

Rendszerfejlesztés II.

Gyakorlati Projektmunkák

Szegedi Tudományegyetem

2020. január 30.

1. Általános követelmények

- A csapatok egy open-source fejlesztésbe kapcsolódnak be a félév során.
- A projekteket optimálisan 5, de min. 4, max. 6 fős csapatok viszik.
- A csapaton belüli bármilyen probléma (pl. inaktív tagok) esetén, a első előfordulás alkalmával tájékoztatni kell a gyakorlatvezetőt e-mailben, melyen minden csapattagot címezni kell (CC). Az értékelés után beérkező csapatot érintő panaszokat nem fogadjuk el.
- A csapatok kialakítása véletlenszerűen történik a gyakorlatokon belül, a kialakított csapatokról a gyakorlatvezető értesíti a csapatokat.
- Projektekből összesen 65 pontot lehet összegyűjteni a félév során. Részenként és összesen min. 50% teljesítendő. A felosztást lásd az ezt követő részekben.
- A gyakorlatvezető a kiemelkedő egyéni teljesítményt (kezdeményező készség, megfelelő kommunikáció, csapatvezetés, stb.) plusz ponttal jutalmazhatja. Egy hallgató a félév során maximum 75 pontot szerezhethet a projektmunkára, az egyéni pluszpontokkal és a „közös” feladatokra kapott pontokkal együtt.
- Bármelyik kötelezőnek megjelölt leadandó hiánya esetén a hallgató (vagy csapat, a leadandó jellegétől függően) elégtelen érdemjegyet kap.
- Csak azt a leadott anyagot tudjuk az egyéni értékelés során figyelembe venni, ami a saját GitLab felhasználó segítségével került feltöltésére. Ez minden esetben a H-s azonosító, vagy az egyetemi LDAP belépés. Csak olyan commit-okat fogadjuk el ami a hallgató saját, stud-os címéről készült. (Az Irinyi Kabinetben való használat esetén minden újraindítás után be kell állítani a Git user nevet és e-mail címet, ezzel elkerülve hogy az alapértelmezett felhasználó nevében szülessenek a commit-ok.¹)
- A leadandók esetén a megadott névkonvenciókat szigorúan tartani kell. A névkonvenciót nem tartó leadás nem minősül elfogadott teljesítésnek (akkor sem, ha határidőig beérkezett).
- A „közös” értékelési módú leadandók esetén lehetőség van pótlásra. A leadási határidőtől számított késés esetén a felelősöktől naponta 2p kerül levonásra. Ha a gyakorlatvezető nem tudja eldönteni, hogy ki a felelős akkor a 2p minden csapattagtól levonásra kerül.
- A „személyenként egyénileg” értékelési módú leadandóknál a gyakorlatvezető lehetőséget adhat egy részfeladat esetében pótlásra. Amennyiben a hallgató az adott részfeladatot az értékelést követő 4 napon belül (23:59) pótolja úgy (ha a többi feladatot teljesíti) megkaphatja a projektmunkára a pontok 50%-át. Ezt a pótlási lehetőséget egy hallgató csak egyszer használhatja. Ilyen esetben a hallgató nem szerezhethet 50%-nál több pontot a projektmunkára.
- A projektmunka elvégzése a félév során folyamatos és az tanórákon túli munkát igényel. Az ipari projektekhez hasonlóan a nem megfelelő minőségű munka elvégzése a gyakorlat során is következményekkel jár. Ezen okok miatt a projektmunkák javítására nincs lehetőség.
- Minden leadandóban csak azok a részek fogadhatóak el új teljesítésnek, amik korábbi, a projekttel kapcsolatos teljesítésben még nem lettek leadva.
- Ahol a „közös” értékelési mód van megadva ott a gyakorlatvezető a hallgatók egyéni teljesítményét és javaslatait figyelembe véve, de a saját szakmai tapasztalatai alapján dönti el hogy milyen arányban osztja el a feladatra kapott pontot.
 - A csapat javaslatot tehet, melyben megadja, hogy a közös részekben ki és milyen mértékben (%) dolgozott. A javaslattal minden csapattagnak bele egyet kell értenie.
 - A javaslat során a hallgatóknak személyenként részletezniük kell az egyes elvégzett részeket, melyeket célszerű Git commit-okkal alátámasztani.
 - A hozzájárulás részletezését gyakorlatvezetőnek az értékelést követő 7 napon belül (23:59) e-mailben kell elküldeni melyen minden csapattagot címezni kell (CC). A levél tárgya: „*[RF2]* *<csapatnév>*: *<leadandó címe>* hozzájárulások”. A csapattagoknak az e-mail beérkezésétől számított 24 óráig van

