

# Esettanulmány: Agilis szoftverfejlesztés egyetemi kurzuson

Ilyés Enikő

ilyese@inf.elte.hu  
ELTE IK

**Absztrakt.** Az agilis módszertanok iparban lévő népszerűségének hatására az oktatás területén is megjelent az igény az agilis módszertanok megismertetésére, alkalmazására. A korábbi években az ELTE Informatikai Kar MSc-s programtervező informatikus hallgatói többnyire egy informatikai projektirányítás témájú előadás-sorozat kapcsán kaphattak részletes betekintést a módszertan (elméleti) működésébe. 2016-ban néhány oktató úgy döntött, hogy a „Szoftverfejlesztés a gyakorlatban” kurzusán Scrumot használ egy-egy csoport fejlesztési munkájának levezetéséhez, így a hallgatók lehetőséget kaptak gyakorlati tapasztalatot is szerezni az agilis módszertanokat illetően. Ebben a cikkben ennek a kísérletnek a tapasztalatait foglaljuk össze, rávilágítva a Scrum módszer kipróbálása kapcsán fölmerülő kérdésekre és az egyes kérdésekre adható egy lehetséges válaszra.

**Kulcsszavak:** agilis, Scrum, oktatás

## 1. Bevezetés

Napjainkban a szoftverfejlesztés divatos modellje az agilis. Hazánkban is számos ipari környezetben átvette a Vizesés modell korábbi uralmát, alkalmazása sikereknek örvend.

Az agilis módszertanok csoportjába soroljuk a Scrumot, extrém programozást, Kanbant, Leant és ezek különböző fúzióit. Mindezek gyakorlatai az agilis elvek érvényesülésére biztosítanak terepet, vagyis a személyes kommunikációra, az ügyféllel való együttműködésre, a változásokra való reagálásra és a működő kód (szemben az átfogó dokumentációval) felmutatására (1).

Az agilis módszertanok a Vizesés modell (2) gyengeségeinek feloldása céljából jöttek létre. Ezek a gyengeségek többek között a modell oktatásban való alkalmazása során is megfigyelhetőek (3). A hallgatók számottevően sok energiát fektetnek a projekt részletes követelmény elemzésébe, szinte több időt töltenek dokumentálással, mint fejlesztéssel. Gyakori, hogy a csoport a gyengébb tagokra bízza a dokumentálási feladatokat, míg az erősebbek a tervezésre és kódolásra fókuszálnak. Így viszont néhány csoporttag peremre szorul, nem fejlődik, illetve a dokumentumok nem tudják a valódi fejlesztési folyamatot tükrözni. Gond lehet, hogy a megjelenítés tervezése, az elemzés és a magas szintű tervezés olyan szignifikáns részt kivesz a félévi munkából, hogy az implementációra, kódolásra csak a késői, nagyon megterhelő időszakban kerül sor. Ilyen körülmények között nem jut már elég fókusz a tesztre sem, sőt, könnyen lehet, hogy egy (legalább viszonylagosan) működő szoftver sem alakul ki a fejlesztési időszak végére. Összefoglalva: egyetemi kurzus keretei között a Vizesés modell alkalmazása nagy valószínűséggel frusztráció, negatív érzések, stressz, kellemetlen eredmények kialakulásához vezet.

A fenn felsoroltak arra buzdítanak bennünket, hogy dolgozzuk ki az agilis módszertanok oktatásban történő alkalmazásának részleteit. Ez nem triviális feladat, tekintettel azokra a keretekre, melyeket az egyetemi oktatási rendszer korlátai idéznek elő, s melyek eltérőek attól az ipari környezettől, melyben az agilis módszertanok virágoznak.

A következő, 2. fejezetben bemutatjuk azokat az körülményeket, melyek között az agilis módszertanok oktatásba történő alkalmazására vonatkozó kísérletünk megvalósult. A 3. fejezet tárgyalja, hogy milyen kérdések merültek fel a kísérlet megtervezése során, a 4. fejezet pedig a kérdésekre adott válaszokat és azok megvalósulását részletezi. Az 5. fejezetben a megvalósulást követő visszajelzésekről esik szó, míg a 6. fejezet a kísérlet kimenetele kapcsán megfogalmazott következtetéseket tartalmazza. Köszönetnyilvánítással zárjuk.

## 2. Körülmények

A „Szoftverfejlesztés a gyakorlatban” az ELTE Informatikai Kar Szoftvertechnológia szakirányú mesterképzésén kötelező tárgy (4). A szakirány elméleti tárgyait ez a gyakorlati foglalkozás köti össze, amelyen a hallgatók az elméleti ismereteket alkalmazzák a gyakorlatban, csapatmunkában készítenek el egy szoftverfejlesztési projektet.

