

SZIKRA: Hibrid agilis módszertan

SZIKRA: Hybrid agile methodology

TÓTH Péter

Eötvös Loránd Tudományegyetem,
1117 Budapest, Pázmány Péter sétány 1/C,
xszerion@inf.elte.hu

Abstract

SZIKRA is a new hybrid software development methodology that provides an opportunity to combine *the benefits of agile methodologies and the waterfall model*. *The methodology is primarily suitable for the development of complete web applications, separated into frontend and backend*. *This article details the roles, events, and products of SZIKRA, and illustrates it through a case study.*

Keywords: hybrid methodology, agile methodology, waterfall model, web development, wireframe

Kivonat

A SZIKRA egy új hibrid szoftverfejlesztési módszertan, mely lehetőséget nyújt arra, hogy egy vízéses modellben gondolkodó cég részére agilis módszertannal tudjunk fejleszteni. A módszertan első sorban teljes webes alkalmazások fejlesztésére alkalmas, frontend és backend bontásban. Ez a cikk részletesen bemutatja a SZIKRA szerepköreit, eseményeit, és termékeit, valamint esettanulmányon keresztül illusztrálja működését.

Kulcsszavak: hibrid módszertan, agilis módszertan, vízéses modell, webfejlesztés, drótváz

1 . Bevezetés

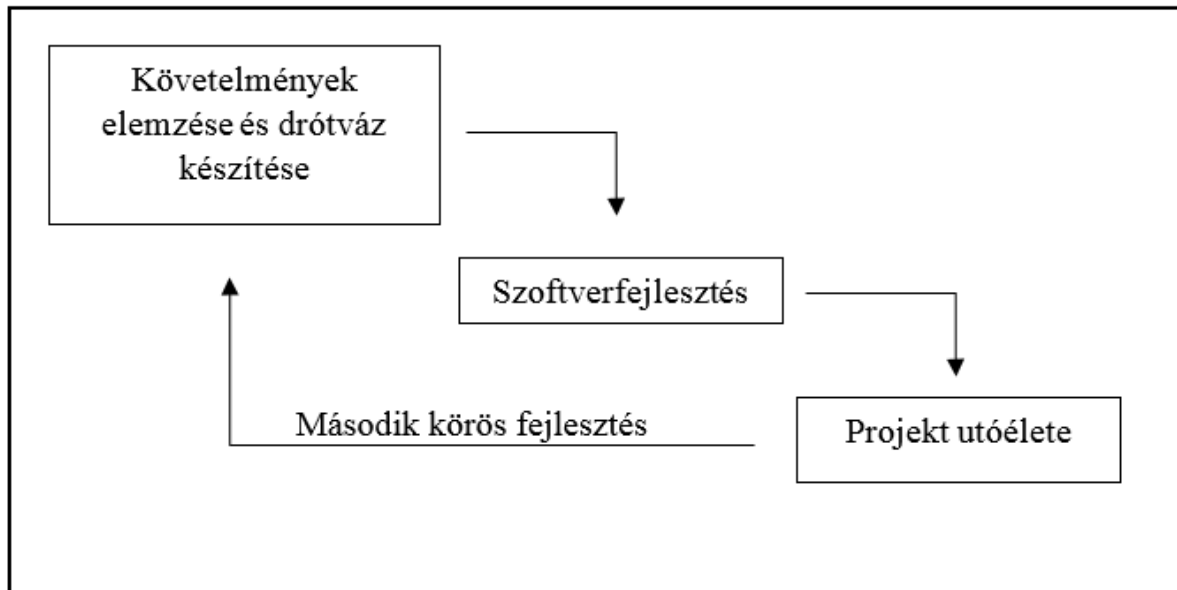
A világon egyre több az informatikai projekt. Specifikus projektszervezési módszertanokra van szükségünk ahhoz, hogy az adott szekcióban az élre törjünk, piacvezetővé válhassunk. Ennek a törekvésnek egyik eredménye a Szervezett, Inkrementális, Klasszikus, Rugalmas, Agilis modell, röviden a SZIKRA. Ez egy teljesen új fejlesztési módszertan mely lehetőséget nyújt arra, hogy egy vízéses modellben gondolkodó cég részére agilis módszertannal tudjunk fejleszteni.

A SZIKRA fő célkitűzése, hogy az agilis és vízéses alapú módszertanok előnyeit megtartsa és a hátrányaiból a lehető legtöbbet kiküszöbölje. A módszertan elsősorban teljes webes alkalmazások fejlesztésére alkalmas, frontend és backend bontásban. Természetesen kínál megoldást arra is, hogy részprogramokat fejlesszünk vele, de elsősorban a teljes alkalmazás fejlesztést támogatja.

