

4IT421 ZLEPŠOVÁNÍ PROCESŮ BUDOVÁNÍ IS – SEMESTRÁLNÍ PRÁCE

AUTOR

Bc. Martin Uřídil (xurim00@vse.cz)

TÉMA PRÁCE

Porovnání Scrum a Kanban

TYP TÉMATU

Individuální téma



Obsah

1	Úvod.....	2
1.1	Scrum v kostce.....	2
1.2	Kanban v kostce.....	3
2	Normativní vs. adaptivní.....	4
2.1	Priority v product backlog	5
2.2	Každodenní porady (Daily standup meetings)	6
2.3	Vizualizace postupu – grafy	6
3	Scrum/Kanban tabule.....	8
3.1	Vlastnictví tabule.....	9
4	Rozdělení do týmů	9
5	Omezení množství paralelních úkolů (WIP).....	11
6	Volba metodiky.....	11
7	Souhrnné srovnání	12
8	Závěrem	13

1 ÚVOD

Cílem této práce je přiblížit čtenáři dvě agilní metodiky Scrum a Kanban, které se obě zaměřují na iterativní vývoj software. Nejprve ve stručnosti uvedu, čím se každá metodika vyznačuje a jaké prvky používá a hned poté přistoupím k jejich porovnání dle různých aspektů. Cílem porovnání těchto metodik nebude stanovit, která z nich je lepší, nebo která se více hodí pro daný typ vývoje. Cílem bude spíše ukázat, jak každá metodika přistupuje k dílčím procesům řízení vývoje software a dokázat, že největších přínosů dosáhne vývojový tým, pokud si z každé metodiky vybere pouze určitou část a zbytek si upraví přesně na míru svému prostředí, k čemuž agilní přístupy přímo vybízejí.

Zároveň budu metodiky hodnotit podle toho, jak naplňují principy lean¹ přístupu k řízení.

1.1 SCRUM V KOSTCE

Autory této metodiky jsou Ken Schwaber, Jeff Sutherland a Mike Beedle. Přístup Scrumu k řízení projektů je velice jednoduchý, což zároveň podtrhuje jeho agilnost. Tato metodika velice dobře splňuje lean principy i když můžeme narazit na oblasti, které jsou s nimi v rozporu (blíže se k této problematice vrátím později). Celý proces řízení spočívá v rozdělení prací na jednotlivé iterace (sprinty) se stejnou délkou trvání. V rámci každé iterace jsou pak dílčími týmy vypracovávány úkoly, které vychází ze seznamu požadavků na produkt (dále *product backlog*). Nejprve nastavíme základní parametry řízení projektu. Těmi jsou:

- Nastavení rolí – Ve Scrumu se vyskytují pouze tři základní role. Jsou jimi **Product Owner** – spravuje seznam požadavků na tvorbu produktu a kontroluje směřování projektu. Většinou bývá kontrolním orgánem, který zajišťuje, aby byly požadavky klienta správně implementovány. **ScrumMaster** – zodpovídá za celý průběh procesu řízení projektu. **Team** – Autonomní skupina lidí, která v rámci každého sprintu pracuje na dodání určité části systému. Teamů může být více.
- Délka jednotlivých iterací – „Sprint“ – Scrum sice nepředepisuje, jak dlouhá by tato iterace měla být, ale říká, že by měla mít pevnou délku trvání, která se nemění. Často bývá zvykem praktikovat 30ti denní iterace. Na konci každého sprintu by měl každý tým dodat ucelenou práci, kterou v průběhu sprintu vytvořil.
- Vytvoření úvodní verze „Product backlogu“ – jedná se o prioritizovaný seznam požadavků na finální produkt, na základě kterého se pak vytváří plány úkolů (sprint backlog) pro jednotlivé sprinty. Tento seznam spravuje Product owner a je možné ho mezi každým sprintem měnit.

¹ Vychází ze slova „štlhlý“. Lean manufacturing byla metodika vyvinutá firmou Toyota a jejím hlavním cílem je minimalizace „plýtvání“, tím že dodává opravdu jen to, co zákazník žádá.

