

Projektmenedzsment és az agilis szoftverfejlesztés

CSUTORÁS ZOLTÁN

Adaptive Consulting Kft.

zoltan.csutoras@adaptiveconsulting.hu

KOCSIS ÁRPÁD

Nissan Europe Information Systems

akocsis@nissan-europe.com

Kulcsszavak: projektmenedzsment, agilis, agile PM, SCRUM, szoftverfejlesztés, projekt

Az agilis szoftverfejlesztés és a projektmenedzsment együttműködése egy új és érdekes téma, amelyről egyre többet hallani. A kor kihívásaira válaszolva 2011 második felétől megjelenik a PMI Agile minősítés¹. Vajon milyen viszonyban áll a projektvezető és a szoftverfejlesztő, hogyan lehet összeilleszteni az agilitást a jól bevált projektvezetési módszerekkel?

1. Bevezetés

Mind az agilis szoftverfejlesztés, mind a projektvezetés olyan terület, ahol temérdek könyv, tanfolyam és cikk áll rendelkezésre. Mind a két terület rendelkezik szervezett képzési és minősítési rendszerrel. Azt szeretnénk megvizsgálni, hogyan fog e két terület a valós életben, a gyakorlatban találkozni, amikor a szoftvert agilisan fejlesztik projektszerű keretek között.

Arra a kérdésre keressük a választ, milyen egy igazi, „éles” agilis szoftverfejlesztés üzleti környezetben, azaz olyan vállalatok kontextusában, ahol az IT üzleti célokat szolgál ki, tehát a kiszolgáló folyamatok része. A szoftverfejlesztést a vállalat szemszögéből vizsgáljuk.

A szoftverfejlesztés az ICT és IT területen működő cégek esetén (pl. Nokia, Microsoft, Apple) – tehát ahol a szoftver maga a termék – kicsit más, bár sok megállapítás ebben a környezetben is megállja a helyét. Ugyanígy féltetesszük a kutatás-fejlesztési (R&D) területet. Először arra az esetre koncentrálnunk, ami tipikus lehet egy informatikai cég, egy kkv számára: vállalati környezetben történő üzleti célú szoftverfejlesztés. Nem teszünk különbséget belső, külső vagy kiszervezett fejlesztés között – a cikk megállapításai érvényesek mindhárom esetben.

A célunk az, hogy rámutassunk az összefüggésekre, ok-okozati viszonyokra és azokra a kényszerekre, amelyek mentén az agilis szoftverfejlesztésnek a projektvezetéssel együtt kell mozognia.

2. A vállalati környezet

A környezet leírására a MOST piramist használjuk némi módosítással (1. ábra).

A vállalatot a Tulajdonos érdekei mozgatják – a Tulajdonos *profitot* szeretne termelni. A Menedzsment határozza meg a *célokat* és a *stratégiákat*. A stratégia megvalósítása taktikai szinten *projekteken* keresztül törté-

nik, amelyeket a PM vezet a megfelelő projektvezetési módszertan alkalmazásával. A munkát az informatikusok végzik, azaz ők fejlesztik ki a *szoftvert*. (Természetesen létezik projekt informatikus nélkül is, de az most számunkra nem érdekes.)

Konkrét példa autóipari környezetből:

Küldetés (Mission)	Újautó értékesítések növelése
Feladat (Objectives)	Modellválaszték bővítése
Stratégia (Strategy)	Új modell indítása B szegmensben
Taktika (Tactics)	Projekt indítása az új modell értékesítésének támogatására
Projekt munka	Új modellhez szükséges fejlesztések végrehajtása az értékesítési rendszeren

A hatalom gyakorlása, az érdekek érvényesítése felülről lefelé történik, azaz a tulajdonos céljai adják a menedzsment feladatait, a menedzsment által szabott célok szerint dolgozik a projektvezető, és a projektvezető ad feladatokat az informatikusoknak. A fejlesztői csapatok feladata a felülről meghatározott célok elérése, a feladatok végrehajtása.

