

A termék backlog

Az előző alkalom során számos eszközt ismertünk meg amelyek segítségével kialakíthatjuk a termékkel kapcsolatos víziókat. Ezekkel az eszközökkel jól körülhatároltuk azt, hogy kik azok akik majd használni fogják a termékünket és hogy milyen problémát szeretnénk megoldani a leendő felhasználóinknak. Azonosíthattuk azokat a pontokat is ahol a felhasználók majd találkozni fognak a termékünkkel. Ahhoz azonban, hogy ezek a magas szintű tervek és maga a vízió megvalósuljon olyan a gyakorlatban is kivitelezhető elemeket kell létrehoznunk amely alapján a Fejlesztők létre tudják hozni azt a funkció csoportot amely valódi értéket teremt majd a felhasználók számára. Ezek az elemek a Scrum munkaanyagok (artifactok).

» A Scrum munkaanyagai munkát vagy értéket képviselnek. Arra lettek tervezve, hogy a legfontosabb információk átláthatóságát maximalizálják. Így minden elemzést végző fél ugyanazon alap alapján képes alkalmazkodni.

Minden munkaanyag részét képezi egy kötelezettségvállalás is. Ezzel biztosítható, hogy az átláthatóságot és a fókuszt növelő információk álljanak rendelkezésre, valamint ezek alapot szolgáltatnak az előrehaladás méréséhez is:

- A termék Backlog esetében ez a Product Goal. «

Product Goal

» A Product Goal a termék jövőbeni állapotát adja meg. A Scrum Team ennek elérése érdekében állítja össze a terveit. A Product Goal a Product Backlog részét képezi. A Product Backlog többi része folyamatosan alakul ki, ahogy megtaláljuk a választ arra kérdésre, hogy "mit" szükséges kifejleszteni a Product Goal elérése érdekében.

A termék értéknövelő eszköz, egyértelmű keretek között, ismert érdekelt felekkel, jól azonosított végfelhasználókkal vagy ügyfelekkel. A termék lehet szolgáltatás, egy tapintható fizikai termék vagy valami elvontabb.

A Product Goal a Scrum Team hosszú távú célkitűzése. Ezt kell megvalósítaniuk (vagy elhagyniuk) mielőtt egy újabb célkitűzést állítanak maguk elé. «

A Scrum a termék Backlog kötelezettségvállalásaként a Product Goalt határozza meg. Ahogy láthatjuk a Product Goal leírása nagyban hasonlít ahhoz amit az előző alkalom során a termék vízióról megismertünk.

A termék backlog

» A Termék Backlog a termék fejlesztését szolgáló feladatok folyamatosan alakuló, sorrendezett listája. Ez a Scrum Csapat munkájának egyetlen hiteles forrása. «

A termék backlog tehát minden olyan feladatot tartalmaz amelyről a Product Owner úgy döntött, hogy értéket teremt. Amikor egy új elem bekerül a termék backlogba a Product Owner elhelyezi a többi elem sorában ezzel egyértelmű prioritást rendel hozzá. Az elemek sorrendje természetesen változhat az idők során, hiszen ezzel lehet alkalmazkodni a piaci körülmények változásához, új technológiák megjelenéséhez vagy a felhasználói visszajelzésekhez. Mindegyik prioritáshoz azonban csak és kizárólag egy elem tartozhat. A backlogban nem állhat semelyik helyen egyszerre két elem. A Product Owner feladata, kizárólagos joga és felelőssége, hogy a termék backlog sorrendje minden esetben a lehető legnagyobb érték előállítását segítse.

A mindennapi munka során elő fog fordulni, hogy a termék fejlesztésében érdekelt felek a szervezeten belülről egyaránt fontosnak fogják tartani a saját igényeiket. Például a marketing részleg szeretné ha push üzenetet tudna küldeni a felhasználók

számára speciális ajánlatokkal, míg a vezérigazgató az applikációnk új designját szeretné látni. A Scrum elég egyértelműen fogalmaz a Product Owner szerepével kapcsolatban ilyen helyzetben:

» A Product Owner egyetlen személy, nem egy bizottság. A Product Owner sok érdekelt fél igényeit képviselheti a Product Backlogban. A Product Backlog megváltoztatására a Product Owner meggyőzésén keresztül nyílik lehetőség. «