Když máme projekt takto připraven, můžeme zahájit první sprint. Struktura samotného sprintu pak chronologicky probíhá následovně:

1. Plánovací schůze sprintu – každý tým si zde vybere určité požadavky z product backlogu a rozloží si je do samostatných úkolů, které bude v rámci sprintu realizovat. Pro každý tým tak vznikne tzv. sprint backlog, který obsahuje úkoly realizované v rámci následujícího sprintu.
2. ScrumMaster zahájí sprint s předem stanovenou délkou trvání.
3. Každý den probíhají tzv. „Daily standup meeting“, tedy krátké schůze na začátku každého dne, kde každý člen shrne svou předchozí práci a zmíní problémy, se kterými se potýkal.
4. Po uběhnutí doby sprintu se koná tzv. „Sprint review“, tedy schůze, jejímž cílem je předvést práci udělanou za poslední sprint, zjistit slabiny v procesu řízení a navrhnout zlepšení.
5. Následuje další sprint od bodu 1.

KNIBERG (2009) pak shrnuje celý proces práce dle metodiky Scrum takto: „Na místo velké skupiny lidí, která by trávila mnoho času děláním rozsáhlé věci, rozdělíme organizaci na malé týmy, které tráví méně času na malých úkolech.“ Pravidelnou integrací pak Scrum staví celkový produkt.

1.2 KANBAN V KOSTCE

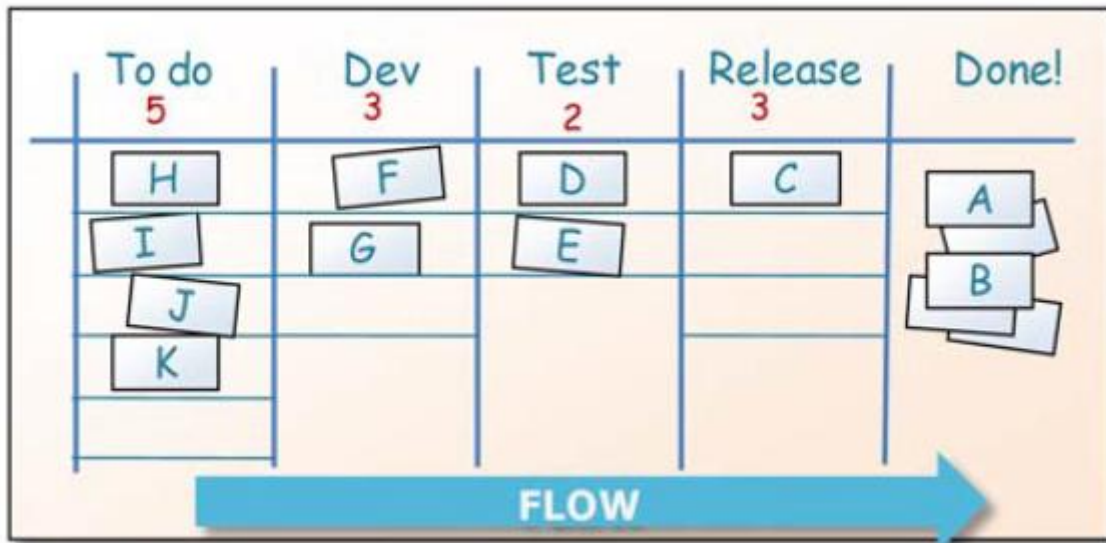
Pokud jsem v předchozí kapitole nazval přístup Scrumu k řízení projektů jako jednoduchý, tak přístup metody Kanban je ještě jednodušší. Metodika Kanban může být rovněž zařazena do množiny lean přístupů k řízení projektů, praktikujících iterativní vývoj software. Tato metodika, jejíž název vychází z japonského slova, které v překladu znamená cedule nebo billboard, byla vytvořena v rámci zavádění lean přístupů v továrnách Toyota po 2. světové válce (viz GROSS, 2003). Jejím hlavním cílem je nabídnout nástroj, který pomůže určit, co je potřeba vyrábět, kdy vyrábět a v jakém objemu. Tyto principy se dají velmi dobře aplikovat také při vývoji software, čehož využili autoři David Anderson a Don Reinertsen a vytvořili adaptaci původní metodiky Kanban s novým názvem „Kanban for software development“, viz CRISP AB (2012).

Metodika Kanban pro vývoj software (dále jen Kanban) se opírá o tři základní pilíře:

1. **Zobrazení toku práce** – Kanban doporučuje stejně jako Scrum rozdělit práci na dílčí položky. Tyto položky pak sepsat na samostatné karty (fyzické nebo virtuální v softwarovém nástroji), které by měly být viditelné pro celý tým. Tyto karty by měly být umístěné do jednoduchého schématu (Kanban board), vytvořeného ze sloupců, kde každý sloupec představuje jednu fázi v celém cyklu vývoje. Viz Obrázek 1-1.