1. ábra Vállalati piramis

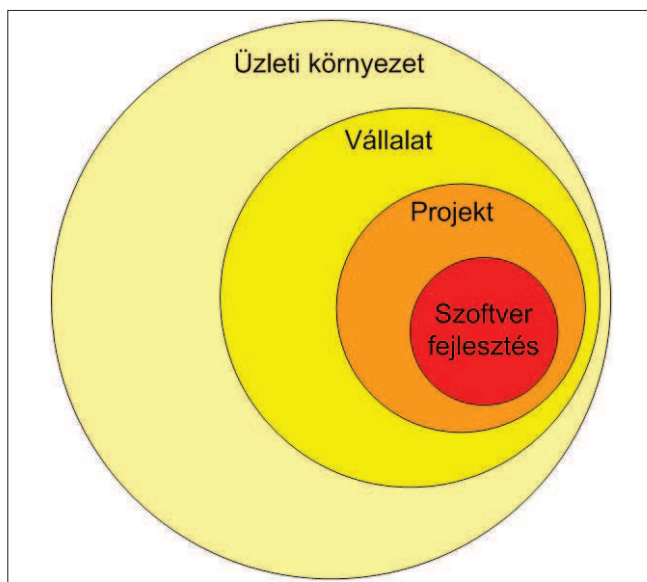


A tulajdonos, a menedzsment és a projektvezető elvárása a projektekkel szemben a *kiszámíthatóság, tervezhetőség és a keretek között maradás* (time-budget-scope-quality). A fejlesztő áll a piramis alján és jól látható, hogy alkalmazkodni kényszerül a felette meghatározott célokhoz és tervekhez. Illetve ha nem akar, akkor majd keresnek másik fejlesztőt.

Az IT projektnek ebben a közegben kell léteznie, és az informatikusnak alkalmazkodnia kell a vállalati környezethez. Más út nincs.

3. Projektmenedzsment

Vállalati környezetben a változás eszköze a projekt – legyen szó bármilyen változásról és bármilyen ipárgőről. Minden nagyobb fejlesztési feladat projektszerűen zajlik. A projekt környezetet a 2. ábra szemlélteti.

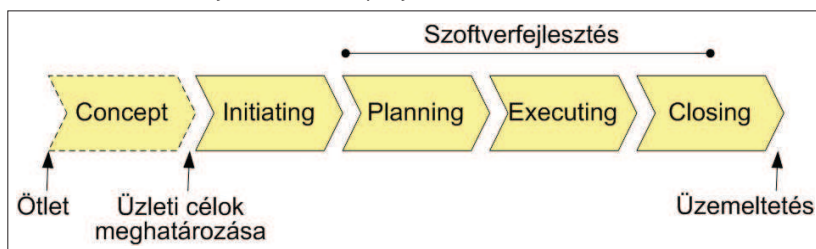


2. ábra Üzleti környezet

A projektek vezetése módszertanok alkalmazásával történik. A nagyvállalatok kialakították maguk módszertanát, amely a nemzetközi szabványok, például PMI ajánlás [1] vagy PRINCE2 adaptációját jelenti.

Az informatikai feladatok szükségszerűen egy üzleti projekt részeként valósulnak meg, annak keretein belül. Azonban a projekt és a szoftverfejlesztés nem azonos! A projekt jóval azelőtt elkezdődik, mielőtt a fejlesztők nekiállnának dolgozni és nem fejeződik be ott, amikor a szoftver elkészül (3. ábra) [2].

3. ábra Szoftverfejlesztés és a projekt



Jól látható, hogy még mielőtt a fejlesztőktől megrendelnék a szoftvert, a projektet fel kell építeni. Illetve látható, hogy a kész szoftver még nem elég, azt át kell adni az üzemeltetésnek, illetve stabilizálni kell az üzemeltést. Az előző szakaszban említett példához visszatérve: a projekt nem akkor van kész, amikor a szoftvert átadták, hanem amikor az új autómódellem értékesítése gond nélkül zajlik.