¹<https://help.github.com/en/github/using-git/setting-your-username-in-git>

- idejük jelezni (e-mail-ben) a ha nem értenek egyet a csapat által küldött hozzájárulás felosztással, mely esetben a pontokat a gyakorlatvezető osztja el.
- Amennyiben a csapat képtelen megegyezni, vagy erről nem küld megfelelő tájékoztatást úgy a pontokat a gyakorlatvezető saját belátása szerint osztja el a csapat tagjai között. A gyakorlatvezető felülbíráhatja a hallgatók által küldött hozzájárulás elosztást.
- A feladatok végrehajtása során mindig jelöljük meg azt a master branch-en lévő commit-ot, mely tartalmazza a feladata minden eredményét. A jelöléshez használjuk a megfelelő tag-eket.

```
1 git tag -a <rövidítés> -m "<megjegyzés>"
2 git push origin <rövidítés>
```

Például:

```
1 git tag -a sample_task -m "final working version"
2 git push origin sample_task
```

- A „/” mindig a GitLab projekthez társított Git repository gyökerét jelzi.
- A leadandók bemutatása során a gyakorlatvezető által szigorúan tartani kell az időkeretet. Ne tervezzünk túl hosszú prezentációt!
- A gyakorlat célja hogy a hallgató elsajátítsa az önálló munkavégzéshez szükséges szaktudást és képességeket. A megadott jegy ennek sikerességét tükrözi. Azt hogy a leadott munka eléri-e a megfelelő minőséget és mennyiséget a gyakorlatvezető dönti el figyelembe véve az ebben a dokumentumban meghatározottakat.
- A kiemelkedő minőségű projektmunkákat a gyakorlatvezető előadás megajánlott jegyre javasolhatja az előadónál. A megajánlott jegy feltételei és megítélése az előadó által történik az általa meghatározott és hirdetett szabályok szerint.

2. Fontosabb formai követelmények

- Minden szövegnek, ábrának és képnek olvashatónak kell lennie A4-es lapra nyomtatás során ill. prezentáció esetén a kivetítőn nagyítás nélkül (javasolt betűméret: 11-14pt).
- A megadott oldalszámok, minden esetben a címlap és az esetleges tartalomjegyzék nélkül értendők.
- A leadott anyagok nyelve magyar vagy angol, de dokumentumon belül ne keverjük a nyelveket (kivétel az egyes forráskód idézetek).
- A kettőnél kevesebb elemből álló ábra (diagram, gráf) nem számít ábrának.
- Az olyan ábra ahol nincs az elemek között kapcsolat (él) nem számít ábrának.
- Az olyan ábrát vagy képernyőképet amihez nem tartozik legalább egy bekezdésnyi magyarázó vagy részletező szöveg nem vesszük figyelembe a teljesítés során.
- Az ábráknak és a képernyőképeknek tartalmaznia kell a bemutatott elemeket. Logikátlan vagy követhetetlen módon feldarabolt ábrákat nem fogadunk el.
- A prezentációk során javasoljuk a „The beamer class User Guide for version 3.57” 5. fejezetében („Guidelines for Creating Presentations”)² leírt irányelvek betartását. (a prezentációkat nem kötelező L^AT_EX-ben készíteni)
- Az itt felsoroltak csak a kiemelten fontos formai dolgokat érintik, egyéb formai kérdésekben a gyakorlatvezető dönt.