2. **Snaha o omezení množství souběžně rozpracovaných úkolů** – Kanban říká, že je nutné stanovit přesný limit množství karet v daném sloupci. Tento limit označuje pomocí čísla umístěného v horní části sloupce.
3. **Měření času jednoho cyklu** – díky měření doby trvání, než karta projde celým cyklem, můžeme stanovit, kolik času týmu zabere, než implementuje jeden úkol.

Čím lépe tým rozděluje úkoly na stejný rozsah náročnosti, tím přesnější odhady délky trvání cyklu pak může stanovovat.



OBRÁZEK 1-1 SCHÉMA ROZDĚLENÍ FÁZÍ V METODICE KANBAN

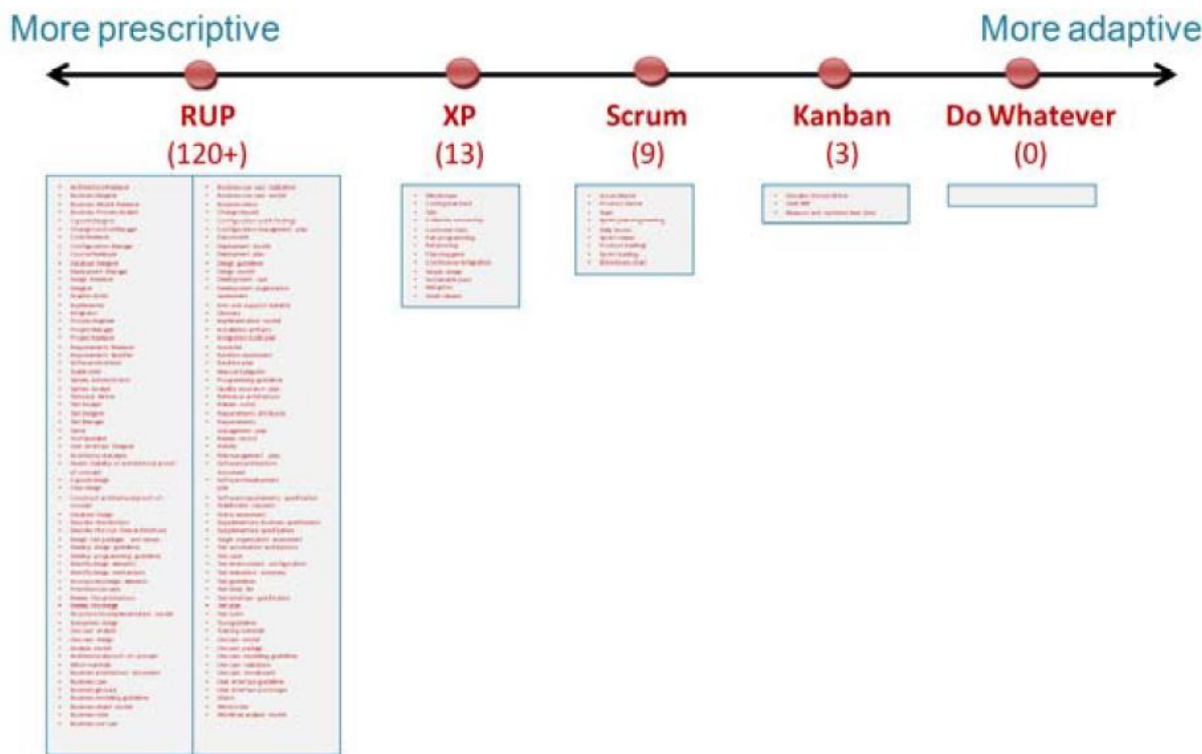
Z tohoto schématu tedy vyplývá, že na rozdíl od metodiky Scrum, Kanban vůbec neříká, jak dlouhá by jedna iterace měla být, omezuje pouze počet rozpracovaných úkolů v jedné fázi. Proto zde ani neexistuje přesnější popis struktury jedné iterace (tak, jak Scrum popisuje průběh sprintu) a Kanban toto ponechává na rozhodnutí týmu. Tento aspekt je velice důležitý při porovnání metodik. Zatímco metodika Scrum praktikuje v rámci sprintu tzv. „push“ systém, přístup Kanbanu můžeme přirovnat k flexibilnějšímu „pull“ systému.