A munka a szigorúan vett fejlesztési feladatoknál jóval szélesebb, a keretek a szoftverfejlesztés megkezdésekor adottak. A rendszerfejlesztéshez kapcsolódó tradicionális szabványok és módszertanok (pl. ISO 12207) figyelembe veszik a rendszerfejlesztésnek ezt a tágabb értelmezését.

4. Módszertanok és szemléletmódok

Az üzleti környezet meghatározása és a projektmenedzsment után most essék szó a szoftverfejlesztésről. Amikor vízéséről vagy agilis fejlesztésről beszélünk, akkor tulajdonképpen nem egy-egy módszertanról van szó, hanem szemléletmódról.

A vízésés vagy PPP szemlélet lényege a munkafolyamatok fázisokra bontása (ezért is használják rá a PPP – phased product planning – elnevezést) és a tervezés fontossága. A vízésés kifejezést ennek a szemléletmódnak az elnevezésére használjuk a továbbiakban és beleértjük mindazon módszertanokat, amelyek megfelelnek a definíciónak. Az *agilis modellt* ezzel szemben arra szemléletmódról használjuk, amely az egyénekbe és a csapatba vetett bizalomra épül, elfogadja a fejlesztési folyamatokban lévő bizonytalanságot és ezért ciklikus fejlesztési megközelítést javasol.

A vízésésmodell kora és kialakulása miatt jól összekapcsolható a projektmenedzsment módszerekkel. Ugyanakkor a projektmenedzsment módszertanok és ajánlások nem mondják azt, hogy csakis vízésésmodell szerint lehet szoftvert fejleszteni. Már csak azért sem, mert a projektvezetés és a szoftverfejlesztés a projekt különböző szintjeit jelentik (lásd az 1. ábrát).

5. Az agilis szoftverfejlesztés

Az agilis szoftverfejlesztésre elsősorban mint *értékrendszerre* érdemes tekinteni. Az agilis kiáltvány és a 12 agilis alapelv is röviden és világosan megfogalmazott értékrendszert rögzít. Ennek az értékrendszernek a lényege a gyorsaság, a változásra való reagálási képesség, az egyének és a csapat képességeibe és motivációjába vetett bizalom, a működő terméknek, mint a siker egyetlen mércéjének elismerése. Az agilis szemlélet nem más, mint a értékrendbeli hangsúlyok erős megváltoztatása a vízésés szemlélethez képest. Amíg a vízésésszemlélet kiindulási pontja, hogy az a team, amelyik jól kidolgozott eljárásokat, szabályo-

kat és szervezeti felépítést követ, hatékony lesz, addig az agilis szemlélet abból indul ki, hogy ha a *megfelelő emberekből* összeállított team elé *világos célokat* tűzünk ki, és *világos kereteket* jelölünk ki számukra, akkor azok *hatékony eljárásokat*, szabályokat és szervezeti felépítést fognak kialakítani. A különbség tehát a kultúra és a team kialakításának sorrendjében van, nem pedig abban, hogy szükség van-e szabályokra. Az agilis szemlélethez igazodó modellek egy olyan keretrendszer definiálnak, amelyek azt írják elő a megvalósító csapatok számára, hogy tudatosan és megállás nélkül vizsgálják felül saját működésüket és a termékkel párhuzamosan saját szabályaikat és eljárásaikat is folyamatosan fejlesszék. Ezek a keretrendszerek nem a termék megvalósítására vonatkozó módszertanok, hanem olyan szabályok és szervezeti keretek, melyek az egyedi problémákra testre szabott eljárások kialakítására készítetik a megvalósító teamet.