3. Projekt megismerése és változtatási igények összeállítása

rövidítés	„ <i>comprehension</i> ” feladat
max. pontszám	50 pont/csapat
min. pontszám	5 pont/fő

Cél az átvett projekt megismerése, a változtatási igény benyújtása. Leadandó a futó program és a projektet röviden bemutató, legalább 8 oldalas dokumentum, ami ismerteti

- az átvett program szerkezetét (UML package, osztály vagy egyéb szerkezeti diagram, pl. E/K) és az átvett program fő funkcióit (UseCase, szekvencia diagram, funkcióleírások screenshotokkal)
 - összesen legalább annyiféle diagramot kell átadni ahány fős a csapat³ (pl.: 4 fő esetén: 2 osztálydiagram, 1 csomagdiagram, 1 szekvenciadiagram és 2 folyamatábra = 4-féle diagram)
- az átvétel során tapasztalt hiányosságokat, nehézségeket, esetleg pozitívumokat

²<http://tug.ctan.org/macros/latex/contrib/beamer/doc/beameruserguide.pdf>

³javasolt diagram típusok: https://en.wikipedia.org/wiki/Unified_Modeling_Language#Diagrams, https://en.wikipedia.org/wiki/Modeling_language#Graphical_types

- a tervezett változtatásokat, amik lehetnek új funkciók és minőségjavítás is⁴

Fontos, hogy a változtatási igényt annyi alfeladatra kell szétbontani, ahány csapattag van, hogy a későbbi feladatok személyenként számonkérhetőek legyenek. A csapat által fejlesztett projekthez kapcsolódó issue-k közül minden tag maga választ. A választást a gyakorlatvezető hagyja jóvá. Egy issue-t fejleszthet több hallgató és egy hallgató több issue-t is vállalhat, amennyiben ezt a gyakorlatvezető jóvá hagyja.

A fejlesztés során felmerülő előre nem látható komplikációkról azonnal tájékoztatni kell a gyakorlatvezetőt (ha másképp nem rendelkezik akkor e-mail-ben). A problémák jellege alapján a gyakorlatvezető enyhíthet bizonyos követelményeken, hogy azok egyensúlyban maradjanak a tárgy teljesítéséhez szükséges erőforrásokkal.

A leadáskor minden csapatnak rendelkezni kell egy GitLab group-pal⁵, melybe minden tagot felvettek. E csoport tulajdonában kell lennie egy GitLab projektnek⁶, melyben a csapat a feladatok megvalósítását követi és teszi közzé.

leadandók és teljesítendőik	
<i>A választott projekt leírása</i>	
névkonvenció	/doc/comprehension/⟨csapatnév⟩-project.pdf
formátum	folyószöveges dokumentum
min. méret	1 diagram és magyarázó szöveg/fő
értékelési mód	közös
kötelező?	igen
max. pontszám	40 pont/csapat
<i>A választott projekt bemutatása</i>	
névkonvenció	/doc/comprehension/⟨csapatnév⟩-presentation.pdf
formátum	prezentáció
max. méret	10 perc/csapat
értékelési mód	közös
kötelező?	nem
max. pontszám	10 pont/csapat
<i>Változtatás igény ismertetése és felosztása</i>	
névkonvenció	/doc/comprehension/⟨csapatnév⟩-changereq.pdf
formátum	folyószöveges dokumentum
min. méret	2 oldal/csapat
értékelési mód	közös
kötelező?	igen
max. pontszám	10 pont/csapat
<i>GitLab csoport</i>	
névkonvenció	⟨gyakorlat Neptun kódja⟩-⟨csapatnév ékezetek nélkül⟩
formátum	GitLab csoport
min. méret	összes csapattag
értékelési mód	személyenként egyénileg
kötelező?	igen
max. pontszám	0 pont
<i>GitLab projekt</i>	
névkonvenció	⟨projekt neve ékezetek nélkül⟩
formátum	GitLab projekt
min. méret	minden kötelező leadandó folyamatosan vezetve
értékelési mód	személyenként egyénileg
kötelező?	igen
max. pontszám	0 pont

⁴Minőségjavítás esetén a bemutatás során ki kell térni a leggyakoribb vagy a legsúlyosabb hibákra amik javításra kerültek, ezeket célszerű kódrészletek segítségével bemutatni.

⁵<https://docs.gitlab.com/ee/user/group/index.html>

⁶<https://docs.gitlab.com/ee/user/project/>

4. Fejlesztés és tesztelés

4.1. Változtatási igény implementálása

rövidítés	„development” feladat
max. pontszám	25 pont/fő
min. pontszám	12 pont/fő

A bemutatón az implementált változtatásokat, új feature-öket kell bemutatni. A csapat minden tagjának ismernie kell a főbb változtatásokat.