A tárgy két félév alatt teljesíthető (2x13 hét). Az oktatók és hallgatók hetente egy alkalommal találkoznak kötelező módon, összesen 90 percnyi időtartamot töltenek együtt. Ezt kiegészítendő hetente egy 90 perces időszáv, mely konzultációs alkalom, és mely arra van fenntartva, hogy a tárgy kapcsán esetlegesen felmerülő kérdéseket a hallgatók feltehessék, átbeszélhessék oktatóikkal. Heti nyolc óra egyéni munka várható el a hallgatóktól.

Egy gyakorlati csoportba körülbelül 10 hallgató és két oktató tartozik. A hallgatók programozással kapcsolatos előismerete, tapasztalata nem egységes, de egy erős közös mag feltételezhető. Az oktatóknak rendszeresen kihívást jelent a megfelelő méretű, típusú feladat kiválasztása. A csoportmunka megvalósításához legalkalmasabb szoftverfejlesztési módszertan kiválasztása ugyancsak körültekintést, figyelmet igényel.

2016-ban a „Szoftverfejlesztés a gyakorlatban” egyetemi kurzus jelentette azt a keretet, ahol kipróbálásra került az agilis módszertanok egyetemi szoftverfejlesztés során való alkalmazása. Az oktatók a Scrum módszer mellett kötelezték el magukat a kurzus kapcsán.

## 3. Kérdések

A Scrum módszer egy keretrendszer, néhány „szabállyal”, gyakorlattal, melyek az agilis elvek megvalósulására kívánnak jó terepet létrehozni (5). A napi Scrum, a visszatekintés a fejlesztők közötti személyes kommunikációt erősítheti, a futamonkénti tervezés az ügyféllel való együttműködést és a változásokra való reagálást célozza, a futamonkénti bemutató a működő kódot hangsúlyozza a dokumentációval szemben. Emellett a Scrum szerepkörök is az agilitás erősítésére irányulnak.

Ezek a gyakorlatok azonban az ipari környezethez vannak igazítva, ahol rendszerint a fejlesztő csapat tagjai napi 8 órát töltenek egy bizonyos helyszínen, megfelelő módon kiépített infrastruktúrában, hosszabb ideig együtt dolgozva, egyéb anyagi és környezeti tényezők által motiválva, stb. Ezek hiányában és az egyetemi környezet egyéb sajátosságai révén, a módszer egyetemi gyakorlatba való átültetése a következőkben felsorolt kérdéseket hozza az oktatók elé.

### 3.1. Szerepkörök

A Scrum projekt megrendelője kapcsán fontos, hogy agilis hozzáállást mutasson – ennek hiányában nehéz agilis projektet megvalósítani. A termékgazda (Product owner) az üzleti célok teljesüléséért felelős, a Scrum mester (Scrum master) a módszertan megvalósulásáért, a Scrum csapat (Scrum team) a tulajdonképpeni, önálló fejlesztésért.

Kérdések a szerepkörök kapcsán:

- Ki legyen a megrendelő (pl. ipari partner vagy oktató?), tekintettel arra, hogy agilis módon tudjon hozzáállni a projekthez, ugyanakkor a hallgatók minél motiváltabbak legyenek?
- b) Ki legyen a termékgyártó (pl. oktató vagy hallgató?), ahhoz, hogy elegendő piaci-üzleti érzékkel rendelkezzen szerepköre betöltéséhez, ugyanakkor a szerepkör mélységeibe a hallgatók is minél inkább beleszállhassanak?
  - c) Ki legyen a Scrum mester (pl. oktató vagy hallgató?), ahhoz, hogy elegendő információval és megfelelő képességekkel rendelkezzen szerepköre betöltéséhez, ugyanakkor a szerepkör mélységeibe a hallgatók is minél inkább betekintést nyerjenek?
  - d) Amennyiben néhány hallgató speciális szerepkört tölt be (pl. Scrum mester, termékgyártó), legyenek ezek a hallgatók a Scrum csoport tagjai is egyben, vagy ragaszkodjunk az ortodox Scrum szereposztásához, ahol nincsenek két szerepkört egyszerre betöltő személyek?
  - e) Mennyire számíthatunk arra, hogy a kevés közös fejlesztés és az oktató-hallgató szerepkörök ellenére is önálló, közös felelősséget vállaló Scrum csoport fejlődhet ki a hallgatókból, illetve az egyes személyek is fejlődnek?

### 3.2. Események

A Scrum iteratív fejlesztési módszer. Alap iterációja a futam (Sprint), mely ipari környezetben 2-4 hetet ölel fel. Tervezéssel (Sprint planning) indul, mely az adott futam során megvalósítandó feladatok kiválasztását célozza. Az ezt követő napok mindegyikén Napi Scrum (Daily Scrum) tart a csoport, mely során minden csoport tag három kérdésre válaszol röviden: Mít csináltam az előző találkozó óta? ; Mít fog csinálni a következő találkozóig?; Van-e valami, ami akadályozza?. A futam végi bemutatóval (Demo) a csoport bemutatja a futam során fejlesztett működő szoftver terméket a megrendelőnek, és itt kap visszajelzéseket a termékre vonatkozóan. A futam visszatekintéssel (Retrospective) ér véget, ahol a csoport tagjai megfogalmazzák a futam során tapasztalt erősségeket, gyengeségeket és a jövőre vonatkozó együttműködési javaslatokat.