A SZIKRÁ-val fejlesztett projekteknél az elsődleges szempont, hogy az ügyfélnek kompromisszumok nélküli webes terméket tudjunk létrehozni, tehát nem szükséges valamely ismert, építőkövekre alapozó webes keretrendszer megkötöttségeihez igazodnia. A rendszer a ma népszerű javascript alapú technológiákhoz hasonlóan a felhasználói felület irányából, az üzleti logika irányába építkezik, hogy az új korszak gondolkodási formájához igazodjon.

1.1 A SZIKRA munkafázisai

A SZIKRA módszertannak 3 nagy munkafázisa van, melyeket 1. ábra foglal össze.



1. ábra
A SZIKRA munkafázisai

Az első fázis a „Követelmény elemzés és drótvázkészítés”. Itt a leendő termékről készítünk egy drótvázat, amit megbeszélésről megbeszélésre finomítunk. A drótvázzal párhuzamosan a backend feladatokhoz is készítünk leírást. Ezzel kapunk egy előzetes képet a projekt nagyságáról és készíteni tudunk egy ütemtervet. Ez a terv a projekt gerincét képezi, viszont rugalmasan alakítható.

Ha megvan az előzetes terv, akkor átléphetünk a második fázisba, a „Szoftver fejlesztés” szakaszba. Ilyenkor a belső projekt tulajdonos egyeztet a fejlesztő csapat vezetőjével és létrehoznak egy tervet, ahol minden szükséges feladatot 1-2 hetes munkaszakaszokba szerveznek. A terv kialakításnál ügyelnek arra, hogy egymásra épülő munkaszakaszokat hozzanak létre, és minden szakasz végén már használható termék jöjjön létre. A kész tervet az ügyféllel egyeztetik és ha megfelel, akkor elkezdik a fejlesztést, ha nem, akkor újratervezik.

A tervezet közös elfogadása után megkezdődik a fejlesztés. A fejlesztő csapat naponta meetingeket tart és együttműködik a megrendelővel. A belső projekt tulajdonos minden munkaszakasz felénél konzultál a megrendelővel, hogy a munkaszakasz milyen végkimenetellel fog zárulni és a sikeresség érdekében itt akár módosíthatnak a feladaton. Ezentúl a megrendelőnek lehetősége van minden nap megtekinteni az előző napi munkafolyamat eredményét és ha nem megfelelő, akkor egyből tudja jelezni a problémákat. Ezzel a folyamattal eljutunk a kész termékig.

Ilyenkor a harmadik fázis, a „Projekt utóélete” következik. Itt átadjuk az ügyfélnek a termék használatához szükséges tudást. Ha már ismeri a terméket az ügyfél, csak az esetleges garanciális hibajavítások, illetve a program üzemeltetése marad hátra.

Habár késznek nyilvánítottuk a terméket, még az ábrán látható a „második körös fejlesztés”. Ez azt takarja, hogy a későbbiekben merülhetnek fel újabb igények a termékkel kapcsolatban. Ilyenkor előről kezdjük a SZIKRA munkafázisait.

A második fejezetben részletesen bemutatom a SZIKRA szerepköreit, majd a harmadik fejezetben ismertetem a SZIKRA eseményeit. A negyedik fejezet a SZIKRA termékeit taglalja. Az ötödik fejezet egy esettanulmányt mutat be. A hatodik fejezet összegzi, hogy a SZIKRA milyen agilis és vízéses modellbeli elemeket tartalmaz és szó esik a továbbfejlesztési lehetőségekről is.

2 Szerepkörök

A SZIKRA csapat a külső-, belső projekt tulajdonosból, a vezető fejlesztőből és a fejlesztő csapatból áll. Az egyes szerepkörök a hatékonyság jegyében jöttek létre. A következőkben részletes bemutatásra kerülnek az egyes szerepkörök felelősségi körei, feladatai.

2.1 Megrendelők

Az ügyfelek, aki megrendelik a fejlesztést. Az első megbeszéléseken ők is részt vesznek és ezeken a találkozókön meghatározzák az elérendő célkitűzéseket. Ennek a csapatnak a tagja a külső projekt tulajdonos.

2.2 Külső projekt tulajdonos

Az külső projekt tulajdonos képviseli a megrendelőt. Feladatát egy személyben látja el, kizárólagos jogkörben jár el a partner nevében. Általában a megrendelő alkalmazottja.

Feladatai:

- Az ügyfél érdekeit képviseli.
- Neki kell tisztába lenni azzal, hogy mi hasznos a cég számára. Ő szabja meg, hogy milyen feladatok elvégzése szükséges a projektben.
- Értékeli a fejlesztő csapat munkáját.
- Ellenőrzi, hogy az elképzelésnek megfelelően halad-e a fejlesztés.