A fejlesztés GitLab környezetben történik. A futtatható kódot a master branch-en kell elhelyezni és az development tag-gel kell ellátni a Git-ben. A leadandóhoz a határidőre fordítható, bemutatható kódnak kell lennie, amit a bemutató időpontján kell bemutatni. A bemutatóhoz prezentációs anyag készíthető, de nem kötelező.

leadandók és teljesítendőik	
<i>Az változtatásokat tartalmazó projekt forráskódja</i>	
névkonvenció	/src/**/*.java
formátum	forráskód fájlok
min. méret	nincs megadva
értékelési mód	személyenként egyénileg, élő bemutató alapján
kötelező?	igen
max. pontszám	25 pont/fő
<i>A változtatások megvalósításának bemutatása</i>	
névkonvenció	/doc/development/<csapatnév>-presentation.pdf
formátum	prezentáció
max. méret	15 perc/csapat
értékelési mód	közös
kötelező?	nem
max. pontszám	10 pont/csapat

4.2. Unit tesztek

rövidítés	„testing” feladat
max. pontszám	10 pont/fő
min. pontszám	5 pont/fő

Feladat a projekthez unit tesztek implementálása és futtatása. A feladat során a unit tesztek forráskódját (*.java fájlok) a master branch-en kell elhelyezni és az testing tag-gel kell ellátni Git-ben. Csapattagonként legalább 10 különböző metódus unit tesztjeit be kell tudni mutatni. A tesztelésre kiválasztott metódusok során vegyük figyelembe a következőket.

- Ne teszteljünk triviális kódot, például egysoros „getter” és „setter”.
- A unit tesztelés célja hogy ellenőrizze hogy a metódus előre meghatározott bemenetere megfelelő (literálként megadott) eredményt ad.
- Elsősorban a saját változtatás megvalósítását teszteljük, vagyis azokat a metódusokat amiket mi írtunk, vagy módosítottunk.
- Másodsorban a saját változás megvalósításához közvetlenül kapcsolódó részeket teszteljük, vagyis amik ugyan abban az osztályban vagy névtérben vannak (package).
- Amennyiben nincs elegendő metódus a projektben úgy ezt azonnal jelezzük a gyakorlat vezető felé. Az értékelés utáni panaszokat nem tudjuk elfogadni.
- Ne teszteljünk többször ugyanazt. Amennyiben nincs elegendő metódus úgy pontosan válasszuk el hogy ki melyik esetet teszteli, például helyes és helytelen bemenetre való viselkedés.
- Az indoklás és magyarázat nélküli lefedettség adatokat nem fogadjuk el.

leadandók és teljesítendőik	
<i>Tesztelés jegyzőkönyv és lefedettségi jelentés</i>	
névkonvenció	/doc/testing/<csapatnév>-testreport.pdf
formátum	folyószöveges dokumentum
min. méret	2 oldal/fő
értékelési mód	személyenként egyénileg
kötelező?	igen
max. pontszám	5 pont/fő

<i>Egység/unit tesztek forráskódja</i>	
névkonvenció	/src/★/★.java
formátum	forráskód fájlok
min. méret	10 különböző metódus tesztjei
értékelési mód	személyenként egyénileg
kötelező?	igen
max. pontszám	5 pont/fő

5. A fejlesztési és egyéb folyamatok követése

rövidítés	„processes” feladat
max. pontszám	50 pont/csapat
min. pontszám	5 pont/fő

A félév során a következő fejlesztési elemek dokumentált használatát pontozzuk: heti megbeszélés, issuek karbantartása, Git és GitLab használat, git flow követése, melyeket legkésőbb az „comprehension” feladat bemutatásának hetétől kezdődően az „processes” feladat leadásáig legalább hetente kell frissíteni. Az agilis módszertanokhoz hasonlóan kövessük nyomon a feladatok végrehajtását. Minden csapattagnak legyenek saját issuejai. Az issue board státuszai tartalmazzák a következőket: Új (To Do, Backlog), Folyamatban (Doing), Tesztelésre vár (To Test), Tesztelés alatt (Testing), Lezárt (Closed) (vagy ezek angol megfelelőit).