Az agilis szoftverfejlesztési alapelvek mentén számos módszertani keretrendszer alakult ki. Ezek közül a legismertebb a Scrum [3], de vannak más érdekes irányzatok is, például az eXtreme Programming, a DSDM, vagy a Kanban System for Lean Software Development. Az agilis keretrendszerek sosem lesznek módszertanok, mivel az alapelvük az igényekre történő adaptáció és a folyamatos javítás érdekében történő állandó változtatás.

Az agilis módszerek hatékonysága akkor mutatkozik meg igazán, ha a projekt célja egy új (még nem létező) termék fejlesztése. Ebben az esetben nincs hová visszanyúlni, alig állnak rendelkezésre tapasztalati alapok, tehát nincs okunk azt hinni, hogy létezik olyan módszer, ami az új problémára megoldást tud kínálni. Ezekre a projektekre az a jellemző, hogy csak homályosan ismerjük az elkészítendő termék körvonalait, nem rendelkezünk kellő információval a pontos specifikációhoz, nincsenek tervezési mintáink a tervek elkészítéséhez és nem tudjuk előre azonosítani azokat a tevékenységeket, amelyek az új termék előállításához fognak vezetni.

A Scrum egyik ihletőjeként számon tartott „The New Product Development Game” című cikkében [4] pont olyan projekteket és csapatokat vizsgált, amelyek ilyen, instabil elvárások mellett értek el kiemelkedő eredményeket. A feladatok, amelyekre ezeket a teameket létrehozták, például ilyenek voltak: *„Ki kell fejleszteni egy olyan nyomtatót, ami a cég jelenlegi csúcskategóriás nyomtatóinak paramétereivel rendelkezik, de a gyártási költsége annak maximum a fele. A termék kifejlesztésére a team 24 hónapot kap, pont fele annyit, mint a termék elődjének kifejlesztésére felhasznált idő.”* (Fuji-Xerox, FX-3500 projekt) Az ő általuk „rugby módszernek” nevezett alapelvek szolgálták kiindulási pontként a Scrum keretrendszer kialakításához.

Fontos tehát kiemelni két olyan tény, ami meglátásunk szerint általában nem kap kellő hangsúlyt az agilis módszerek tárgyalásakor. Az első, hogy a Scrum elvei szerint szervezett teamekkel szemben eredetileg igen kemény, *köbe vésett határidő, költség, minőség és cél* (nem követelmények/scope) elvárásokat támasztottak. A szabadsági fokuk a cél elérésének módjában volt. A másik,

hogy ezeket a teameket olyan termékfejlesztési feladatok megvalósítására alakították, melyek *jelentős mértékű innovációt* igényeltek. Ezekre a projektekre az volt a jellemző, hogy ismeretlenek voltak a módszerek, amelyekkel a cél elérhető lett volna és nem volt világos koncepció a célt kielégítő termék jellemzőire vonatkozóan.

Ha az agilitást úgy értelmezzük, hogy az a teamek felhatalmazása a termék jellemzőinek megfogalmazására és a saját munkamódszereik kialakítására, akkor az agilitás kívánatos szintje az innováció mértékétől, azaz a megvalósítandó termékkel szemben támasztott elvárások előre történő megismerhetőségétől függ. Minél inkább biztosak vagyunk abban, hogy pontosan mit szeretnénk előállítani és ezt hogyan kell megtennünk, annál kevésbé szükséges az agilitás. Ekkor az energiánkat nem arra kell fordítani, hogy teljesen új munkaszervezési módszereket dolgozzunk ki. (A módszerek javítása ugyanakkor továbbra is fontos kell, hogy legyen!) Ezzel ellentétben, ha jelentős az innováció a projektben, nincsenek tapasztalatok és minták a termék koncepciójának részletes meghatározásához, akkor majdnem biztosak lehetünk benne, hogy a szigorúan fázisolt PPP módszerek kudarchoz vezetnek. Ekkor az agilis szemlélet eszköztárához kell nyúlnunk.

