

Projekt zespołowy

WOJCIECH BARCZYŃSKI

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

Najlepsze praktyki

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

Najlepsze praktyki

Iteracyjne poprawianie prowadzenia złożonych projektów:

1. Pre-mortem,
2. Retrospective per project
(along with the team retros),
3. Everything is an experiment and iteration.

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

Warto wziąć pod uwagę

- Rozmiar zespołu i firmy,
- Doświadczenie inżynierów,
- Etap rozwoju produktu,
- itp. itd.

0. Validate idea

Warto przetestować pomysł przed napisaniem linijki kodu:

1. Mockupy,
2. Przykłady kodu / API,
3. Design nowego feature-u.

Większy zespół, warto spojrzeć na Lean UX.

0. Engineering Culture

1. Safe environment,
2. **Ownership**,
3. Drum beat by the people who do the most of the work,
4. Pragmatism,
5. Product teams!

1. Design (engineering) docs

- 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: [DRI](#), 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

- Nie musi to być odrazu bardzo złożony process na początku,
- Design Doc, fragment kodu, ...,
- Qualitative and Quantitative,
- Nie tylko na samym początku.

4. Initail plan

Plan:

- Planning is everyting,
- The fist iteration is a sketch,
- ETA,
- Czy dalej w przyszłość, tym większa jednostka miary.

4. Initail plan

- kilka scenariuszy kompletnosc/jakosc/szybkosc
- v1/v2
- Deliver -> patch/patch/patch
- co musimy zwalidować najpierw / MVP
- Invest first in a Spike? PoC

5. Do work

- 1 week perspective
- Focus and cutting Work-In-Progress
- Important project/feature -> more time/best engineers

Kanban, Scrum

6. Work as a team

- Pair working/programming
- Daily syncs
- Do-ocracy / owning inputs

7. Track progress update the plan

- Every 2 weeks, update the plan,
- Add new tickets/taks,
- Update plan and the ETA for v1, v2...,
- Be honest with yourself.

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.

Closing thoughts

- obawa przed estymatami,
- czym lepsi i zaangażowany zespół, tym łatwiej,
- sposobem na wyladowanie projektu:
 - focus
 - otwarta komunikacja
 - dobrzy inżynierowie

Nasz projekt

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

Temat

1. Złożona aplikacja

2. ...

3. ...

Następne kroki

1. Utwórz zespół 2 do 3 osób,
2. Repozytorium githuba, przesłać je emailem
`wojciech.barczynski@wsb.wroclaw.pl`,
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
- for small tasks, you can use the github project

Następne kroki

`docs/plan.md` - tracker:

- co 2 tygodnie update estymacji

Następne kroki

(Opcjonalnie) Zadania w github project.

Następne kroki

- typy pair programming

Narzędzia

Diagramy:

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