Egy ilyen esetben tehát nem az igény megfogalmazójának szervezeti hierarchiában betöltött szerepe vagy informális hatalma dönt arról, hogy a push üzenetek vagy az új design kerül a Fejlesztők asztalára először. A Product Owner az aki kellő felhatalmazással és termék ismerettel rendelkezik ahhoz, hogy meghatározza ezen elemek sorrendjét a backlogban, majd a döntését átláthatóvá tegye a teljes szervezet számára. Ha úgy dönt, hogy a push üzenetekkel nagyobb bevételt tud termelni a termék és az új ajánlatok növelik a felhasználók elégedettségét akkor ezt fogja magasabbra rangsorolni. Ha az új design kényelmesebbé teszi az oldal használatát és még több felhasználót ösztönöz az applikáció használatára akkor pedig ezt fogja priorizálni.

A termék backlog másik sajátossága, hogy az egyes elemei eltérő méretűek - azaz eltérő mértékű idő és tudás ráfordítással lehet őket megvalósítani - és eltérő részletességű leírást tartalmaznak arról, hogy mi is a mögöttük rejlő pontos igény.

Amikor a termék víziót elkezdjük lebontani kivitelezhető egységekre akkor először mindig a nagyobb egységek születnek meg először. Vegyünk most példának egy szakmai hírekkel foglalkozó portált. Ha egy ilyen terméket szeretnénk megvalósítani akkor szükségünk lesz arra, hogy az olvasók el tudják olvasni azokat a híreket amelyeket megjelenítünk. Kell egy olyan felület ahol a cikkek szerzői létre tudják hozni az olvasók számára a tartalmat. Ha a portál bevételét nagyrészt reklámok adják akkor a reklámkampányok anyagait is kezelni kell.

Máris van három, igencsak nagy méretű, de már kézzelfogható backlog elemünk amely mentén elkezdődhet a fejlesztés. Persze ha ezekkel az igényekkel állunk a Fejlesztők elé, akkor jó esetben rengeteg kérdést, rosszabb esetben keresetlen szavakat kapunk tőlük. A fenti sorrendet követve az olvasói felület kapcsán el kell döntenünk olyan kérdéseket, hogy "Hány hír jelenjen meg egyszerre az oldalon?", "Legyenek-e kiemelt hírek?", "Akarunk-e külön rovatokat és eszerinti csoportosítást?" és így tovább. Ezen kérdések nem csak az olvasói, de a szerkesztői felület vagy a kampány menedzsment felület kapcsán is fel fognak merülni.

Ha a Product Owner jól végzi a munkáját akkor már a fejlesztési folyamat elején és onnan folyamatosan bevonja a Fejlesztőket a backlog finomításába, kisebb egységekre történő bontásába. Ez azért is hasznos mert így a Fejlesztők is beleszólást kapnak a termék kialakításába amely számukra is motivációt ad, valamint számos olyan kérdést feltehetnek amely új backlog elemek létrehozására ösztönzi majd a Product Ownert.

Miután az egyes backlog elemeket kisebb részekre osztottuk érdemes újra megvizsgálni a kisebb, jobban definiált elemek sorrendjét. Egy kezdetleges hírportálnál például elég ha formázás nélkül megjelennek a cikkek, de már meg tudunk jeleníteni egyszerű kép formátumú hirdetéseket az egyes cikkek alatt.

User story

A termék backlog létrehozásakor már tárgyaltuk azt, hogy a backlogban szereplő elemek eltérő méretűek és részletességűek lehetnek. A tényleges megvalósíthatósághoz időben legközelebb eső elemek ideális esetben a legrészletesebbek és legkisebbek, míg a távolabbiak kevesebb részletet tartalmaznak és funkciók nagyobb körét ölelik fel. A Scrum csupán azt határozza

meg, hogy a termék backlog elemekből áll és ezeket a Scrum csapat folyamatosan finomítja:

» Azon termék Backlog elemek, melyek egy Sprinten belül elvégezhetőek a Scrum Team által, készen álltnak tekinthetőek a Sprint Planningen történő kiválasztásra. Ezt az átláthatóságot általában finomítási tevékenységek során érik el. A termék Backlog finomítása a termék Backlog elemeinek lebontása és tovább definiálása kisebb, pontosabb tételekre. Ez egy folyamatos tevékenység, amely részletekkel, például leírással, sorrendezéssel és méretek megadásával bővíti az elemeket. Ezek az attribútumok a munka területétől függően változhatnak. «

A legegyszerűbb ilyen termék Backlog elemek leírásának egyik módszere a User Storyk létrehozása. A User Story alkotás nem része a Scrum keretrendszernek, de egy jó kiegészítője lehet annak.