2.3 Belső projekt tulajdonos

A fejlesztői csapaton belül az ő feladata az ügyfél képviselete. Az egész fejlesztési folyamat során aktívan jelen van. Tartja a kapcsolatot a megrendelővel. Fejlesztőként is kiveszi a részét a munkából.

Feladatai:

- Az ügyfél igényeinek felmérése.
- A leendő termék környezeti függésének megértése.
- A termék, a drótváz és napi munka megtervezése.
- Ütemterv elkészítése a csapat és a fejlesztő vezető segítségével.
- A fejlesztők kérdéseinek megválaszolása.
- Figyelemmel kell kísérnie az ütemterv megvalósulását és az eseményeknek megfelelően módosítania kell azt.
- Kapcsolattartó az ügyféllel, leginkább az ügyfél projekt tulajdonosával.

2.4 Fejlesztő vezető

Ő a csapat legtapasztaltabb tagja. Általában ő rendelkezik a legnagyobb fejlesztői rutinnal, valamint jó vezetői képességei is vannak.

Feladatai:

- A sprintek ütemezésének megtervezése a belső projekt tulajdonossal.
- A napi megbeszéléseken kiosztja a feladatokat.
- Biztosítja, hogy a napi megbeszélés a Scrum módszertannak megfelelően valósuljon meg. (jó gyakorlatok, értékek)
- A csapat védelme (pl. túlterheléstől, túlvállalástól).
- Fejlesztésre vonatkozó minden döntés (technológia, módszertan, alvállalkozó, csapat-szervezés).

2.5 Fejlesztői csapat

A fejlesztői csapat általában 3-8 személyből áll. A csapatban a fejlesztő vezető és a belső projekt tulajdonos is fejleszthet. A fejlesztő vezető esetében ez kifejezetten ajánlott.

Feladatai:

- Az egyes feladatok becslése.
- A fejlesztés és a kész termék elkészítése.

Ha a belső projekt tulajdonos fejleszt – lehetséges pozitív hatások:

- Amikor nincsen belső projekt tulajdonosi feladata, akkor tudja gyorsítani a projekt fejlesztését.
- Jobban rálát arra, hogy melyik feladat mikor készül el.
- Rutinban marad a programozói tudása és így a tervezési folyamatai is hatékonyabbak maradnak.
- Jobban tisztába lesz az egyes csapat tagok képességeivel.

Ha a belső projekt tulajdonos fejleszt – lehetséges negatív hatások:

- A feladatokat a saját és nem a csapat tudásához méretezi.
- A külső projekt tulajdonossal való kommunikáció során elsőbbségbe helyezi a saját feladatainak a problémáját.

Ha a fejlesztő vezető fejleszt – lehetséges pozitív hatások:

- Gyorsítja a fejlesztés ütemét, hisz neki van a legnagyobb tapasztalata.
- Pontosán ismeri a projekt állását.
- Magas szaktudásának nincsen holtideje.

Ha a fejlesztő vezető fejleszt – lehetséges negatív hatások:

- A magas szaktudással rendelkező fejlesztő (akár) egyszerű feladatok megoldásával tölti értékes idejét.

3 Események

A SZIKRÁ-ban, a Szoftverfejlesztési fázisban, az események beépítése és egymásra épülése a hatékonyságot szolgálják. Bármelyik eseményt elhagyjuk, azzal csökkentjük a hatékonyságot. Az egyes eseményeket lehet az adott csapat sajátosságaihoz igazítani. A következőben a SZIKRA eseményei kerülnek bemutatásra.

3.1 Sprint

Általában egy-két hét időtartamú munkaszakasz. Az egymást követő sprintek egymásra épülő funkciókat tartalmaznak. Minden sprint végén a végleges program egy újabb, működő de kezdetleges verziója jön létre, míg a legvégső sprintben a teljes program. A sprintekbe ütemezett feladatok nem teljesen fixek, a sprintek során a megrendelő igényeinek megfelelően változhatnak a még hátra levő feladatok – ez ugyan a sprintre betervezett feladatok csúsztatását eredményezheti. Ilyen feltételekkel a már lefejlesztett funkciók is módosíthatók.

3.2 Ütemterv tervezés

Mikor készen van a drótváz és az API specifikáció, akkor a feladatokat elkezdjük sprintekbe szervezni. Ez lehetővé teszi, hogy a teljes projekt anyagi és időbeli vonzatára becslést lehessen adni. Fontos kihangsúlyozni, hogy ez csak egy terv, ami az igények függvényében változhat.