Kérdések az események kapcsán:

- a) Milyen hosszúak legyenek a futamok, hány futam legyen a két félév során?
- b) Mennyire lesz nehéz egy kevés tapasztalattal rendelkező csoportnak a futamok megtervezése, tekintettel az rendelkezésre álló tervezési időre is?
- c) Milyen sűrűn tudunk napi Scrumot tartani (pl. hetente egy), és mivel segíthetjük ennek hatékonyságát?
- d) Hogyan érhetjük el, hogy a bemutatók valódi komolysággal folynak?
- e) Milyen visszatekintési technikákat alkalmazzunk?

### 3.3. Termékek

A szoftver teméken kívül számos más termék is van a Scrum módszer alkalmazásának. A termék vízió (Product Vision) egy alap elképzelést, célt fogalmaz meg a megvalósítani kívánt termék kapcsán. A termék kívánságlista (Product backlog) a főbb, kívánt funkciókat foglalja össze. Futam cél (Sprint Goal), futam feladatlista (Sprint Backlog) jelentése értelemszerű. A Scrum tábla (Scrum Table) rámutat arra, hogy az egyes feladatok milyen fejlesztési szakaszban vannak (pl. folyamatban, felülbírálatra várva, készen stb.) Különböző diagramok megjeleníthetik a folyamat haladását, és tervezett haladása közötti különbséget. Mindezeket a Scrum termékeket általában egy projekt követő rendszer képes tárolni, megjeleníteni.

Kérdések az termékek kapcsán:

- A kevés együtt-dolgozási lehetőség ellenére létrejöhet a kívánt szoftvertermék a kurzus végére?
- b.) Milyen projekt követő rendszert alkalmazzunk?
  - c.) Adjunk a hallgatóknak termék kívánságlistát, vagy ennek kidolgozását bízzuk rájuk?

- d.) Számíthatunk arra, hogy az elektronikus Scrum tábla, a hallgatók rendszeres adminisztrációja alapján a fejlesztés valódi státusát mutatja?

### 3.4. Értékelés

Az agilis csoportban az egyén értékelésének fontossága helyett a csapat értékelése kap hangsúlyt, hiszen a csapat tagok egyenlők, és közösen felelősek a termékért. Ezzel szemben az oktatásban minden hallgatót személyesen kell értékelni. Felmerül a kérdés: Hogyan történjen a hallgatók egyéni értékelése?

Ezen cikk „Megvalósítás” című fejezetében ezekre a kérdésekre adott válaszokat és azok következményeit témakörönként csoportosítva mutatjuk be a mi esettanulmányunkat illetően.

## 4. Megvalósítás

### 4.1. Szerepkörök

Egyetemi környezetben az oktató szerepköre hasonló az agilis megrendelő szerepköréhez. Bár motiváló megoldás lehet külső ipari partnert vagy egyéb oktatót, szakmai munkatársat felkérni a megrendelő szerepkörének betöltésére egyetemi szoftverfejlesztési kurzus során, megvannak ennek a veszélyei is. (Például, ha nem eléggé együttműködő, elérhető a megrendelő, akkor a teljes kurzus bukásához vezethet.) Esetünkben a kurzust vezető oktatók töltötték be a megrendelő szerepét az agilis projektben. Ezzel biztosítva volt a megrendelő könnyen elérhetőségének kritériuma, viszont a hallgatók nem föltétlen éltek vele. Például ritkán keresték fel az oktatókat részletek vagy éppen felhasználói felület egyeztetése céljából, inkább saját elképzelésük szerint haladtak a fejlesztésben. Úgy tűnt, hogy nem tudták eléggé komolyan venni a megrendelőnek való nem megfelelés kockázatát.