„Push“ systém se soustředí na dosažení předem stanoveného limitu, nehledě na aktuální potřeby zákazníka nebo tržní situaci. Naopak „pull“ systém je plně flexibilní a okamžitě se přizpůsobuje potřebám odběratele, viz DUBAKOV (2009). „Pull“ systémy, včetně Kanbanu, tak mnohem lépe zapadají do celého lean konceptu.

2 NORMATIVNÍ VS. ADAPTIVNÍ

V předchozí kapitole jsem stručně popsal fungování obou metodik, nyní se již budu plně soustředit na vzájemné porovnání jejich principů.

Jako první se podívám na to, do jaké míry je každá metodika svazující a kolik pravidel s sebou přináší. Už z předchozího popisu metodik je zřejmé, že Scrum nastavuje mnohem více pravidel, než Kanban. Následující Obrázek 2-1, pak na ose seřazuje agilní metodiky dle množství jimi stanovených pravidel. Pořadí vychází od metodik s mnoha pravidly až po ty nejvíce volné/flexibilní.



OBRÁZEK 2-1 MNOŽSTVÍ PRAVIDEL U AGILNÍCH METODIK

Je vidět, že nejvíce pravidel v kategorii agilních metodik definuje RUP² (přes 120 pravidel), na druhé straně stojí „ad hoc“ přístupy (tedy žádná metodika) a metodika Kanban, která má pouze 3 již popsaná pravidla. Metodika Scrum definuje 9 pravidel a stojí tak hned před Kanbanem.

Nikde není řečeno, že týmy musí respektovat přesně jednu metodiku. Proto, jak i praxe dokazuje, se osvědčuje prolnutí různých prvků z různých metodik. Takové synergické využití Scrumu a Kanbanu předvedu v další části práce. Nyní následují tři neformální pravidla, které mohou být použity v obou metodikách, ale každá metodika k nim přistupuje vlastním způsobem.

2.1 PRIORITY V PRODUCT BACKLOG

Nastavení priorit požadavků je velmi důležité, protože určují, které úkoly mají přednost a které mohou být pozdrženy. Scrum tyto priority určuje přímo v seznamu požadavků na produkt (Product backlog) a tyto priority se pak přímo promítnou do dalšího sprintu.

² Rational Unified Process, viz <http://www-01.ibm.com/software/awdtools/rup/>.

Problémem je, že realizace dalšího sprintu může nastat za jeden den nebo až za 30 dní (délka jednoho sprintu) a nově prioritizovaný požadavek, tak bude značně pozdržen.

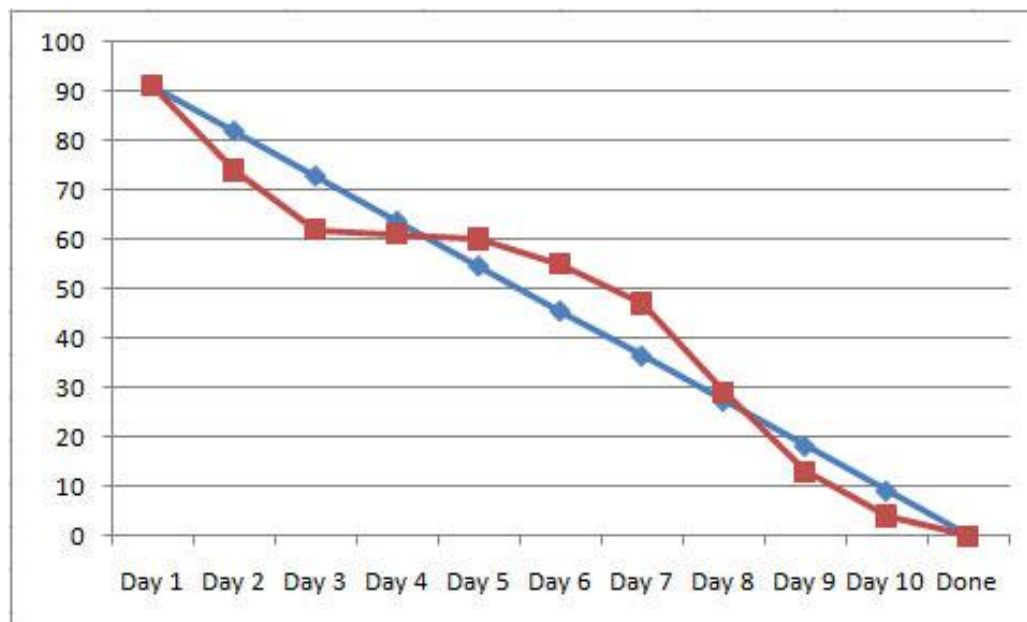
Kanban volí cestu agilnější, protože nemá přímý předpis na udržování seznamu požadavků a jejich řazení. Buď tedy tým realizující metodiku Kanban považuje levý sloupec svého schématu za období seznamu požadavků nebo si seznam vede samostatně. To už je na volbě každého týmu. Důležité je, že pokud chceme realizovat nový požadavek co nejdříve, zařadíme ho do tohoto seznamu a ve chvíli, kdy se uvolní kapacita sloupce následujícího (nebo prvního), požadavek může být realizován.