3.3 Napi megbeszélés

Egy megbeszélés, ami minden nap megtörténik. Itt, a Scrum módszertanhoz hasonlóan[3], minden fejlesztő beszámol az előző napi haladásáról és problémáiról. A további feladatokat a fejlesztő vezető osztja ki a csapattagoknak. Ez a megbeszélés maximum 30 percre korlátozott. A teljes csapatot érintő problémákat is itt tudják megbeszélni.

3.4 Napi tájékoztatás

Egy maximum 5 perces megbeszélés, ahol a belső projekt tulajdonos tájékoztatja a külső projekt tulajdonost az előző napi haladásról. A külső projekt tulajdonosnak itt kérdéseket tehet fel, visszajelzéseket adhat. Általában telefonon zajlik.

3.5 Felező megbeszélés (mid sprint)

Ez egy megbeszélés, melyet minden sprint felénél megtartanak. Itt a belső és külső projekt tulajdonos vesz részt, illetve, ha szükséges a fejlesztő vezető. Ezen a megbeszélésen egyeztetik, hogy minden a megfelelő ütemben halad-e vagy újra kell szervezni a sprintet. Az újra-szervezés azzal járhat, hogy feladatokat kell előrébb hozni, vagy feladatokat kell csúsztatni későbbi sprintekbe.

3.6 Alkalmi visszatekintés

Ha a fejlesztés során egy szereplő úgy látja, hogy nem elég hatékony a munka, akkor jelzi az érintetteknek és egy Scrum módszertanhoz hasonló visszatekintést tartanak. Itt elhangzik, hogy mi okoz gondot és a problémagazda felveti, hogy szerinte mi oldaná fel azt. Új ötleteket, módszereket is be lehet hozni és ha a csapat elfogadja és úgy tekinti, hogy ténylegesen hasznos lehet a számukra, akkor beépítik a fejlesztésbe. Ha egy új ötlet nem válik be, akkor azt egy következő alkalommal meg lehet beszélni és törölni, esetleg átalakítani.

3.7 Sprint bemutatása

A belső projekt tulajdonos minden sprint végén bemutatja az elkészült (rész)terméket. Szükség szerint fejlesztők is bevonhatók a bemutatásba. A külső projekt tulajdonos kérdéseket tehet fel, majd eldöntheti, hogy elfogadja vagy megbuktatja a sprintet.

3.8 Termék betanítása

A belső projekt tulajdonos megtanítja a megrendelőt a kész termék használatára.

Lehetséges módszerek:

- Felhasználói dokumentációt ad át a megrendelőnek.
- Megtanítja a külső projekt tulajdonosnak a teljes rendszer használatát, aki majd továbbadhatja a tudást.
- A megrendelő különböző alkalmazottjainak megtanítja a rájuk vonatkozó részeket.

4 Termékek

A SZIKRA módszerrel történő fejlesztés során résztermékek és segédtermékek is létrejönnek. Ezek részben a SZIKRA folyamatok hatékonyságának és pontosságának mérését is szolgálják. A következőben a SZIKRA termékei kerülnek bemutatásra.

4.1 Drótváz

Egy prototípus, mely képekkel és statikusan leprogramozott funkciókkal prezentálja a megrendelő elképzeléseit a jövőbeli termékről. A teljes fejlesztési folyamat erre épül, így ez egy nagyon fontos része a módszertannak. [6]

4.2 API specifikáció

A drótvázhoz társuló API hívások működését és formátumát határozza meg. Leírja, hogy a háttér folyamatok hogyan fussanak le. Az API specifikáció [2] még alkalmas arra is, hogy tisztázzuk, hogyan jutunk el a drótváz A képéről a B képre.

4.3 Ütemterv

Az összes sprintet tartalmazó terv. Ez határozza meg a teljes fejlesztés erőforrás- és feladatmegosztási tervét. Ez csak egy tervezet, a fejlesztési fázisban dinamikusan módosulhat. Az egyes sprinteket illetően a nagyobb feladatok megnevezései szerepelnek, amit a sprint során majd részfeladatokra bontanak.

4.4 Sprint backlog

Az ütemtervben az adott sprintre kiosztott részfeladatok halmaza. [3]

4.5 Kanban board

A teljes termék összes feladatát ábrázoló tábla, amiben látszik, hogy az egyes feladatok milyen készenléti fázisban járnak. Minden feladat leírása mellé címkék társulnak, melyek pontosítják a feladat

jellegét, fontosságát, stb. A feladatok mindig abban az oszlopban állnak, ami az éppen rájuk jellemző munkafázist jelöli (pl. „előkészítve”, „fejlesztés alatt”, „kész”). [4]

4.6 Inkrementum

Az egyes sprintek végterméke. Ez lényegében egy letesztelt, működőképes szoftver, mely a végső terméknek egy kezdetlegesebb verziója, és amely az előző sprint eredményeire épül.