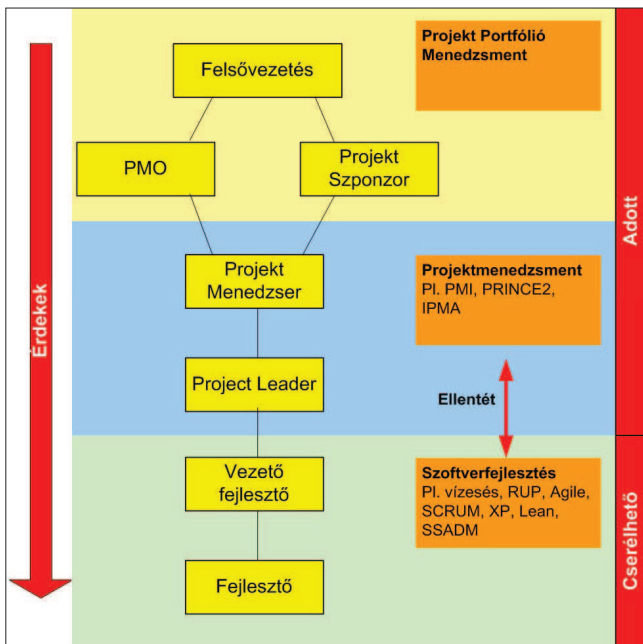
6. A látszólagos ellentmondás

Mi fog történni akkor, amikor a Scrum Master (tegyük fel, hogy a fejlesztés Scrum szerint történik) összeül megbeszélni a projektmenedzserrel a munka indítását? Bábéli zűrzavar lesz. A Scrum Mastert felkészítették arra, hogy hatékonyan irányítsa egy agilis szoftverfejlesztést, de arra nem, hogy egy klasszikus projekt keretein belül dolgozzon. A Scrum Master képzések jellemzően nem szólnak a projektek tágabb környezetéről.

A másik oldalról nézve, a projektvezetőnek nem mondták el, hogy léteznek agilis módszertanok, ezek mit jelentenek és mi az ő szerepe egy Scrum fejlesztésben. Az agilitás szemléletidegen lesz. Az ellentéteket tovább fokozzák a *terminológiai eltérések*. Például a tervezés (planning) kifejezést mind a két oldal használja és mind a két oldalon mást jelent. Szoftverfejlesztés során a tervezés alatt a szoftver műszaki és ütemtervének kialakítását értjük. Ez a munka azonban projektvezetési szempontból már a végrehajtás (execution) része, nem pedig a tervezési fázisé. A félreértések oka az, hogy a Scrum Master és a projektvezető *különböző szemléletet* képvisel, ennek megfelelően más terminológiát, más eszközöket és más folyamatokat.

A 4. ábra mutatja be a különböző módszertanok helyét és szerepét. A fejlesztő és vezető fejlesztő szintjén szoftverfejlesztési modellről beszélhetünk, miközben a projektvezető az ő szintjén projektvezetési módszereket használ. Az egész pedig csokorba fogja a Portfólió Menedzsment.

Továbbmenve: ahogyan az agilis fejlesztésről általában beszélnek, az *ellentmondásban van a projektvezetési módszertanokkal*. Ilyen például a tervezés fontossága:



4. ábra Módszertanok kavalkádja

minőségbiztosítási szempontból kulcsfontosságú a projektterv megléte, miközben az Agile Manifesto szerint ez másodlagos [5]. Ugyanilyen nézőpontbeli eltéréseket találunk, ha a szerződés kidolgozottságáról, a dokumentáció szükségességéről vagy az igények előzetes megismeréséről beszélünk, csak hogy néhányat említsünk.

Az eltérések oda vezetnek, hogy barikád emelkedik az informatikusok és a projektvezetők között, és mindkét oldal próbálja meggyőzni igazáról a másikat. Harcolni nem érdemes, hiszen a projektszerű működés adottság, amit el kell fogadni. Az üzleti célokat el kell érni – és az informatikus lecserélhető.

