***Fantasztikus Ötletek...***

...az IssueR-be.

**Diagramok**

<http://www.chartjs.org/> - NAGGYON JÓ CUCC!!! Javascript könyvtár, de igazából csak JSON objektumokat kell átadogatni neki. Szerver oldalon kiszámoljuk a szükséges adatokat, átadjuk a nézet modellen keresztül listák és felhasználói objektumk formájában, és a felületen átkonvertáljuk őket (@Html.Raw(Json.Encode(...))). Open source, NuGet-ben is fent van, és kellően részletes a doksija.

Piték és fánkok:

* Feladatok aránya státuszuk szerint (ToDo, InProgress, stb)
* Feladatok aránya típusuk szerint (Bug, Feature, Refactor)

Radar:

* Feladatok megoszlása a projekttagok között

Vonal:

* Feladatok számának időbeli változása típusuk/státuszuk szerint. A diagramon ezek szerint 3 vonal szerepelne, a bugok, feature-ök és refactorok számának megfelelően. Esetlegesen berakhatóak checkbox-ok amivel eltűntethető bármely adatsor. A megvalósításhoz egy ötlet: vegyünk fel egy táblát, amiben logoljuk az issue-k státusz változását. Minden módosításhoz bekerül egy rekord dátummal meg mindennel együtt. A lekérdezéskor pedig az adott projekthez tartozó logokat kiszűrjük, és nyomunk rá egy foldl-t (vagy végigiterálunk rajta) és előállítunk olyan adatsorokat, amiket a line chart meg tud jeleníteni.

Oszlop-és polár pite:

* Egyelőre nincs ötlet, de az oszlop is elég jól néz ki, érdemes lenne felhasználni. A polár pitét egyelőre nem értem, kicsit ökörkaki szaga van. Dehát biztos hasznos...

Időelemzés és kritikus út:

Forrás: Langer Tamás PVE előadása (4-5.). Minden feladathoz meg kell adni, hogy:

* mennyi az ő vegrahajtásához szükséges idő
* mely feladat(ok)nak kell befejeződnie, mielőtt ő elindulhatna (rákövetkezősésgi relációk)
* az első feladat legkorábbi kezdési idejét (vagy ha több is elindítható kezdetben, akkor mindegyik legkorábbi kezdése kell)

Ezek alapján kiszámolható minden feladatra:

* Legkorábbi befejezés
* Legkésőbbi befejezés
* Legkésőbbi kezdés
* Mely feladatokkal végezhetőek párhuzamosan

Az elemzésből felrajzolható:

* A kritikus út: melyek azok a feladatok, amelyek nem csúszhatnak.
* Gantt-chart: sávdiagramban jelzi a kritikus utat és a fennmaradó szabad időt, amivel egyes feladatok eltolhatóak.

Ezekhez még nem kerestem vizualizáló könyvtárakat, de nem lepedőnék meg rajta, ha lenne egy csomó ingyenes, open source cucc hozzá.