4.7 Verzió

Ha egy projektet túl nagyoknak ítélünk meg és fontos, hogy gyorsan piacra kerüljön, akkor verziókat vezetünk be, amelyek kisebb, önállóan működő változatai a teljes és végleges, nagy terméknek. Ebben az esetben egy verzió lefejlesztése lényegében magába foglalja a teljes folyamatot. Az újabb verzióba a második körös fejlesztéssel lépünk át. Erre csak akkor van szükség, ha nagyon fontos a gyorsaság, hogy minél korábban kerüljön ki a termék éles használatra. A megbeszéléseken ügyeljünk arra, hogy csak a leglényegesebb termék-funkciók kerüljenek bele a korai verziókba.

5 Esettanulmány

A következőkben bemutatásra kerül egy valós szoftverfejlesztési projekt, melynek elkészítése a SZIKRA módszertannal valósult meg. Mivel hosszú lenne a folyamat minden részletét bemutatni, csak a lényeges részek kerülnek hangsúlyozásra. Az esettanulmány neve Fémhal.

5.1 A feladat

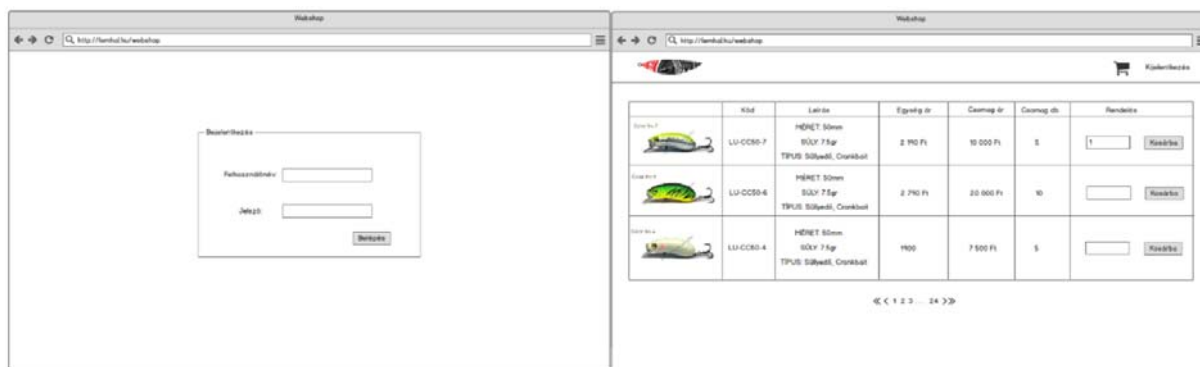
Egy meglévő horgászattal kapcsolatos WordPress weboldalhoz kell egy egyedi plugin-t fejleszteni. A weboldalon horgász felszereléseket szeretnének árulni. A plugin célja egy egyedi webshop létrehozása.

5.2 A csapat és a feladatkörök

A weblap megrendelője egy személy volt (tehát nem több személyes cég), így értelemszerűen ő töltötte be a külső projekt tulajdonos szerepét is. Ezen projekten egy 3 tagú PHP fejlesztő csapat dolgozott, egy junior fejlesztő, egy medior fejlesztő és egy senior fejlesztő. A medior fejlesztő jó kommunikációs készséggel bírt, illetve a tervezésben is volt tapasztalata, így ő töltötte be a belső projekt tulajdonos szerepét. A senior fejlesztőnek volt a legnagyobb szakmai tapasztalata, így ő kapta a vezető fejlesztő szerepet.

5.3 Követelmények elemzése és drótváz készítése fázis

Az első találkozón a belső- és külső projekt tulajdonos vett részt. Tisztázódott, hogy a megrendelő egy horgász nagykereskedő és csak a kiskereskedőknek (horgászboltoknak) árul. A webshop célközönsége tehát a horgászboltok. A webshopot a jelenleg meglévő weboldalhoz kell csatolni. A belső projekt tulajdonos megismerte a jelenlegi vásárlás folyamatát és az elképzelést a webshop-ot illetően. Ezt követően a belső projekt tulajdonos átgondolta a kéréseket és elkészítette az első drótvázat. Ennek egy része a 2. ábrán látható.

