

# Príloha D - Úvodné pokyny

## TODO:

- ☐ web sídlo
- ☐ rozchodiť project management system — všetci účty
- ☐ priradiť si roly

## utorok 9:00 — štandardný čas stretnutí

- *1st milestone* — nejaky docu, prezentácia (riadenia — akou formou sme manažovali v rámci tímu, roly, produkcia, continuous integration, plnosť testovania, dodržiavanie scrumu, retrospektiva, planning poker etc)
- *2nd milestone* — final document, prezentácia
- prihlásenie sa do *TP Cup* — prezentácia

## Web sídlo tímu

- bez ohľadu na to, či to od nás budú vyžadovať
- dostaneme virtualku
- idealne staticky page
  - dynamicky nema zmysel, pajcnuť free html sablonu
  - kontent ohľadom tímu
    - opis projektu
    - zoznam ľudí v tíme
    - stránka, kde budeme vesat zapisnice + priebežne vypisy
  - nahrávanie suborov staci vyriesit tym, ze raz za tyzden nieco pridáme do html
- html, js, nic ine

## Zápisnice

- 1 scrum master — dohľad na to, aby v tíme všetko fungovalo (nosiť kavu :D, kontaktovaný veducim), nie je veduci
- template — kto bol prítomný, čo sa na stretnutí robilo, prípadne retrospec

## Šprinty

- jeden sprint = 2 tyzdne
- 1. stretnutie — čo sa bude robiť
- pracovať
- na konci — ako sme splnili jednotlivé ulohy
- retrospec — čo sa pacilo, čo nie

## Project management

- cloudovy **DevOps** cez studentsku licenciu
  - nie [možno fungujúci] stuba, ale ms cloud server
- vytvoriť accounts + sample project
- alternatívy:
  - jira? nie optimalna pre 3+4h sprinty
  - trello, gitops?

## Role

- každý z nás za niečo zodpovedá
- manager of [...]
  - [documentation] — za všetku dokumentáciu
  - nerobi všetko v dokumentácii ten manažer, ale berie na seba zodpovednosť, keby sa niečo pokazilo

## Backlog

- *externy* = bude zadávaný niekym externým mimo tímu
  - môže prejsť do nevhodnej granularity
- *interny* = prispôbenie externého pod náš projekt
  - budeme vidieť iba jeho v DevOps

## User stories [US], tasks

- user story > tasks > subtasks
  - k US budeme mať pridelené niekoľko taskov (k jednotlivým taskom prípadne subtasks)
  - US — na úrovni storypointov, tasky — na úrovni hodín
- *user stories*
  - budeme to aj ohodnocovať cez storypoints
  - **planning poker**
    - nevzťahuje sa na hodiny, číslo nie je konkrétne
    - *kava* — vykaslime sa na to, dajme radšej pauzu, nikam to nevedie
    - ? — nepoužívať, radšej si niečo tipnime
    - 1 jednoducho, 21 strasne
      - ak dostaneme 21, tak to budeme musieť robiť
- v rámci sprintu sa rozhodnúť, ktoré US implementovať
  - do sprintu napchať niečo s najvyššou prioritou
- zodpovedný za US > rozbijeme na tasky > spravíme estimates > priradíme úlohy jednotlivým členom

## Burndown chart [BC]

- nie na úrovni storypointov — ubudalo by to iba za US (ktorych máme iba 2)
- cize radšej ubudat z taskov
- úroveň estimateov a remainingov
- **treba sledovať, že sa BC mení naozaj**
  - v poriadku, ak sa na prvých sprintoch nepodarí; ale na konci vyhodnotenia riadenia sa *pozrú na burndown chart*
- keď BC stupňuje — zle naplánovanie, lebo nám stúpili tasky (a keď to dropne, tak je vidieť, že sme to spravili v jeden deň na konci)
- ako s bugmi a defektami?
  - mohlo by sa to dať aj cez git
  - v rámci Azure DevOps [pravdepodobne task] má typ bug — reportnes, naplánujeme kedy to budeš fixovať
  - neprihodíš to ako nejaký report
  - či to included priamo v project managemente, tak sa ešte dohodneme

## Analytická časť

- od nás sa najviac očakáva návrh a implementácia
- grafová databáza
  - predbežne, ešte sa dohodne, či sa pôjde touto cestou
  - pretože je to vždy prepájané info
  - verziovanie
  - (nemusíme riešiť vizualizáciu)
  - **neo4j** = java graph db
    - backend môže byť v hocikom jazyku
- identifikovať typy projektov a fáz — ako vyzerá flow? (nie/len tento prípad sa bude riešiť)
  - napíše paper > pošle do systému 1 > reviewers napíšu review > review sa vráti chairovi > chair povie či prijať alebo nie ten paper > review sa vráti autorovi > spracovať, odovzdať odovzdať do ďalšieho systému > program konferencii je z prvého systému (stará verzia)
- môžu byť branchy dokumentov — spadlo by to do gitu, nemusíme asi
- management [spomínaný v zadani] — akoby 2. produkt, od nás sa to nečaká, ale aspoň niečo by sme mali touchnúť čo sa týka odrazok
- evidencia — upload sprievodných dokumentov ku konkrétnym dokumentom = prílohy
- web, kt. renderne pdf
  - nebudem pracovať so všetkými formátmi
  - formulár - web z db potiahne info, zobrazí v template
- eurocris — standard v rámci EU [zdroj zo zadania]
- typy projektov, fázy, proces výskumnej skupiny vzhľadom na fázu/typ projektu
- z toho všetkého sa spraví backlog a my si vyberieme, ktorú časť ideme implementovať

## Návrh systému

- modelovanie — EA
  - use cases sa robia v komunikácii s nimi (radšej sa vyhnúť)
- diagram komponentov — pohľad na architekturu
- class diagram, sekvenčáky — nie

## Implementácia

- rozchodiť environment
- docker — ak povieme, že bude len docker, tak bude len docker
- kubernetes
- na začiatku — test env, potom kvazi produkčný — stabilnejšia verzia, ktorú potom ukážeme
- pridať testy → *manazer testovania*
  - unit testy môžu byť všade — v kontexte toho celého nemusia povedať veľa though
  - integration (pokryť viacero komponentov); selenium, simulácia používateľov (vykon automaticky)
- možno to bude bežať na rovnakom serveri ako súvisiaci projekt

## Komunikácia

- dôležité z pohľadu riadenia (či bola integrácia)
  - msg do channelu
    - v rámci deploymentu
    - keď failnú testy
- continuous integration až keď spravíme env