A termékgazda szerepkör kapcsán felmerült, hogy oktató vagy hallgató lássa el. A fejlesztés szempontjából fontos lenne, hogy a termékgazda átfogó műszaki, piaci és üzleti ismeretekkel rendelkezzen, legyen döntésképes stb. – ilyen szempontból az oktatókra esne a választás. Az egyetemi kurzus szempontjából viszont előnyös lenne, hogy a szerepkör betöltése által a hallgatók megtapasztalják és elsajátítsák annak velejáróit. Esetünkben ez utóbbira esett a választás, azzal a kiegészítéssel, hogy a termékgazda szerepkört betöltő hallgatók több figyelmet, segítséget kaptak az oktatóktól, például egy nagyjából kész termékvíziót és termék kívánságlistát, javaslatokat ez utóbbi prioritizálásához. Az első félév során egy hallgató töltötte be a termékgazda szerepét, ő dolgozta ki az elfogadási kritériumokat, ő fogadta el vagy utasította vissza ezek alapján a futam eredményeit. A második félévben egy második hallgató társult hozzá. A hallgatók meglepően pontosak és sokoldalúak voltak az elfogadási kritériumok meghatározásánál, jól megfogták azokat az igényeket, melyeket egy átlagos megrendelő támasztana. Az elfogadási kritériumok megvalósulásának elbírálása kapcsán viszont engedékenyebbek voltak, mint az ezt felülbíró oktatók, feltehetően a kollegiális viszonyok megléte miatt.

A Scrum folyamatok és értékek megvalósulására rendszerint nagy hatással van a Scrum mester hozzáállása, ilyen szempontból döntő fontosságú a Scrum mester szerepkör megfelelő betöltése is. Mivel nem lehet hatalma a csapatagok felett, a Scrum mester szerepkör betöltésére hallgató ajánlott. A kiválasztottnak ugyanakkor mélyen ismernie kell a Scrum folyamatokat, szerepköröket, és a megfelelő kiszolgáló-vezetői képességekkel kell érvényesítenie az agilis értékeket. Az egyetemi kurzusok kezdetén az oktatók többsége nem ismeri a hallgatókat, és ez nagyban megnehezíti a megfelelő személy felismerését. Az első félévben ez a nehézség kiütközött, a Scrum mesternek jelölt hallgató ugyanis „nem érzett rá” szerepére: nem tudta igazán kézben tartani a folyamatokat, nem volt számára természetes, hogy egyféle felülnézetből figyelje a csoportot, nem volt egy lendületet adó, csoportot összefogó alkat. A csoport érezte egy valódi Scrum mester hiányát, és tulajdonképpen természetes módon termelt ki magából egy újat, aki a második félévtől hivatalosan is Scrum mester lett. Ez a

személy szerepkörébe illeszkedően viselkedett, befolyásolási képességével megfelelően vezette a csoportot a Scrum folyamatok mentén, az agilis értékeket megvalósította és szorgalmazta. Az oktatóknak könnyű volt együttműködni vele, ahogyan a hallgatóknak is. A Scrum csoport is egyre inkább csapatként kezdett működni.

A kurzuson résztvevő összes hallgató fejlesztői feladatköröket is ellátott, így tulajdonképpen minden hallgató a Scrum csapat tagja is volt, esetleges egyéb szerepkörei mellett. Bár ez nem ajánlott felállás az ortodox Scrumban, a munkamegosztás kiegyensúlyozottsága, a hallgatók értékelése és a kurzus céljainak érdekében szükségeszerűnek minősült. Az első félév során összesen tíz, a második félév során kilenc tagú volt a csapat, melyből egy tag új volt. Ez kissé hátráltatta a csapatkohézió kialakulását, de megtapasztalhatóvá tette a csapattagok cserélődésének kihívását is, mely ugyancsak ipar közeli élmény. A csapat önszerveződővé válása nehézkesen indult, bizonyára az oktatási rendszer sajátosságainak is köszönhetően (A hallgatók órán kívül kevés időt tudtak együtt tölteni, hiszen szinte mindenkinek más órarendje volt. Kevés korábbi csoport-munkabeli tapasztalatuk volt. Szokva voltak ahhoz, hogy az oktatók instrukcióit kövessék, mintsem, hogy saját maguknak szervezzék meg a munkát.) A fejlesztés során azonban egyre önállóbb lett a csapat (közös kódolást szerveztek, felkeresték egymást, ha kérdéseik voltak, nyomon követték egymás munkáját), és az egyes tagok is átalakulásokon mentek keresztül, „agilisabbá” váltak. Néhány sikertörténet:

- X nagyon nagy technológiai tudással rendelkezett, az ilyen jellegű témákban elsőként hallatta hangját, így kezdetben nagy mértékben elnyomta a többiekét. Később megtalálta érvényesülésének egyensúlyát, és már nem dominált annyira a megbeszéléseken. Technikai tudásával, dicsereteivel, elismeréseivel, melyekkel a többiek felé irányult, ösztönzően hatott a jobb kódminőség elérésében.
- Y technikai tudása alul marad a többiekéhez képest (előzetes tanulmányai miatt). A csapatnak sikerült úgy elosztania a feladatokat, hogy szorgalma és a többiektől kapott segítség révén neki is sikerült teljesítenie a rábízottakat, így a csapat aktív tagjává vált, és sokat fejlődött technikailag is.
- Z körül több gond akadt: hetekre eltűnt, többször jelentett lemaradást, nem követte a levelezéseket, nem tudta, hogy mi a feladata, stb. A csapat csendesen tűrte a panaszokat, és érdekes módon Z (bár nem jelesre, de) teljesítette utóvégre a rábízottakat, és a csapattól sem szigetelődött el.
- W sokáig a csapat egyik legcsendesebb tagja volt. Nem vállalt nagyon sok feladatot, és azt is „halkan” teljesítette. A félévek során magabiztosabbá vált, a későbbi hetekben egyre több feladatra ajánlotta magát, és bekapcsolódott a csoportbeszélgetésekbe is.