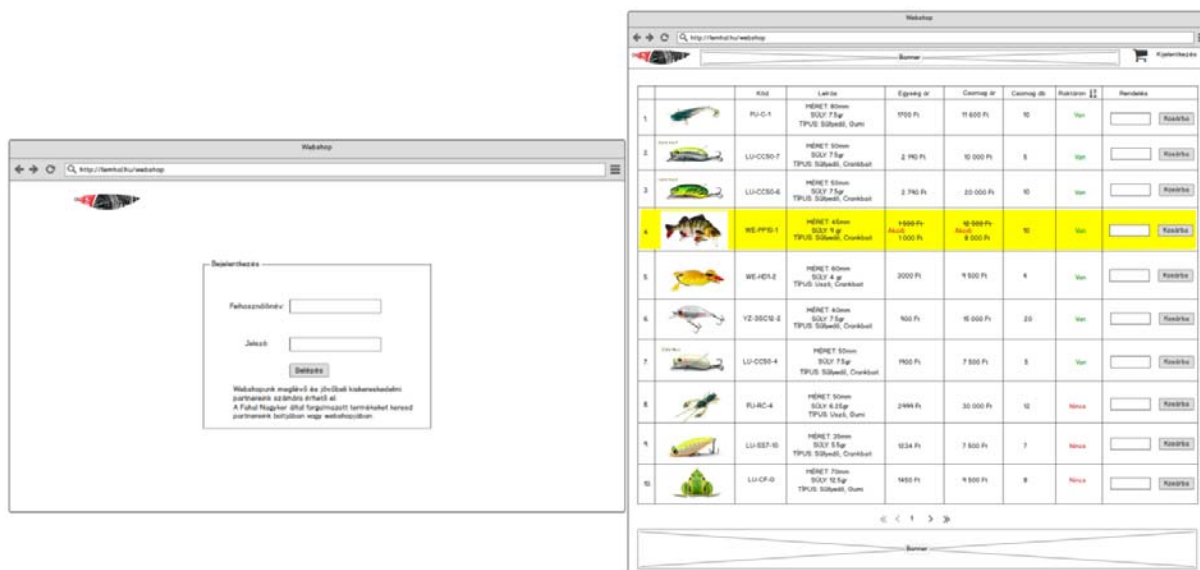


2. ábra

Első megbeszélést követő drótváz - részletek

A projekt 4. napján újra megbeszélést tartott a belső- és külső projektulajdonos. A megbeszélésen bemutatta a belső projektulajdonos az elkészült drótvázat. A külső projektulajdonos átnézte és jelezte, hogy milyen új ötleteket, illetve módosításokat szeretne, pl.: legyen egy felirat a bejelentkezésnél, mely a regisztráció lehetőségét tisztázza; a webshop megrendelésre szolgáló táblázata, bármely fejléc szerinti kattintás esetén rendezhető legyen először abc, fordított abc, majd eredeti sorrendbe; a táblázat sorait lehessen kiemelni beszínezéssel; az akciós termékeknél legyen megjelenítve az eredeti ár és az akciós ár is; a táblázatba legyen egy raktáron „van” vagy „nincs” oszlop is; egy oldalon 10 termék legyen megjelenítve; a lap alján legyen lehetőség előre és hátra lapozásra, illetve az első és utolsó oldalra ugrásra; a fejléccen és a lábléccen legyen banner elhelyezve.

Ezt követően a belső projektulajdonos a kéréseknek megfelelően elkészítette a drótvázon a módosításokat. A módosított drótváz 3. ábrán látható.



Az ütemterv:

1. sprint:

- Fejlesztő környezet kialakítása.
- Rendszer megismerése.
- Bejelentkezés elkészítése.

2. sprint:

- A webshop adatainak az integrációja.
- Webshop admin funkcióinak megvalósítása.

3. sprint:

- A kosár kialakítása.
- Az alap termék-táblázat elkészítése.
- A megrendelő folyamat kialakítása.

4. sprint:

- Az weblap többnyelvűsítése.
- A termék-táblázatban az „akció” funkció lefejlesztése.
- A teljes webshopnak a meglévő weblaphoz igazítása.

A negyedik megbeszélésen a belső projekt tulajdonos ismertette a külső projekt tulajdonossal az ütemtervet, aki elfogadta azt. Ez a munkafázis befejeződött mindössze 8 nap alatt.

5.4 Szoftverfejlesztés fázis

Az 1. sprint zavartalanul zajlott. A napi meetingek, a sprint felező és a sprint bemutató is. A megrendelő kért két kisebb változtatásokat, de azok belefértek a sprintbe.