2.2 KAŽDODENNÍ PORADY (DAILY STANDUP MEETINGS)

Podobnou situaci můžeme vidět v řešení každodenních porad. Tato praktika může být určitě přínosná, je ale zapotřebí na ni mít exaktní pravidlo? Jak uvádí KNIBERG (2009), přestože Kanban toto pravidlo přímo neobsahuje, praxe potvrzuje, že týmy praktikující Kanban také pořádají každodenní porady. Většinou jsou si totiž týmy vědomy jejich přínosů v podobě šíření informací o stavu projektu v rámci celé organizace. Je samozřejmé, že reálný přínos těchto porad záleží z velké části na jejich organizaci a formátu, ten ale nenařizuje ani Scrum.

2.3 VIZUALIZACE POSTUPU – GRAFY

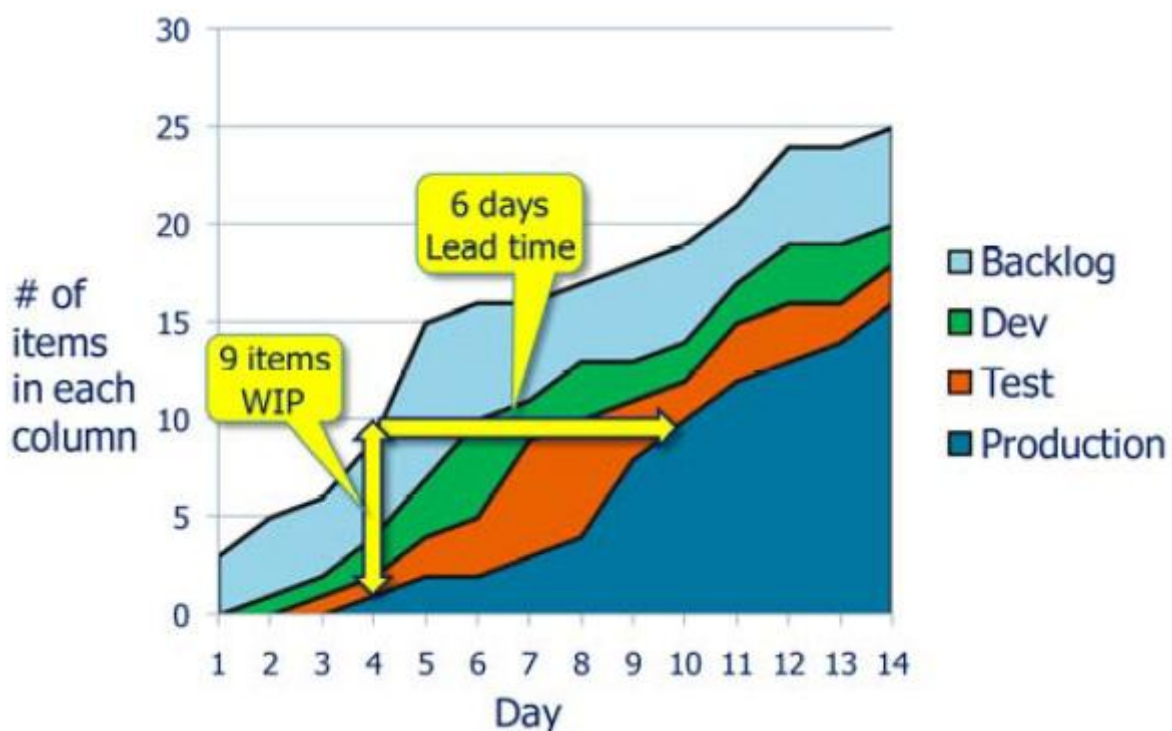
Metodika Scrum používá tzv. burndown grafy jako hlavní prostředek pro vyjádření množství práce, která zbývá a k zobrazení odchylek od plánu. Jejich cílem je podávat srozumitelnou informaci o tom, jak plníme plán a umožňuje nám tak reagovat na vzniklá zpoždění. Takový jednoduchý typ grafu lze vidět na následujícím Obrázek 2-2.