7. A vízésés modell

Ha az agilis szemlélet konfliktusokat teremt, akkor nem lenne-e jobb vízésésmodell szerint fejleszteni? Elvégre ez a szemlélet kiszolgálja a vezetőség igényét a kiszámíthatóság és tervezhetőség iránt.

Ha ma, a 21. században megnézzük a „hagyományos” módszerekkel dolgozó fejlesztő csapatokat, akkor kiderül, hogy amit a 20. században gondoltunk vízésés módszer alatt, az már nem állja meg a helyét. Ennek egyik oka a változás: *nincs projekt változás nélkül*, nincs szoftverfejlesztés változásokérem nélkül. Az üzleti élet felgyorsult, az igényeket követni kell. A projektvezetési módszertan szerves része a változáskezelés [6]. Tehát pontosan a projektvezetés lesz az, ami rugalmasságra kényszeríti a kötött módszertant. Következmény: a specifikáció nincs kőbe vésve.

A másik ok pedig a hatékonyság: a nagyvállalatok felismerték, hogy a kis csapatok rugalmasabban és nagyobb hatékonysággal képesek szoftvert fejleszteni, mint a nagyok. Manapság már nincs olyan, hogy 100 fejlesztő egy nagy irodában ülve dolgozik egy 1000 oldalas specifikáció alapján. A kiszervezés megváltoztatta a munkamódszereket és a működési környezetet. Ez a vízésés már nem az a vízésés.

Végeredményben azt láthatjuk, hogy a merevnek tartott, folyamat alapú fejlesztési módszertanokat is rugalmasabban kezeljük, azaz *tetten érhető az agilizálódás*. Az agilizálódás pedig pontosan felülről lefelé, a vezetőség irányából jön, akik a változó üzleti igényeknek megfelelő, jól működő szoftvert szeretnének – és mindezt holnapra.

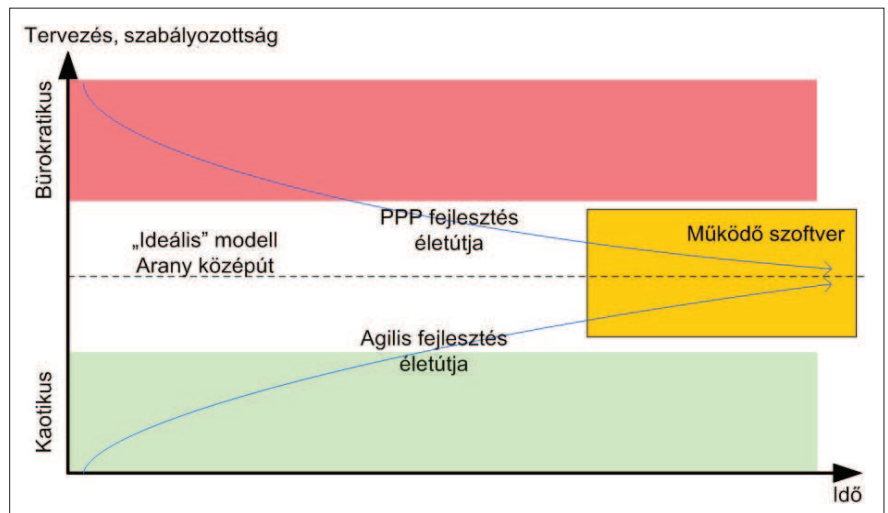
8. A két modell találkozása

A szoftverfejlesztési keretrendszerek és modellek csak eszközök, amelyek a jó projektvezető eszköztárának a részei. Nem szembeállítani kell őket, hanem alkalmazni az adott helyzetnek megfelelően, megtalálva az arany középutat.