A 2 sprint már nehézkesebben alakult. Első napján a külső projekt tulajdonos jelezte a belső projekt tulajdonosnak, hogy nem tiszta számára az első sprintben elkészült admin felület működése. A belső projekt tulajdonos egy teljes napot szánt arra, hogy betanítsa a külső projekt tulajdonost az admin felület kezelésére. Emiatt nem tudott haladni saját, betervezett feladataival. A futam 2. napjának reggelén jelezte ezt a fejlesztő csapatnak. A junior fejlesztő is jelezte, hogy neki sem sikerült befejeznie a betervezett feladatát. A 3. nap elején kiderült, hogy a külső projekt tulajdonostól kapott 10 000 termékekről készült képek gond van az elnevezésével és így nem lehet automatizálni a feltöltést. Ezen a napon az egész csapat ezzel az elakadással foglalkozott. Ugyanezen a napon, a felező megbeszélésen a belső projekt tulajdonos megbeszélte a külső projekt tulajdonossal a problémákat és közösen átütemeztek néhány feladatot a következő sprintekbe. Az így módosított sprint sikeresen elkészült.

A 3. sprintben már a felező megbeszélésen látszott, hogy a csapat korábban elkészül a feladataival a sprintbe tervezett feladatokkal. Ezért a belső és külső projekt tulajdonos feladatokat hoztak át a következő sprintből.

A 4. sprint sikeresen zajlott, egy nappal korábban véget is ért a fejlesztés.

Így ez a szakasz 19 napot vett igénybe.

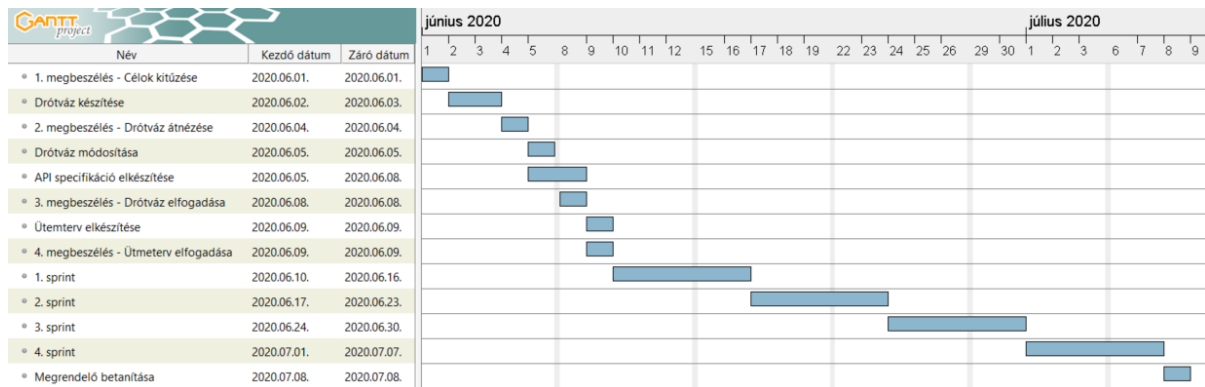
5.5 Projekt utóétele fázis

Megtörtént a teljes termék átadása. A belső projekt tulajdonos megtanította a megrendelőt a webshop használatára egy nap alatt. A belső- és külső projekt tulajdonos megbeszéltek, hogy ha más fejlesztésre is szükség lesz, akkor azt is szívesen elvállalják 2. körös fejlesztés gyanánt.

5.6 Összefoglaló

Ebben az esettanulmányban láthattunk példát a rugalmasságra, mint a drótvázak inkrementális fejlesztése, a felező megbeszéléseken történő átütemezés. Ez alátámasztja, hogy a módszertan agilis elvekre épül. A vízés modell alapú tervezhetőséget is alátámasztja a példa, hiszen 8. nap után tiszta volt, hogy mennyi ideig fog tartani a fejlesztés és ehhez képest egy nappal tartott kevesebb ideig.

A 4. ábra a projekt teljes életútját összegzi.



4. ábra
Projekt életútja

6 Összefoglaló

A SZIKRA egy agilis és vízéses modell által inspirált, hibrid szoftverfejlesztési módszertan. Az agilis kiáltványban[1] megfogalmazott elveket hangsúlyosan támogatja a modell.

Számos esetben előnyben részesíti az egyéneket és a személyes kommunikációt, például: a követelmények elemzése és drótváz készítése személyes megbeszélések során készül és fejlődik; a fejlesztés során napi megbeszélések vannak; felező megbeszélések, alkalmi visszatekintések támogatják a folyamatot; a rendszer folyamatos betanítása is közvetlen megbeszéléseken zajlik.