A User Story nem más mint egy a termékkel kapcsolatos igény megfogalmazása a felhasználó szemszögéből. A legnépszerűbb és legegyszerűbb formátum alapján a User Story felépítése a következő:

{Ügyfél}ként

Azt akarom, hogy {a várt termékfunkció leírása}

Azért, hogy {az érték amit adni fog nekem ez a funkció}

Felmerülhet a kérdés, hogy miért van szükség arra, hogy ilyen körülményes formában írjunk le minden új funkciót? Előfordulhat, hogy valami nagyon triviális dolgot akarunk megvalósítani, akkor rengeteg időt vesz igénybe csak a leírás létrehozása. Ha csak annyit szeretnék a termékben látni, hogy a felhasználó be tudjon jelentkezni egy adott felületen az mindenki számára egyértelmű, nem?

Két okból is érdemes azonban ragaszkodni a fenti formátumhoz.

Egyrésztől kapcsolatot hozunk létre a funkcióhoz kapcsolódó részletek között. Meghatározzuk ki a felhasználó aki számára értéket termet majd a funkció, mi lesz konkrétan az ami megteremti majd ezt az értéket és leírjuk azt is miért szeretnénk ezt az értéket megteremteni. A felhasználó jelenléte a User Storyban segíti a Fejlesztőket abban, hogy azonosulni tudjanak a megoldandó problémával, a miért pedig megadja a lehetőséget arra, hogy kellő szabadságot kapjanak a funkció megvalósításában.

Másrésztől a "lehessen bejelentkezni" jellegű termék Backlog elemek nem nyújtanak elég információt egy új csapattag részére, sőt ha valamilyen okból vissza kell tekinteni egy korábbi User Storyra akkor szintén kevésbé lesz érthető, hogy miért hoztunk létre egy adott funkciót.

A fenti formátumot a három rész megtartásával szabadon bővíthetjük, hogy még pontosabb képet tudjunk adni a Fejlesztőknek arról mi is az amivel bővíteni szeretnénk a terméket.

Példaként vegyünk most egy csomagszállítással foglalkozó vállalkozást és annak a feladó számára létrehozott felületét. Szeretnénk ha a feladók le tudnák kérdezni a feladott csomagjait állapotát. Egy ilyen esetben a User Story a már ismert formátumban így nézhet ki:

Feladóként

Azt akarom, hogy lássam az elküldött csomagjaim állapotát

Azért hogy, tájékoztathassam a címzettet a várható szállítási időről

Ezt a user storyt használva több megoldás is szóba jöhet. A feladónk számára az jelenti az értéket, hogy a címzett és ő is értesül a szállítás várható időpontjáról.

Csinálhatunk egy egyszerű listát az összes feladott csomagról ahol a feladó megkeresheti az aktuáliast és saját maga értesítheti a címzettet a várható időről.

Pontosítsuk most egy új hozzáadott elemmel a User Storyt.

Feladóként aki most adott fel egy új csomagot

Azt akarom, hogy lássam az elküldött csomagjaim állapotát

Azért hogy, tájékoztathassam a címzettet a várható szállítási időről

Ebben a megfogalmazásban szűkítettük a megoldási lehetőségeket, hiszen egy adott csomag - az épp most feladott - szállítási időpontját szeretnénk látni. Ez történhet a feladás sikerességét visszaigazoló képernyőn vagy akár egyszerűen úgy, hogy időrendi sorrendben jelenítjük meg a listát.

Ahogy a fenti példából is látjuk az, hogy az érték hogyan fog előállni, nincs meghatározva az User Storyban. Ez lehetőséget ad arra, hogy a Fejlesztők még a megvalósítás előtt több megoldási javaslatot tegyenek és a felhasználók a lehető legjobbat kapják majd kézhez ezek közül. Az hogy ténylegesen mennyi értéket adtunk persze továbbra is a felhasználók visszajelzéseiből fog látszani.