Gitben nagy bináris állományok (*.class, *.jar, *.zip, *.rar, ...) tárolása tilos (max. 3 pont levonás jár érte), a .gitignore fájl segíthet ennek követésében.⁷ A példa projekt a következő címen érhető el: <http://gitlab-okt.sed.hu/root/Sample>. A példa projekt módosítása szigorúan tilos. Aki ennek ellenére módosítja, attól módosításonként 2 pont kerül levonásra.

A projekt lezárásaként egy összegző dokumentumot kell készíteni és a csapattal a Scrum módszertanhoz hasonlóan írásbeli retrospekiót kell tartania. Az összegzést a csapat GitLab projektjéhez tartozó Git repozitoriban kell elhelyezni a mater branch-en. A dokumentum (README.md) tartalmaznia kell az minden részfeladatok teljesítésére vonatkozó hivatkozásokat⁸ és Markdown⁹ formátumban kell elkészülnie.

A retrospekió egy Coospace kérdőív kitöltése jelenti, melyet a gyakorlatvezető fog kihirdetni. A kitöltés kötelező és nem anoním. A retrospekió során a hallgató röviden értékeli saját és csapattársai teljesítményét valamint a csapatban betöltött szerepét és hozzájárulását a projekt sikeréhez.

Az „processes” feladat átadandóinak tartalmaznia kell:

- Legalább 5 db meeting memo, melyet az adott projekthez GitLab-ban, a Wiki-be kell rögzíteni.¹⁰
- Minden feladat felvitele, pontos határidőkkel az adott projektbe GitLab-ban¹¹. Valamint az alábbi követelmények szerint Issue-k létrehozása az egyes feladatok számára.
 - Feladatonként legalább 3 issue létrehozása
 - Felhasználónként legalább 4 issue létrehozása
- Minden issue folyamatos vezetése a task board (Scrum vagy Kanban tábla) segítségével¹². Ellenőrzésre az issue-k státusz log-ja kerül felhasználása, ahol az egyes státusz változások között indokolható időnek kell eltelnie. Vitás kérdésekben a log összehasonlításra kerül a Git repository commitjaival.
- Projekt összegző leírás
- Retrospekió a csapatról

⁷A legtöbb projekt már tartalmaz ilyen fájlt, amelyik mégsem ahhoz könnyen létrehozhatunk egyet a <https://www.gitignore.io/> segítségével

⁸<http://gitlab-okt.sed.hu/root/Sample/blob/master/README.md>

⁹<https://docs.gitlab.com/ee/user/markdown.html>

¹⁰<http://gitlab-okt.sed.hu/root/Sample/wikis/sample-memo>

¹¹<http://gitlab-okt.sed.hu/root/Sample/milestones>

¹²[http://gitlab-okt.sed.hu/root/Sample/boards?="](http://gitlab-okt.sed.hu/root/Sample/boards?=)

leadandók és teljesítendőik	
<i>Megbeszélés jegyzet</i>	
névkonvenció	http://gitlab-okt.sed.hu/root/Sample/wikis/sample-memo
formátum	GitLab Wiki bejegyzés
min. méret	5 darab/csapat
értékelési mód	közös
kötelező?	igen
max. pontszám	20 pont/csapat
<i>Mérföldkövek</i>	
névkonvenció	http://gitlab-okt.sed.hu/root/Sample/milestones
formátum	GitLab mérföldkő
min. méret	2 darab/csapat
értékelési mód	közös
kötelező?	igen
max. pontszám	10 pont/csapat
<i>Issue-k</i>	
névkonvenció	http://gitlab-okt.sed.hu/root/Sample/boards?="
formátum	GitLab issue
min. méret	3 darab/mérföldkő és 4 darab/fő
értékelési mód	közös
kötelező?	igen
max. pontszám	20 pont/csapat
<i>Projekt összegző leírás</i>	
névkonvenció	http://gitlab-okt.sed.hu/root/Sample/blob/master/README.md
formátum	Markdown
min. méret	minden leadandóra és csapattagra mutató hivatkozás
értékelési mód	közös
kötelező?	igen
max. pontszám	5 pont/csapat
<i>Retrospekció a csapatról</i>	
névkonvenció	nincs megadva
formátum	Coospace kérdőív
min. méret	nincs megadva
értékelési mód	személyenként egyénileg
kötelező?	igen
max. pontszám	10 pont/fő