A SZIKRÁ-ban a működő szoftver elsődleges érték: a sprintek működő szoftvert hoznak létre; a módszertannak az egyik építőeleme a verzió létrehozás, mely rövid időn belül kész és kiadható terméket eredményez; a módszertan drótvázat használ a követelmények rögzítésére - ez a működő szoftverhez közelebb álló követelmény dokumentálási módszer. A SZIKRA nagy hangsúlyt fektet a megrendelővel történő együttműködésre, ugyanis: az elejétől a végéig kíséri a megrendelő a fejlesztés folyamatát; mindennapi munka eredményét követheti a megrendelő - akár óráról órára is láthatja a Kanba tábla alapján; a megrendelő heti bemutatókon vesz részt; a külső projekt tulajdonos által a megrendelőnek nagy hatalma van a módszertan illetően; a megrendelő módosíthatja az igényeit a fejlesztési folyamat során és ez nem jár együtt azzal, hogy az elejéről kell kezdeni a fejlesztést.

A SZIKRA nagy mértékben kész a változásra, rugalmas: azokat a funkciókat, melyekkel a fejlesztők még nem foglalkoztak, bármikor lehet módosítani; a kész funkciókat bizonyos kereteken belül lehet változtatni. A SZIKRA csak akkor nem tudja támogatni a változást, ha a megrendelő ragaszkodik ahhoz, hogy az ütemterv teljesen változatlan maradjon.

A SZIKRA a vízéses modell[5] elemeit is magán hordja. Az 1. tábla összefoglalja a Vízéses modell elemeinek megnyilvánulási formáját a SZIKRA modellben.

A SZIKRA Vízéses modellbeli behatásai

Vízéses modell bahatásai a SZIKRÁra nézve

1.tábla

Vízéses modellbeli elem	Tartalmazás	SZIKRA-beli megfelelő elem
Követelmény specifikáció	Igen	Drótváz, API specifikáció, Ütemterv
Rendszer- és szoftvertervezés	Igen	
Implementáció és tesztelés	Igen	Szoftverfejlesztés fázis
Integráció és rendszerteszt	Nem	- nincsen külön ilyen fázis -
Üzemeltetés és karbantartás	Igen	- ha a megrendelő kívánja -

A további kutatást illetően izgalmasak a SZIKRA következő szempontjait: a SZIKRA skálázhatósága, más módszertanokkal való együttműködése, bevezetési technikája, a megrendelő igényeinek megszorítása a hatékonyabb becslés elérése érdekében, de a megrendelőnek 100%-ban megfelelő termék elkészítését nem sértve.

Összefoglalva: A módszertan támogatja a becslhetőségét a drótváz, API specifikáció, ütemterv által. A rugalmasság azáltal van jelen, hogy a megrendelőnek a külső projekt tulajdonoson keresztül

mindig van beleszólása a projekt alakulásában. A rendszeres és gyakori események jó kommunikációt biztosítanak a projekt érintettjei között és gyorsítják a folyamat biztosan jó irányú haladását. A SZIKRA a piacon lévő más módszertanokhoz képest kissé specifikusabb elemeket is hordoz, de ez a specifikusság teszi lehetővé, hogy jobb megoldásokat nyújtson a webfejlesztés esetében.

7 Köszönetnyilvánítás

A kutatást az „Integrált kutatói utánpótlás-képzési program az informatika és számítástudomány diszciplináris területein” (EFOP-3.6.3-VEKOP-16-2017-00002) című projekt támogatta. A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósult meg.

Köszönöm az ELTE Agilis Kutatócsoportjának, külön kiemelve Ilyés Enikőnek az ELTE Informatikai Karának PhD hallgatójának. Az ő támogatása nélkül nem jöhetett volna létre ez a cikk.

Végül, de nem utolsó sorban köszönöm az Uldin cégnek, hiszen a velük közösen végzett munka során alakult ki a SZIKRA módszertan.

8 Irodalmi hivatkozás

- [1] Kent Beck, James Grenning, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas, Jim magasmester, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, Agile manifesto, 2001, - <https://agilemanifesto.org/>
- [2] Joshua S. Ponelat and Lukas L. Rosenstock, Designing APIs with Swagger and OpenAPI, Manning, 2019, -<https://livebook.manning.com/book/designing-apis-with-swagger-and-openapi/welcome/v-4/>
- [3] Ken Schwaber, Jeff Sutherland, The Scrum Guide, 2017 - <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>
- [4] James Turner, Kanban: The Ultimate Intermediate Guide to Learn Kanban Step by Step, nelly BL International Consulting LTD, 2019.
- [5] Gerardus Blokdyk, Waterfall Model A Complete Guide - 2020 Edition, 5STARCOOKS, 2019
- [6] Terri Jones , Wireframing Design Notebook with 5x5 Graph Paper, Independently published, 2018