OBRÁZEK 2-2 BURNDOWN GRAF V METODICE SCRUM

Na horizontální ose jsou vyznačeny jednotlivé dny sprintu, na ose vertikální pak lze nalézt množství odvedené práce v procentech z celkového množství v sprint backlogu. Dle autora KNESL (2009) jsou tyto grafy běžněji uváděné v rámci jednoho sprintu. Pro tým je pak dobře zřetelné, jak včas jsou schopni splnit úkoly, které si vybrali z product backlogu. Je ale možné pomocí burndown grafů zobrazovat průběh prací za celý projekt.

Kanban tradičně nepředepisuje žádný typ grafů, zároveň je ale ani nezakazuje. Vzhledem k tomu, že Kanban nedefinuje přesně omezené iterace, nemají klasické burndown grafy příliš velký význam. Většina Kanban týmů tak využívá spíše tzv. kumulativní tokové diagramy, které vhodně zachycují průchod úkolů jednotlivými fázemi projektu. Příklad takového grafu je na následujícím Obrázek 2-3:



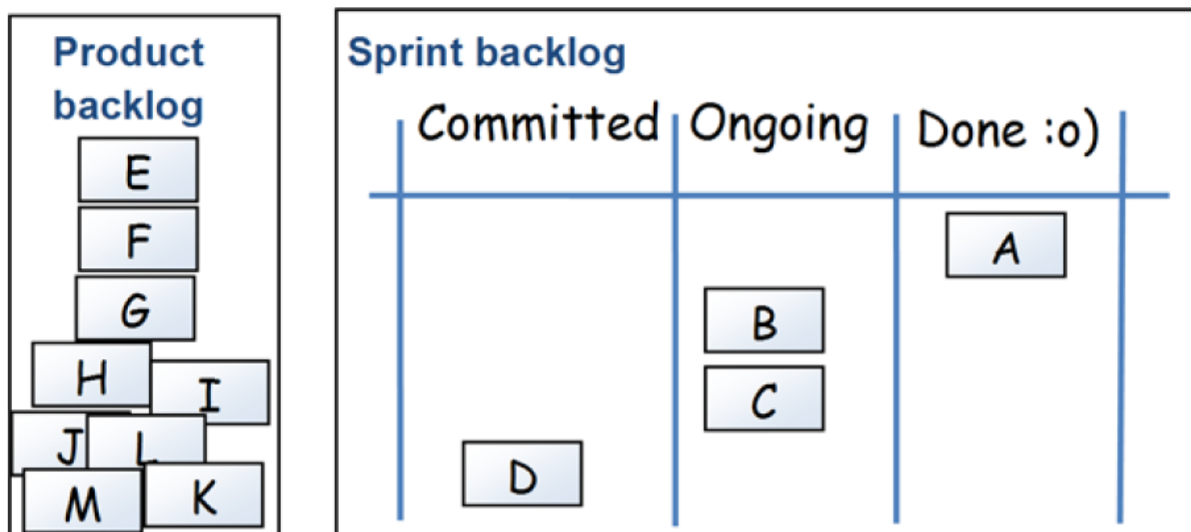
OBRÁZEK 2-3 KUMULATIVNÍ TOKOVÝ DIAGRAM V METODICE KANBAN

Osa X na tomto diagramu zachycuje množství dní trvání projektu a osa Y množství úkolů. Výkonnost týmu je pak vyjádřena délkou celého cyklu, tedy jak dlouho trvá, než úkol projde všemi fázemi vývoje (vzdálenost na ose X). To kolik položek je paralelně zpracováváno v jedné fázi ukazuje rozpětí na ose Y. Relativní výkonnost týmu lze vyčíst ze sklonu celého grafu, čím je strmější, tím rychleji se úkoly dostávají do poslední fáze a systém má větší propustnost.