Belátható tehát, hogy - mint a csoportmunkák többsége esetén - egyes tagok „húzó tagok” voltak, vagyis több munkát és energiát áldoztak a csapat előremenetele érdekében, mint mások. Ezek a tagok általában pontosan azok voltak, akik a sajátos szerepköröket is bevállalták (Scrum mester, termékgazda). Örvendetes viszont, hogy a szoftverfejlesztés során egyre jobban fejlődött a közös felelősségvállalás a csapatban, a munkamegosztás kiegyensúlyozottabbá vált. Ezt az fejlődést a Scrum módszer jótékony hatásának tudjuk be.

## 4.2. Események

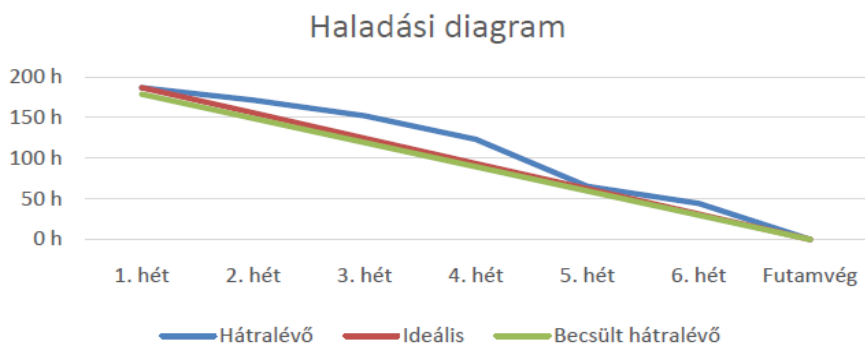
Az egyetemi körülmények természetesen nem csak a Scrum szerepkörök kibontakozásának mértékét, de a Scrum események menetét is nagyban befolyásolták. A közösen töltött idők rövidege miatt a Scrum eseményeknek kissé mintavételszerű, szimulációs ízük volt, viszont úgy gondoljuk, hogy átjött lényegük, ritmikusságuk megvalósult. Az első félév egy előkészítő időszakkal (3 hét) kezdődött, majd egy futammal folytatódott. A második félévben két futam valósult meg, ahol az utolsó stabilizációs futam volt.

A futamtervezés az első alkalommal bizonyult a legkörülményesebbnek. A hallgatók teljesen elmerültek a technológiai részletekben, nem érezték még szerepkörüket (elméleti ismereteket egy korábban teljesített kurzus során szereztek). Ennek következményeképpen a tervezés a fejlesztés megkezdése utánig húzódott, illetve a felhasználói történetek súlyának becslésére nem került sor közösen – ezt az oktatók pótolták. A második félév során viszont a hallgatók megismerkedtek a tervezési pókerrel (planning poker), és egy pár felhasználói történet (User Story) felhasználói történet pont számát közösen határozták meg, a többit pedig a termékgyázdák. Fejlődés volt megfigyelhető az elfogadási kritériumok kapcsán is, melyeket ez alkalommal időben és megfelelő részletességgel tisztáztak.

A napi Scrum mindennapi jellege értelemszerűen nem tudott érvényesülni a kurzus egyetemi körülményei miatt, helyette a heti rendszerességgű találkozók során valósult meg, a közösen töltött másfél óra elején, ami egyébként összhangban volt azzal, hogy a hallgatóktól körülbelül egy hét alatt vártunk el egy munkanapnyi (8 órányi) munkateljesítményt. Bár megvolt a kísértés az elkalandozásra, a hallgatók idővel megszokták, hogy tömören nyilvánuljanak meg a napi Scrum alatt, és a sikeresen felszínre került elakadásokat a gyakorlat későbbi részeiben tárgyalják meg. Sokkal kevésbé ment át, hogy nem a Scrum mesternek vagy az oktatóknak, hanem egymásnak kell beszámoljanak, annak megfelelően, hogy ők -mint agilis csapat- elsősorban egymásnak tartoznak felelősséggel. Nem segítség e tekintetben a tanterem szokványos berendezése sem (a hallgatók egymás háta mögötti sorokban ülnek, az oktatók szemben). A körbe vagy U alakban rendeződés - a második félév során - sokkal kedvezőbbnek bizonyult.

