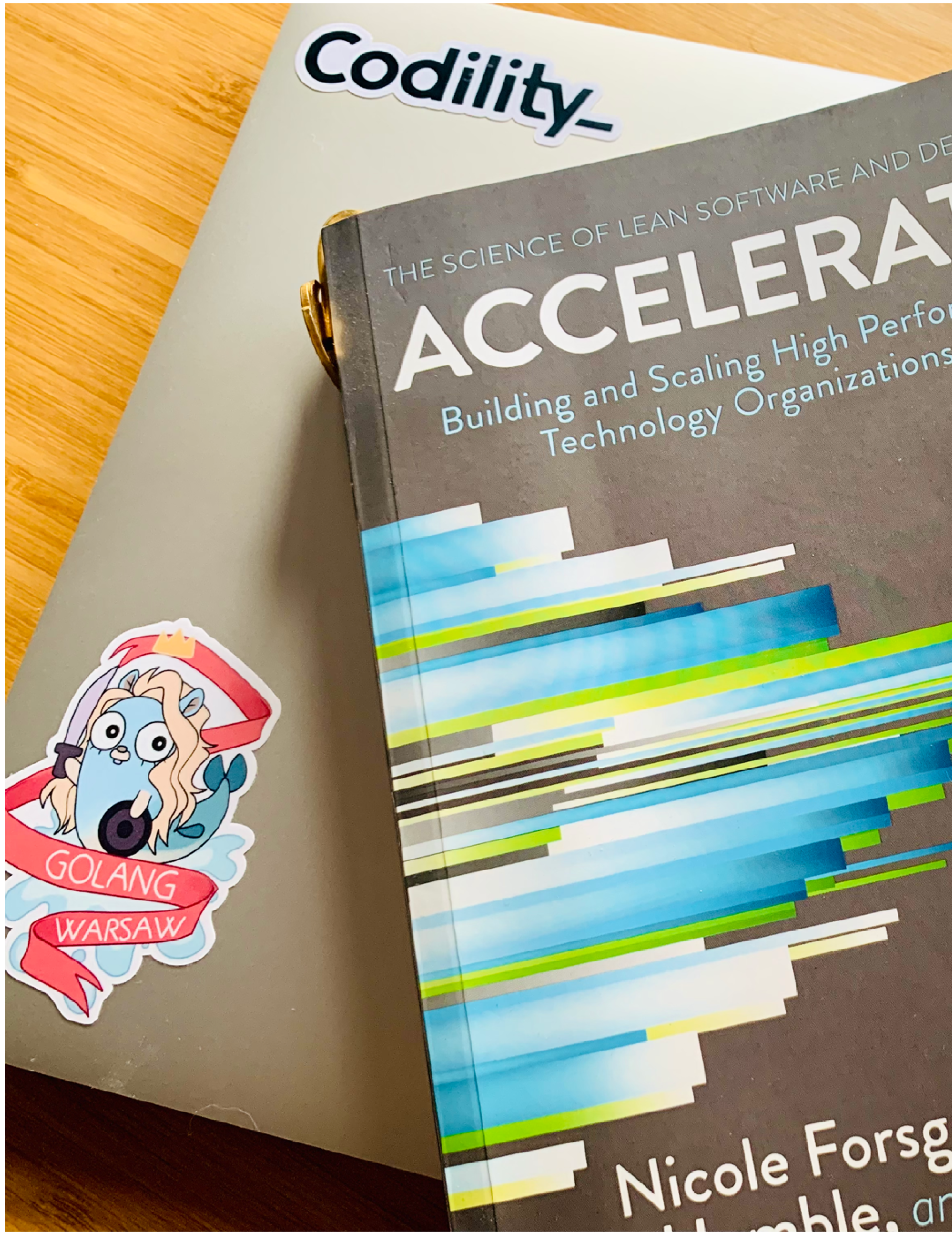
- Alpha, (Closed) Beta, RC.

8. Metrics

- Start with DORA;
- Deployent frequency and lead time.

High delivery performace

- Lead Time
- Deployment frequency
- Mean time to Recovery
- Change Fail Percent


Codility_
THE SCIENCE OF LEAN SOFTWARE AND DE
ACCELERAT
Building and Scaling High Perfo
Technology Organizations
Nicole Forsg
GOLANG
WARSAW

9. Transparency and communication

- communicate only on public channels (slack),
- documents only on shared GDrives,
- starting work? Open a PR in draft.

Metodyka?

- Kanban with weekly cycles following prokanban.org - my preference,
- SCRUM with 1-week or 2-weeks cycles.

For larger companies, the concept of flight levels is worth exploring.

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

Przesłać 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`.

Rekomentacje

- 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.