6. Minősbiztosítás és elemzés

Az alábbi feladatok közül a csapat a projekt jellegét figyelembe véve és a gyakorlatvezető jóváhagyásával választ egy feladatot. A kiválasztott feladat nevét még a „*development*” feladat leadása előtt el kell küldeni e-mail-ben a gyakorlatvezetőnek. Az e-mail tárgya legyen: „[RF2] minősbiztosítás feladatválasztás”. A választás csak akkor számít elfogadottnak, ha azt a gyakorlatvezető válaszlevélben megerősíti. A gyakorlatvezető jóváhagyása nélküli feladat végrehajtását és eredményét nem fogadjuk el.

6.1. Minőségriport statikus kódelemzés alapján

rövidítés	„ <i>static</i> ” feladat
max. pontszám	50 pont/csapat
min. pontszám	5 pont/fő

A feladat célja hogy képet kapjunk a projekt minőségéről és ennek változásáról a módosítások elvégzése előtt, közben és után. Ezért a fejlesztések végén a projektről minőségriport készül a projekt kezdeti és végső állapota

alapján. A riportot a gyakorlatokon bemutatott eszközök segítségével ajánlott elkészíteni, de használható más eszköz is saját felelősségre.

Az elemzésben és a bemutató során részletezni kell a következőket:

- A projekt forráskódjának metrikák és egyéb mért jellemzőkkel alátámasztott kezdeti és fejlesztés utáni minősége, annak változása.
- A megvalósítás (forráskód) metrikákból levonható és konkrét példákkal alátámasztott jellegzetességei. Elemezhetőek többek között az egyes metrikák
 - kiemelkedő értékeinek magyarázata és jogossága
 - értékeinek egymással való kapcsolata és ezek pozitív, negatív vagy semleges hatása a fejlesztett projektekre
 - időbeli változásának értékelése és magyarázata
- A forráskód szabálysértések vizsgálata.
 - jogos és jogtalan jelzések értékelése
 - leggyakoribb és legritkább jelzések magyarázata
- figyelmet érdemlő (pozitív vagy negatív) kódrészek bemutatása és magyarázata

leadandók és teljesítendő	
<i>Minőségriport</i>	
névkonvenció	/doc/quality/static/<csapatnév>-qualityreport.pdf
formátum	folyószöveges dokumentum
min. méret	5 oldal/csapat
értékelési mód	közös
kötelező?	igen (feladat választása esetén)
max. pontszám	50 pont/csapat
<i>A projekt minőségének és változásának bemutatása</i>	
névkonvenció	/doc/quality/static/<csapatnév>-presentation.pdf
formátum	prezentáció
max. méret	10 perc/csapat
értékelési mód	közös
kötelező?	nem
max. pontszám	5 pont/csapat

6.2. Szoftver vizualizáció

rövidítés	„visualization” feladat
max. pontszám	50 pont/csapat
min. pontszám	5 pont/fő

A fejlesztés végén bemutatandó a minőség mérés során használt metrikák vizualizációja, grafikus megjelenítése, ill. az ehhez kapcsolódó szöveges elemzés. A vizualizáláshoz a gyakorlaton bemutatott CodeMetropolis¹³ vagy más vizualizációs eszközt¹⁴ (pl.: Gource¹⁵) lehet használni a gyakorlatvezetővel való egyeztetés után.