A bemutató fogalmához általában, a hallgatók viszonylag szokva vannak az egyetemi kurzusok kapcsán, viszont azok megvalósítása legtöbbször lazább keretek között mozog. Csoportunk esetén sikerült kidomborítani a megtervezettség, pontosság, időkorlátok fontosságát a Scrum bemutató viszonylatában. Ennek módszere a többszöri figyelemfelhívás, próba bemutató tartása, illetve egy ünnepélyes záró bemutató megszervezése volt. Ez utóbbi esetében vendég oktatók vettek részt a kurzuson, és a hallgatók képviselve körütekintően és látványosan bemutatta a csoport által fejlesztett működő szoftvert, a célra létrehozott adatok és menetrend segítségével. Az egyetemi kurzus kellemes és vidám hangulatban ért véget.

Az oktatásban különösen megvan a helye a Scrum visszatekintőknek is, hiszen ez a csapat fejlődésének záloga. Az első visszatekintőt az oktatók vezették, pozitív és negatív visszajelzéseket megfelelő színű papírlapok segítségével csoportosítva. Ezek a tábla felületére kerültek. Olyanok voltak, mint: „túl sok technikai kihívás”, „nem elég hatékony kommunikáció”, „számonkérés hangulatú napi Scrum”, „érdekes módszertan”, „segítő közösség”, „elégedettséget kiváltó eredmény” stb. A visszatekintés nyomán sikerült pár ötletet is megfogalmazni a sikeresebb jövőbeli együttműködés érdekében. A második futam végén tartott visszatekintést már a Scrum mester vezette, és diagramokkal színesítette azt. A hallgatók nagy érdeklődéssel figyelték az összesítő adatokat, rajzokat – kíváncsisággal néztek vissza a közösen megtett útra. Példa diagramra az itt bemutatott 1. ábra, mely a Scrum mester alkotása.



1. ábra: A második futam haladási diagramja

### 4.3. Termékek

Az egyetemi, csoportos szoftverfejlesztési kurzus során a hallgatók egy közösségi hírfolyam létrehozására alkalmas rendszert valósítottak meg, mely olyan felhasználókat vonzhat be, akiknek információmegosztási igényeik vannak a velük szakmailag vagy közös érdeklődési kör által összetartozó személyekkel. (Fejlesztő eszközök voltak: Java Platform, Standard Edition, Hibernate, Eclipse, MySQL, Tomcat.)

A termékvizíót az oktatók biztosították – ez nagyon hasznos is volt, mert ennek hiányában sokat csúszhatott volna a fejlesztési munka elkezdése. Hasonlóan nagy segítséget jelentett, hogy egy tömör felhasználói történet lista is adott volt. A termékgazda ennek alapján, egy minta segítségével dolgozta ki a részletesebbet. Az idő hiánya miatt azonban az elfogadási kritériumok rögzítésére a fejlesztés megkezdése után került sor, és ez kisebb csalódásokat vont maga után. (Egy-egy késznek ítélt funkcionalitáshoz újra hozzá kellett nyúlni.) A futamcélra nem került hangsúly, a futam feladatlistája alapvetően a felhasználói történet lista egy részhalmaza volt. A Scrum tábla fizikailag nem valósult meg, de a Jira (6) projektkövető rendszerben látható volt. A hallgatók esetében nagyon nehéz volt elérni, hogy folyamatosan frissítsék az adminisztráló felületet, így a Scrum tábla állapota nem mindig tükrözte a fejlesztés valódi állapotát. Ilyen módon a rendszer által kigenerált haladási diagramok sem tudtak hitelesek lenni.

A Scrum termékeken kívül említésre méltónak tartjuk a létrejött 59 teszt forgatókönyvet, melyből 58 lett végrehajtva, ennek során 45 felvett hiba keletkezett, és ezek javítására is sor került.

Oktatási szempontból fontos termék az a Wiki oldal, mely a projekt architektúrális döntéseit, fejlesztői segédanyagokat, személyes adatokat tartalmaz. A bejegyzéseket az oktatók és a hallgatók közösen hozták létre (például hallgató által létrehozott, fejlesztőeszközt bemutató kiselőadás). Az ezt kiegészítő levelezési lista is a fejlesztés dokumentuma, egyféle naplója.