Attól függően, hogy milyen komplexitású megoldásra váró problémát fogalmazunk meg egy User Storyban elképzelhető, hogy nem lesz megvalósítható egy Sprintem belül. Némi irodalmi párhuzamot behozva egy-egy User Storyt úgy is elképzelhetünk mint egy bekezdést vagy egy fejezetet egy nagyobb műből. Ha ennek a műnek az egyik kötetét vesszük akkor az már egy hősköltemény (Epic). Ha pedig kötetek során rágjuk át magunkat, mondjuk a viking mitológiát végigjárva akkor pedig egy egész Sagaról beszélhetünk. Mindegyik egység önmagában is értelmezhető, közösen pedig egy nagy egységet alkotnak.

Ezt a párhuzamot a backlog elemek létrehozásakor is szokták használni általában úgy, hogy egy nagyobb egységet bont le a Product Owner a Fejlesztők bevonásával kisebb, alacsonyabb komplexitású és lehetőség szerint egy Sprinten belül megvalósítható elemekre. A nagy átfogó elemek - például a felhasználói adatkezelés - lesz a Saga. Ennek egyik fontos eleme a bejelentkezés már egy kisebb Epic szintű elem. Ha pedig ezt még tovább bontjuk akkor a felhasználói fiók megjegyzése egy adott eszközön már lehet egy User Story.

Bárhogyan is nevezzük azonban a backlog elemeit azok alapvetően méretükben és komplexitásukban különböznek csak egymástól azonban közös tulajdonságuk, hogy a felhasználó szemszögéből íródnak, megfogalmazzák, hogy mivel és hogyan teremtenek értéket. Akár a User Story akár más formátumot használunk a backlog elemek leírására érdemes egy adott termékhez kapcsolódóan egy fajta módszert használni.

INVEST

A fentiekben megismerhettük a User Storyk formátumát és alapvető tulajdonságait. Egészítsük ezt most ki egy olyan módszerrel amely alkalmazásával még jobb User Storykat hozhatunk létre. Az INVEST egy betűszó és minden egyes eleme a kiváló User Story egy-egy tulajdonságát mutatja be. Nézzük meg ezeket egyesével példákon keresztül.

Független (Independent)

A User Story függetlensége azt jelenti, hogy nincs átfedésben másik User Storyval. Függőségek természetesen lehetnek egyes User Storyk között például a regisztráció nélkül értelmetlen a felhasználó profil oldalt létrehozni. Az egymással átfedésben lévő storyra akkor találunk példát, ha nem keresztfunkcionális csapat dolgozik a megvalósításon. Maradva a regisztrációs folyamatnál szükséges ehhez nagyvonalakban szükségünk lesz egy felületre ahol a felhasználók megadják az

adataikat és egy adatbázisra ahol ezeket tároljuk. Ha ezen két elemet külön-külön backlog elemként hozzuk létre akkor azok nem csak kölcsönösen függnek majd egymástól, de nagy valószínűséggel a Sprint végén önállóan kevés vagy semennyi értéket nem fognak képviselni. Az ilyen horizontális típusú backlog finomítást, amikor az egyes backlog elemeket nem önállóan is használható egységekre hanem a technológia mentén bontjuk kisebb részekre érdemes a gyakorlatban elkerülni, mert lassítja a piacra lépést és feleslegesen pazarolja a szűkös erőforrásokat.

Tárgyalható (**N**egotiable)

Amíg a Product Owner azt mondja meg, hogy mi az amit megvalósítsunk addig a Fejlesztők azt, hogy hogyan. Egy User Story akkor működik jól, ha teret ad a kreativitásnak és arra ösztönzi a Fejlesztőket, hogy több választ is adjanak a felhasználók problémáira. Vegyünk először egy olyan példát ami több elemében ellentmond ennek így kerülendő:

Product Ownerként

Azt akarom, hogy az új felhasználók végignézzenek egy videót a regisztráció után

Azért hogy, a marketing részleg videóba fektetett munkája értelmet nyerjen

A fenti - szinte minden elemében rossz - User Story kiválóan példázza azt, hogy a formátum megtartásával, de a szellemiség semmibe vételével milyen zombit lehet alkotni. Nézzük sorban a hibákat. Nem a felhasználó szemszögéből van megfogalmazva, hanem egy utasítást fogalmaz meg, pedig a Product Ownernek nincs ilyen felhatalmazása. Pontosán leírja hogyan kellene megoldani a problémát: vetítsük le a felhasználónak a videót - nincs lehetőség más megoldásra. A miéltre adott válasz pedig szinte biztos, hogy a Product Ownerre gyakorolt nyomás hatására jön létre.

Nézzük meg, hogyan lehetne ezt jól átírni.

Új felhasználóként aki most fejezte be a regisztrációt

Azt akarom, hogy részletes információt kapjak a termék működéséről

Azért, hogy könnyen megtehessem az első lépéseket

Ez a User Story a felhasználó szempontjából mutatja be a problémát. Az, hogy nem írja le a megoldást, csak azt hogy a felhasználónak több információra van szüksége nyitva hagyja a lehetőséget arra, hogy a Product Owner és a Fejlesztők közösen találjanak megoldást, ami lehet egy videó is, de egy szöveges dokumentum vagy képsorozat esetleg egy tutorial is. Az első lépések mint a probléma fókuszpedig a segítség időbeli pontját és komplexitását is körbeírja, de nem határolja le teljesen ezért újabb beszélgetést fog igényelni.

Értékteremtő (Valuable)

A termék Backlog feladata, hogy garantálja azt, hogy a felhasználók mindegyike Sprint után a lehető legnagyobb értéket kapják meg. A backlog elemeit ezért mindig úgy kell létrehozni, hogy az elemek által teremtett érték egyértelmű legyen. A User Story formátuma lehetővé teszi azt, hogy az "Azért, hogy..." részben teret adjunk annak, hogy miért is fog értéket teremteni a storyban megfogalmazott funkció. Gyorsabban tudja használni az applikációkat a felhasználó? Kényelmesebben? Egy régen fennálló problémájára adunk választ? Bármelyik fenti kérdésre is felelünk igennel az értéket fog teremteni a felhasználónak. Az érték megfogalmazása a user storyban pedig segít a Fejlesztőknek, hogy a munkájuk során lássák miért dolgoznak és közel hozzuk hozzájuk a felhasználót.

Becsülhető (Estimable)

A User Storyk azért születnek, hogy alapul szolgáljanak a későbbi fejlesztésnek. A Fejlesztők a munka elkezdése előtt megpróbálják hozzávetőlegesen megbecsülni, hogy egy adott Sprint alatt mennyi és milyen komplexitású feladatot tudnak elvégezni, hogy előálljon az egyes User Storykban megfogalmazott érték. A becsléshez arra van szükség, hogy az egyes storyk jól körülhatárolható igényeket fogalmazzanak meg. A becsülhetőséghez a Fejlesztők számtalan kérdést és javaslatot fogalmaznak meg a Product Owner felé és ebből születik meg a pontos igény majd az ehhez kapcsolódó becslés.

Ennek részleteiről és a becslési technikákról a következő alkalommal lesz szó.

Kicsi (Small)

Az, hogy egy User Story kicsi nem arra utal, hogy milyen hosszú vagy részletes leírást tartalmaz a szövege. Ideális esetben a User Story nem nagyobb annál, hogy egy teljes Sprintet igénybe vegyen a megvalósítása. Már akkor is érdemes elgondolkodni azon, hogy tudjuk-e kisebb részekre bontani a storyt ha önmagában egy teljes Sprint szükséges ahhoz, hogy létrehozzuk a benne megfogalmazott értéket, de ha ennél is több idő szükséges biztos, hogy további finomításra van szükség.

Fontos megjegyezni, hogy egy User Story méretét nem a Product Owner, főleg nem a szervezeten belüli más stakeholder mondja meg, csak és kizárólag a Fejlesztők. Az ezzel ellentétes működés egy olyan negatív spirált tud kialakítani ahol becslések helyett elvárások fogalmazódnak meg a csapat felé, akiknek csökken a motivációja és még kevesebb értéket tudnak előállítani aminek hatására a stakeholderek még komolyabb elvárásokat fogalmaznak meg és így tovább. Ez persze nem jelenti azt, hogy nem elvárható a fejlődés a Fejlesztők részéről, de egy magasan motivált, kellő felhatalmazással rendelkező és egy komolyabb termék ismerettel rendelkező csapat ezt automatikusan szállítani fogja.