Bármelyik módszerrel is indul el, egy sikeres vezető nagyon hasonló gyakorlati alkalmazásra fog jutni. A vízésés is lehetőséget ad a változáskezelési eljárásokra. Ha ezek gyakoriak, ha a tervezés a valós tudásra és nem erőltetett feltételezésekre épül, akkor működni fog és meglehetősen sok „agilis” elem köszön majd vissza belőle. Az agilis szemléletben induló csapatok pedig minden iteráció után alakítanak a szabályaikon, egyre szervezettebbek és szabályozottabbak lesznek. Az igazán jó agilis csapatok egy idő után legalább annyi szabályt és normát vezetnek be, amennyit egy vízésés modellre alapuló módszertan is megirigyelne.

Az 5. ábra mutatja be a két modell találkozását. A vízésés módszertanok a bürokratikus szakaszból indulnak, de az üzleti elvárásoknak engedve az idő előrehaladtával a projektvezető és csapata kénytelen lesz rugalmasnak lenni.

5. ábra Fejlesztési modellek találkozása



A másik oldalról indulva, az agilis keretrendszerek kaotikusnak tűnhetnek eredeti formájukban, de ahogyan a csapat egyre több szabályt alakít ki, lesz egyre bürokratikusabb.

A kétféle szemlélet közelíteni fog egymáshoz.

Ez elsősre meglepő, de másodjára már teljesen logikus, a következő okok miatt:

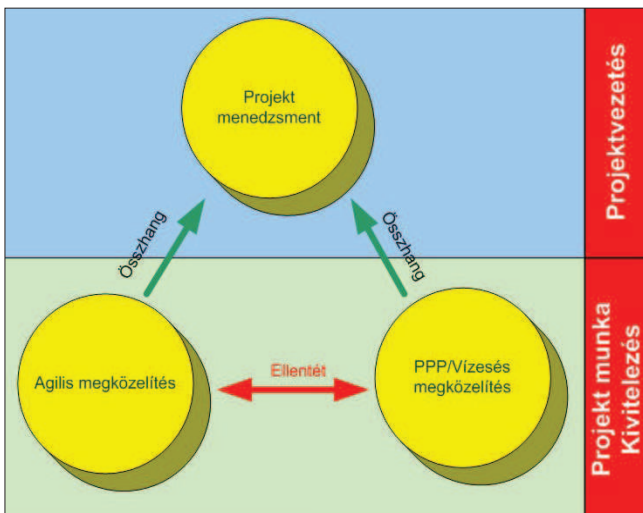
- A cél azonos: működő szoftver, elégedett ügyfél.
- A 2. szakaszban leírt vállalati környezet azonos, tehát gyakorlatban a megvalósításnak is hasonlóknak kell lennie függetlenül attól, hogy agilis vagy nem agilis.
- A 3. szakaszban ismertetett projektmenedzsment környezet is adottságnak tekinthető, ami a kivitelezéstől függetlenül (agilis vagy nem agilis) létezik.
- A 4. szakaszból kiderült, hogy a különféle módszertanokat és szemléletmódokat azonos céllal hozták létre, az eltérés csak a megközelítésben és az eszközökben van.
- Kiderült, hogy a 6. szakaszban ismertetett, módszertanok közötti ellentmondás a projektmenedzsment alatti szinten jelentkezik csak, egyik sem ellentétes, vagy támogató a projektvezetési területtel (6. ábra).

Mindezek miatt a módszertanok gyakorlati alkalmazása nem lehet túlságosan eltérő.

Mivel a módszertanok egymás felé közelítenek, feltehetjük, hogy létezik a kettő között egyfajta optimum: az „ideális” szoftverfejlesztési modell. A „működő szoftver” címszóval jelölt terület az, ahol a projekt csapat magabiztosan, tervezetten, vezetetten fejleszti a szoftvert és a fejlesztés sikeres lesz. Mivel a fejlesztés folyamat-optimalizálás útján jutott ide, párhuzam érezhető a CMMI 5-ös szintjével [7].