### 4.4. Értékelés

Tekintve a Scrum fejlesztés sok rétegét, és ezek közül leginkább a csapat, mint egység elvét, felmerül a kérdés: mi lenne a hallgatók megfelelő egyéni értékelésének a módja? Esetünkben az oktatók igyekeztek egy viszonylag pontos rendszert kidolgozni, melyben a személyek értékelésében a saját és a csapatuk teljesítménye is helyet kapott, fele-fele arányban. Bátorkodtak a csapattagok egymásról nyilatkozott értékelését is beleszámolni a végső jegybe, ahogyan a csapat önértékelését is. (Ezeket az információkat titkos módon gyűjtötték be.) Az első félév után szembesülniük kellett azzal, hogy a hallgatók eléggé „kollegiálisan” viszonyultak egymáshoz, így minden csapattársukat körülbelül ugyanolyan mértékben és magasan értékelték. Ebből a tapasztalatból kifolyólag, az oktatók úgy döntöttek, hogy a második félév folyamán kevesebb pontszám szerezhető meg a többi csapattag értékelés nyomán, a tanárok szubjektív véleményének beszámítási aránya emelkedett. Ez utóbbi kategóriában az oktatók a Scrum szerepkörök betöltésének milyenségét és az egyéb külön feladatok (pl. a csoport tudásának felzárkóztatása egy adott témakörben) bevállalását is díjazták. A két félévben használt értékelési stratégiát az 1. táblázat foglalja össze.

Egyéni teljesítmény megítélése (összesen 50 pont értékben)	1.félévben	2.félévben
A többi csapattag értékelés (titkos)	20 pont	5 pont
Jira-ból kinyerhető objektív információk alapján	20 pont	20 pont
Tanár szubjektív értékelése (aktivitás, szerepek, stb.)	10 pont	25 pont
Csapat teljesítmény megítélése (összesen 50 pont értékben)	1.félévben	2.félévben

A projekt eredménye alapján (bemutató, mérőszámok)	25 pont	25 pont
Folyamatok alapján (visszatekintés, mérőszámok)	20 pont	20 pont
A csapat önértékelése	5 pont	5 pont
Osztályozás összpontszám alapján: 50-59 pont között elégséges, 60-69 közepes, 70-79 jó, 80 pont felett jeles.		

1. táblázat: A hallgatók értékelésének szempontjai

## 5. Visszajelzések

A csoport szubjektív értékelése kapcsán az egyik oktató elégedettségéről számolt be, szerinte a termék jól összeállt a fejlesztési szakasz végére, és a csapat is önszerveződővé, önállóvá formálódott. Ha valamin módosíthatna, akkor több időt biztosítana a hallgatók számára az közös munkára, vagyis növelné a kurzus heti óraszámát. A csoport másik oktatója a fejlesztési technológia korábbi rögzítésével vívna ki több időt és erőt a konkrét fejlesztési feladatokra egy jövőbeli csoport esetén.

A csapattagok csapatteljesítményre vonatkozó értékelésének átlaga mindkét félévben 4.22 volt. (Értékelési skála: 1-től 5-ig, ahol az 5 a kiválót jelöli.) Érdekes módon az önértékeléseik átlaga az első félévben 4.44, a második félévben 4.00 volt. Egyéb izgalmas visszajelzések is voltak, például: a hallgatók a projekt feladat megfelelését emelték ki a legjobban (8.33), emellett fejlődést véltek felfedezni a módszertan értékeinek elsajátítása (1.félév: 7.11, 2.félév: 7.56) és a csapat közérzetének alakulása kapcsán is (1.félév: 7.00, 2.félév: 7.44). (Értékelési skála: 1-től 10-ig, ahol az 10 a kiválót jelöli.)

A feleletbekérő kérdések segítségével az is kirajzolódott, hogy a hallgatók megterhelőnek érezték a fejlesztés kezdetén a sok technológiai újdonságot, ahogyan a kommunikációs kihívásokat is. „Szerencsés volna, ha a heti megbeszéléseken túl is találkoznánk, ha intenzívebb tudna lenni a közös munka.” – fogalmazta meg az egyik csapattag. Egy másik visszajelzés: „Sajnos nem megvalósítható, hogy ennél is több futam legyen a gyakorlat során, de szerintem az agilitás előnyei igazán hosszú távon tudnának megmutatkozni.” Valaki szerint a legjobb tapasztalat az volt: „Hogyan tud együtt dolgozni egy nagyobb csapat.” A korábban tárgyaltak alátámasztják: a hallgatóknak minden bizonynyal sikerült is együtt dolgozni, és az adott egyetemi körülmények között meglehetősen jó csapatot alkotni. Hiszük, hogy ennek megvalósulásához nagy mértékben hozzájárult az agilis módszertan alkalmazása.

## 6. Következtetések

Az iparban virágzó agilis módszertanok optimális alkalmazása az egyetemi kurzuson folyó szoftverfejlesztés során nem triviális feladat. A szűkös idő és térbeli keretek megnehezítik a kommunikációt a csapattagok között és a jó gyakorlatok, szerepkörök kibontakozását. 2016 tavaszi és őszi félévében az ELTE IK két oktatója úgy döntött, hogy vállalja ezt a kihívást az ígérező előnyök reményében. Kísérletük kimenetele a következő ajánlásokkal fogható össze.