Az elemzésben és a bemutató során részletezni kell a következőket:

- A szoftver tulajdonságok és jellegzetességek szemléltetése a megadott vizualizációs eszközök segítségével. A határidőre minden csapatnak rendelkeznie kell egy saját vizualizációval, melyet be kell mutatni.
 - a létrehozott grafikus megjelenítésen a megfelelő rész megjelölése
 - a grafikus és az eredeti program elem vagy tulajdonság nevét, azonosítóját, stb.
 - a megjelenített tulajdonságok nevét, ha van értékét
- figyelmet érdemlő (pozitív vagy negatív) kódrészek bemutatása és magyarázata
- A felhasznált vizualizációs eljárás hasznosságának elemzése, konkrét példákkal alátámasztott érvekkel. Részletezendő a következőkkel kapcsolatban szerzett negatív és pozitív tapasztalatok, ötletek, javaslatok egyaránt.
 - a használt metafora
 - ennek megvalósításához használt technológiák és módszerek
 - a konkrét vizualizációs programcsomag

¹³<http://codemetropolis.github.io/CodeMetropolis/>

¹⁴<https://lmgtyf.com/?q=software+visualization>

¹⁵<https://gource.io/>

leadandók és teljesítendőik	
<i>Forráskód vizualizáció</i>	
névkonvenció	nem kell feltölteni csak bemutatandó
formátum	mozgó vagy állókép
min. méret	nincs megadva
értékelési mód	közös, élő bemutató alapján
kötelező?	igen (feladat választása esetén)
max. pontszám	20 pont/csapat
<i>Szoftvervizualizáció ismertetése</i>	
névkonvenció	/doc/quality/visualization/<csapatnév>-visualization.pdf
formátum	flyszöveges dokumentum
min. méret	3 oldal ábrák és képek nélkül
értékelési mód	közös
kötelező?	igen (feladat választása esetén)
max. pontszám	30 pont/csapat
<i>A szoftver vizualizáció bemutatása</i>	
névkonvenció	/doc/quality/visualization/<csapatnév>-presentation.pdf
formátum	prezentáció
max. méret	10 perc/csapat
értékelési mód	közös
kötelező?	nem
max. pontszám	5 pont/csapat

6.3. Tesztlefedettség vizsgálata

rövidítés	„coverage” feladat
max. pontszám	50 pont/csapat
min. pontszám	5 pont/fő

Feladat a projekthez a csapat által végzett módosításokat ellenőrző (lefedő) tesztek implementálása és végrehajtása valamint tesztjegyzőkönyv készítése. A feladat során a unit tesztek forráskódját (*.java fájlok) és a lefedettségi eredményeket bemutató riportot a master branch-en kell elhelyezni és az testing tag-gel kell ellátni Git-ben.

A leadott anyagnak röviden magyaráznia kell a tesztelés és a hozzá kapcsolódó mérés során kapott értékeket, különös tekintettel a megvalósított módosításokhoz kapcsolódó forráskódét. A tesztelés során vegyük figyelembe és a leadott anyagban foglaljuk bele a következőket.

- tesztelés során felhasznált eszközök és technikák ismertetése
- létrehozott teszt csomag bemutatása
- létrehozott tesztek pontos megadása (forráskód vagy szöveges)
 - előkészítés/előfeltételek (arrange)
 - végrehajtási lépések (act)
 - viselkedés és eredmények kiértékelése (assert)
- végrehajtott tesztek sikerességének ismertetése
- tesztek végrehajtása során a lefedettség szisztematikus (akár automatikus) mérése
 - felhasznált technikák és eszközök
 - lefedettségi érték indoklása
 - erősen és kevésbé lefedett kódelemek bemutatása (csapat által fejlesztett részekből)
 - könnyen és nehezebben tesztelhető funkciók és változtatások

leadandók és teljesítendőik

Tesztjegyzőkönyv

névkonvenció	/doc/quality/coverage/⟨csapatnév⟩-testreport.pdf
formátum	folyószöveges dokumentum
min. méret	a csapat által végzett változtatások tesztelése
értékelési mód	közös
kötelező?	igen (feladat választása esetén)
max. pontszám	25 pont/csapat

Tesztlefedettség riport

névkonvenció	/doc/quality/coverage/⟨csapatnév⟩-coveragereport.pdf
formátum	folyószöveges dokumentum
min. méret	a csapat által végzett változtatások lefedettsége
értékelési mód	közös
kötelező?	igen (feladat választása esetén)
max. pontszám	25 pont/csapat

A tesztlefedettség vizsgálatának bemutatása

névkonvenció	/doc/quality/coverage/⟨csapatnév⟩-presentation.pdf
formátum	prezentáció
max. méret	10 perc/csapat
értékelési mód	közös
kötelező?	nem
max. pontszám	5 pont/csapat