A projektmenedzser feladata az, hogy a módszertani elemeket és az eszközöket felhasználva – szükség szerint variálva – megtalálja ezt a középutat. Tulajdonképpen semmilyen újdonság nincs ebben, hiszen a PMI is eleve csak ajánlást fogalmaz meg – építőkockákat (folyamatokat) ad, amelyekből felépíthető a projekt.

6. ábra Egy projekt szintjei



9. Összegzés

Elindultunk a vállalati környezetből, megvizsgáltuk a szoftverfejlesztést a projektmenedzsment szemüvegén át, majd szemügyre vettük fejlesztési módszertanokat, különös tekintettel az agilis szoftverfejlesztésre.

Kiderült, hogy az agilis modell eltér ugyan a hagyományos megközelítéstől, de ez nem gond, hiszen egyik modellt sem alkalmazzuk vakon, és a cél mindenhol azonos lesz. Bármilyen megközelítésből is indulunk, a projektvezetési eszközök alkalmazása és a gyakorlat azonos lesz. Mindegyik modell mellé kell egy projektmenedzser, aki látja az elvárásokat és ezek alapján felépíti a projektjét a módszertani elemek (gyakorlatok és folyamatok) alkalmazásával.

A szerzőkről



CSUTORÁS ZOLTÁN agilis projektvezetési tanácsadó. Több, mint 10 éves tapasztalattal rendelkezik informatikai projektek vezetésében. Hazai környezetben az első között kezdte alkalmazni a Scrum és a Kanban System for Lean Software Development keretrendszereket. Szoftvermérnöki végzettségén túl a Budapesti Műszaki és Gazdaságtudományi Egyetemen humán erőforrás-menedzsmentet tanult, majd MBA diplomát szerzett. Certified Scrum Master és Certified Scrum Professional minősítéssel rendelkezik. A modern szoftverfejlesztési csapatirányítási módszereket mindvégig a tradicionális menedzsment-tudományokra alapozva vizsgálta és alkalmazta. Az általa képviselt adaptív menedzsment eszköztudományokra egyaránt épít a kreatív csapatok vezetésére kialakított agilis módszerekre és a hagyományosan elismert tradicionális eszközökre.



KOCSIS ÁRPÁD informatikai menedzser a Nissannál. 1997-ben végzett a József Attila Tudományegyetem programozó matematikus szakán, azóta rendszergazdaként, programozóként, projektvezetőként és menedzserként dolgozott kisebb és nagyobb (Wincor Nixdorf, Morgan Stanley, TCS, Nissan) cégeknél. Operational Management és Nissan PM minősítéssel rendelkezik. 2007 óta a Nissan európai központjának dolgozik, kezdetben mint a közép-kelet európai régióért felelős informatikai menedzser. Jelenleg az AMIE (Africa-Middle East-India-Europe) régió garanciális informatikai rendszereit felelős. Az agilis alapelvekkel 2001-ben az USA-ban ismerkedett meg, majd hazatérve a Morgan Stanley-nél dolgozott Scrum keretek között. A Nissan-nál, a manufacturing és IT metszéspontjában az asztal másik oldalán ülve lát rá a projektvezetés és a szoftverfejlesztés kérdéskörére.

Irodalom

- [1] Project Management Institute, <http://www.pmi.org/en/Certification/New-PMI-Agile-Certification.aspx>
- [2] PMBOK Guide, 4th edition, pp.18–19., 2008.
- [3] Scrum Guide: <http://www.scrum.org/storage/scrumguides/Scrum%20Guide.pdf>
- [4] “The new product development game”, Harvard Business Review, January-February 1986.
- [5] Agile Manifesto, 2001.
- [6] PMBOK Guide, 4th edition, “Perform Integrated Change Control”, pp.99., 2008.
- [7] “CMMI for Development, Version 1.3” CMMI-DEV (Version 1.3, November 2010). Carnegie Mellon Univ. Software Engineering Inst., 2010.