Az agilis megrendelő szerepkör megvalósítására egy lehetőség az oktató, aki kellően elérhetőnek és agilisnak bizonyulhat az oktatási rendszer sajátosságaiból adódóan, de ez a kettős szerepvállalás el is nyomhatja a megrendelő szerepkörének kibontakozását. A termékgazda és a Scrum mester szerepkörök megvalósítása esetén az azt elvállaló hallgatók rendszerint sokat fejlődnek, és nagyon értékes tapasztalatokra tesznek szert, viszont a csapat és a teljes kurzus szempontjából nagyon fontos kezdettől a megfelelő személyek, személyiségek kiválasztása, ami elég nehéz, hiszen az oktatónak



nem nagyon vannak előzetes információi a csapat tagjairól. A Scrum mester esetén különösen fontos a szolgáló-vezetői képességek megléte, a termékgazda esetén a körültekintő üzleti, piaci érzék. A Scrum csapat önállóságának és közös felelősségvállalásának kifejlődése várhatóan lassú folyamat, de fontosságának hangsúlyozásával érdemes ezt erősíteni.

A termékvízió és a tömör termék kívánságlista biztosítása a kurzus elején jó oktatási módszer, mely segít abban, hogy a fejlesztés mielőbb megkezdődhessen, a tervezés minél gördülékenyebben haladhasson. Ebben a fejlesztői eszközök oktató általi, korai rögzítése is segítség lehet. A becslés, a tervezési póker alkalmazása a tervezés során érdekes tapasztalat a hallgatók számára, ezért ezt legálább ízelítő szintjén érdemes kipróbálni. A napi Scrum során használjuk ki, hogy ez az agilisabbá válás legkézenfekvőbb eszköze. Sokat segíthet, hogyha megfelelő térbeli elhelyezkedéssel erősítjük (például körbeállunk). Szorgalmazzuk, hogy legalább a napi Scrummal egy ütemben frissüljön a feladatkövető rendszer Scrum táblájának állása is, hogy az nagyjából hiteles haladási diagramot tudjon kigenerálni. A bemutatók kapcsán kérjünk felkészültséget, és erősítsük fontosságukat vendégek meghívásával. A visszatekintő ne maradjon el, sőt: használjuk ki az oktatás, kurzusmenet hatékonyságának növelése érdekében is. A hallgatók motiválását illetően a megfelelő fejlesztési feladat és technológia kiválasztása is fontos szempont.

Az értékelést illetően érdemes szem előtt tartanunk, hogy nemcsak az egyéni teljesítmény, hanem a csapat teljesítménye is beszámítható, ezzel is hangsúlyozva az agilis értékeket. Amennyiben a hallgatók egymást is értékelik, kísérjük ezt fokozott figyelemmel, szükség esetén pedig bíráljuk felül stratégiánkat.

A Scrum módszer egyetemi szoftverfejlesztési kurzuson történő alkalmazása sok izgalmas kérdést vet fel, számos döntéshelyzetbe hoz. Irodalma még korántsem gazdag. A szerző további kísérletek és azok bemutatása fele irányul a jövőben, abból a meggyőződésből kifolyólag, hogy az agilis módszertanok egyetemi kurzuson való alkalmazása még így, gyerekcipőben járva is hatékony. Meglátásairól részletesebben beszámol a (7) forrásban.

## 7. Köszönetnyilvánítás

Ez a cikk nem jöhetett volna létre dr. Langer Tamás bátor ötlete és annak körültekintő kivitelezése nélkül, melyet a „Szoftverfejlesztés a gyakorlatban” tárgy oktatójaként valósított meg. Megtiszteltetés, hogy kutatásának részese lehettem, és annak szélesebb körű bemutatásával és folytatásával foglalkozhatom. Továbbá köszönetemet fejezem ki Horváth Attila oktatónak és a kísérletben résztvevő minden hallgatónak is.

## 8. Szakirodalom

1. *Agile Manifesto*. [Online] [Hivatkozva: 2017. október 15.] <http://agilemanifesto.org/iso/hu/manifesto.html>.
2. *ISTQB Exam Certification*. [Online] [Hivatkozva: 2017. szeptember 30.] <http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/>.
3. *Incorporation of Agile Development Methodology into a Capstone Software Engineering Project Course*. Kuhl, Jon G. Iowa City : ASEE, 2014. The 2014 ASEE North Midwest Section Conference. old.: 1-3.
4. *Eötvös Loránd Tudományegyetem Informatikai Kar*. [Online] [Hivatkozva: 2017. május 28.] <http://www.inf.elte.hu>.
5. Scrum Alliance. *Scrum Alliance*. [Online] [Hivatkozva: 2017. október 20.] <https://www.scrumalliance.org/>.
6. *Atlassian*. [Online] [Hivatkozva: 2017. október 30.] <https://www.atlassian.com/software/jira>.

7. Ilyés, Enikő: *Agilis szoftverfejlesztési gyakorlat megvalósításának lehetősége az ELTE Informatikai Karán*. Budapest : Ilyés Enikő, 2017.