Tesztelhető (Testable)

Nem elég ha értéket állítunk elő az User Story megvalósításával, de úgy kell ezt tennünk, hogy az megfeleljen bizonyos minőségi kritériumoknak is. Ezeket a minőségi kritériumokat a Fejlesztők állítják fel a saját munkájukkal szemben. A tesztek elvégzése szintén az ő feladatuk és ez a tevékenység része a Sprintben folyó munkának. A Product Owner feladata ebben az, hogy a Fejlesztőkkel közösen megfogalmazza azokat az üzleti szempontból releváns elvárásokat amelyek mentén elfogadja a Scrum Csapat, hogy egy User Story értéket teremtett.

Vegyünk példának egy repülőjegy vásárlással kapcsolatos termék backlogjából egy User Storyt.

Utazóként aki kiválasztotta a jegyeit

Azt akarom, hogy láthassam a kiválasztott jegyeimet a fizetés előtt

Azért, hogy módosíthassak rajtuk szükség esetén.

Ehhez a User Storyhoz a lentihez hasonló acceptance criteria tartozhat:

- A jegyekre kattintva a kiválasztott jegyek egy új ablakban nyílnak meg
- Az új ablakban látszik a kiválasztott ülőhely
- A felhasználó tudja módosítani az ülőhelyet
- A jegyek összértéke látszik a felhasználó számára
- Az utazás időpontja látszik a felhasználó számára
- A fizetést választva a felhasználó kiválaszthatja az elérhető fizetési lehetőségeket
- A fizetési opciót kiválasztva átirányítjuk a felhasználót a banki felületre

A fenti lista nem teljes, de ennyi részlet mentén a Fejlesztők jól körül tudják határolni azokat a feladatokat amelyeket el kell végezniük, hogy a leírás alapján

működő funkció jöjjön létre és vissza is tudják ellenőrizni, hogy a munkájuk megfelel-e az összes üzleti kritériumnak.

Definition of Ready

Megismerhettük, hogy az egyes backlog elemeket hogyan érdemes felépíteni. Az Agilis Kiáltvány alapján a személyeket és a köztük létrejött interakciót nagyobbra értékeljük a módszertanokkal és az eszközökkel szemben. Ennek szellemében előfordulhat, hogy a Fejlesztők megfogalmazzanak olyan elvárásokat az egyes backlog elemekkel szemben, amely megkönnyítheti a közös munkát. Az átláthatóság jegyében ezt érdemes mindenki számára elérhető módon dokumentálni. Ezt a dokumentumot fogjuk Definition of Readynek nevezni.

Egy ilyen dokumentum tartalma nem előre rögzített, hiszen egy párbeszédből fog kialakulni, de néhány elemet érdemes megfontolni a megalkotása során:

- Fejlesztésre kész egy backlog elem, ha minden Fejlesztő kellően jól érti milyen értéket hoz létre és ez a megértés közös a Product Ownerrel
- Tartalmazza a technikai specifikáció megalkotásához szükséges részleteket (például egy felület drótvázát)
- Megfelel a korábban bemutatott INVEST tulajdonságoknak
- Nincs átfedésben más backlog elemmel

Ha közös megállapodás alapján születik meg a Definition of Ready akkor a Fejlesztők mondhatják azt, hogy addig nem tudnak egy User Story megvalósításán dolgozni, amíg az nem felel meg a Definition of Readynek. Fontos azonban az egyensúly: ez nem egy fegyver, hanem a közös megértést segítő eszköz. Akár a Product Owner hanyagolja el a termék Backlog elemeinek kezelését, akár a Fejlesztők nem mozdítják a kisujjukat sem amíg pontos specifikációt nem kapnak azt jelzi, hogy a Scrum csapat letért az útról. Egy jó Scrum Master felismeri az ilyen helyzeteket és segít elmozdítani ezeket az akadályokat.