

Techniczny projekt nowatorski



Wojciech Barczyński

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

Working agreement

I will - TBA

Working agreement

I ask you to - TBA

Plan

3 semestry:

1. MVP aplikacji + temat i spis treści,
2. Aplikacja w End2End - happy path (tracer bullet) + draft część pisemna,
3. Część pisemna, patch-patch-extend dla aplikacji.

Cel

Według prorytetu:

1. Oddanie na czas,
2. Dobra ocena,
3. Nauczenie się czegoś nowego,
4. Bez tracenia nerwów.

Wspólna praca

1. Rażniej pracuje się w grupie,
2. Umówicie się na regularną wspólną pracę,
3. Możecie wspierać w nauce technologii lub metodologii,
4. Warto, aby kontakt trwał po studiach.

Tematy

- szeroko inżynieria oprogramowania,
- element oprogramowania,
- technologia w chmurze,
- może być projekt związany z obecnym zatrudnieniem.

Patrz: [przykładowe tematy](#).

Tematy

Wymagania formalne [1]

Praca dyplomowa – inżynierska powinna być opracowaniem określonego zagadnienia praktycznego albo dokonaniem technicznym prezentującym wiedzę i umiejętności studenta związane z danym kierunkiem studiów, poziomem studiów pierwszego stopnia i praktycznym profilem kształcenia dyscypliną wiodącą, do której kierunek został przypisany.

Tematy

Wymagania formalne [1]

Rezultatem pracy inżynierskiej powinien być projekt lub koncepcja szeroko rozumianego systemu IT w tym m. in. systemu informatycznego, sieci komputerowej, sieci telekomunikacyjnej oraz uruchomiona i przetestowana implementacja tego systemu lub jego fragmentu, albo komputerowa symulacja jego działania.

Aplikacja

- tracer bullet development / deliver patch-patch,
- iterations, po każdej push do gita.

Po dzisiejszym spotkaniu:

Przeczytać: [wymagania formalne](#).

Po dzisiejszym spotkaniu:

Przesać email:

- propozycją tematu + 4 zdania opisu, prefix w tytule emaila `[projekt nowatorski]`,
- Link to repozytorium githuba (sugerowane prywatne) z `README.md`,
- Proszę zaprosić `wojciech11` do repozytorium github*.

może też być `gitlab`

README.md

Trzy części:

- jaki problem rozwiązuje/dlaczego jest to ważne (*why*),
- co będzie dostarczone (*what*),
- jak będzie dostarczone (*how*).

Repozytorium

Docelowo, chcemy mieć zarówno kod, jak i część pisemną razem:

```
| - ...  
| - docs/  
|   | - related_work/  
|   | - data/  
|   | - ...  
|   \- (nazw)_(imie)_(numer albumu)_praca.docx  
|  
| - kod i implementacja  
|  
\- README.md
```

Repozytorium

Nazwa pliku z pracą, jeśli jest to `docx` (zauważ):

- `(nazwisko)(imie)(numer albumu)_praca.docx`

Składamy pracę w formacie pdf:

- `(nazwisko)(imie)(numer albumu)_praca.pdf`

Plan

Spotkanie 2:

- Zaakceptowany temat;
- Mieć gotowe:
 - Repozytorium git,
 - Plik `README.md` z tytułem oraz opisem.
- oraz pytania do tematu oraz wymagań formalnych

Plan

Spotkanie 3 - pierwsza iteracja tematu:

- 4 minut na osobę,
- prezentacja pomysłu i projektu.

Plan

Spotkanie 4 - druga iteracja tematu:

- demo MVP/PoC,
- 5 minut na osobę,
- prezentacja related work (książki, artykuły, dokumentacja tech, istniejące systemy),

Plan

Spotkanie 5 - Na zaliczenie:

- [x] github,
- [x] MVP,
- [x] spis treści,
- [x] 10 odnośników do literatury,
- [x] ogólnikowy plan.

Zaliczenie

Dodakowo, można dodać:

- mockup/wireframes/high-q
- schemat architektury, itp. itd.

Dziękuję za uwagę

Wojciech Barczyński

wojciech.barczynski@wroclaw.merito.pl

BACKUP SLIDES