3 SCRUM/KANBAN TABULE

Vizuální zobrazení právě realizovaných a plánovaných úkolů je velice důležité, protože podává jasný přehled o aktuálním stavu projektu. I k tomuto aspektu přistupují metodiky Scrum a Kanban mírně odlišně, ale s vidinou podobného cíle.

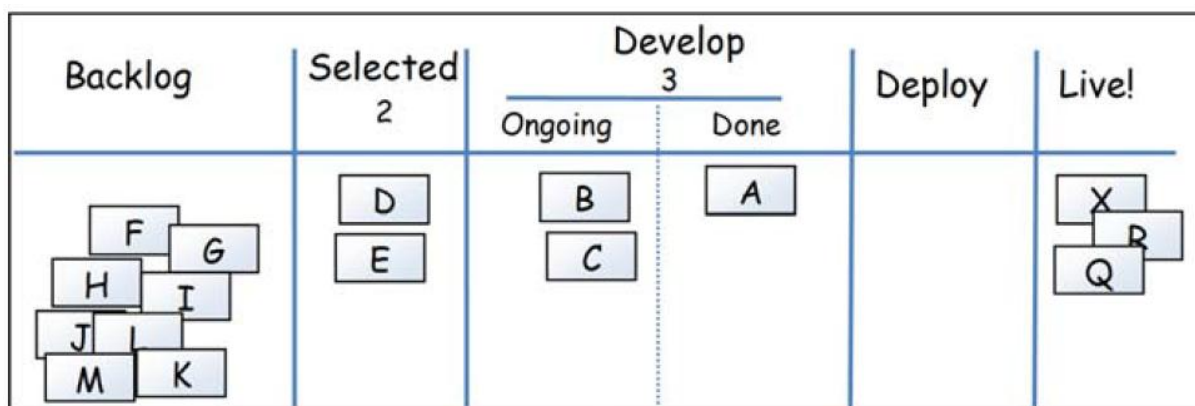
Scrum stanovuje dva základní vizuální prvky s úkoly, kterými jsou product backlog a sprint backlog. Zatímco product owner může přidávat nové úkoly a měnit jejich pořadí v product backlogu, ve chvíli, kdy si tým vybere úkoly do svého sprint backlogu, nelze je zde již měnit a je nutné vyčkat do dalšího sprintu. Scrum tedy zavádí dvě oddělené tabule, kde ta druhá – sprint backlog podává pouze část celkového obrazu stavu systému a zobrazuje ty úkoly, které si příslušný tým vybral ke splnění pro daný sprint. Následující Obrázek 3-1 ukazuje příklad jednoduché tabule používané v rámci Scrum.



OBRÁZEK 3-1 TABULE S ÚKOLY VE SCRUMU

Kanban tento vizuální prvek definuje v podstatě nepřímě. Vychází z pravidla o zobrazení toku práce, které říká, že celý cyklus je rozdělen na jednotlivé fáze a ty jsou zobrazeny ve sloupcích. Realizujeme-li Kanban, nic nám nebrání označit první sloupec jako product backlog, s neomezenou kapacitou úkolů a nastavit pravidlo, které bude definovat, jakým způsobem jsou úkoly přesouvány do dalšího sloupce, uvolní-li se kapacita. Výhodou tohoto řešení je, že stále máme komplexní obraz o celém projektu na jednom místě.

Realizujeme-li však příliš rozsáhlý projekt s množstvím dílčích úkolů, může takový systém vytvářet nepřehledné prostředí a zvážíme, jestli přeci jen nebude vhodné rozdělení do autonomních celků. Je pak zapotřebí přijmout určité úpravy původního schématu. Následující obrázek ukazuje řešení tabule pro Kanban, kde u každého sloupce je zároveň zaznamenaný limit pro počet úkolů v dané fázi.



OBRÁZEK 3-2 SOUHRNNÁ TABULE V METODICE KANBAN

3.1 VLASTNICTVÍ TABULE

Důležitým aspektem obou metodik je přístup k rozdělení zodpovědností za správu tabule s úkoly. V předchozí části jsem uvedl, že Scrum odděluje společný product backlog a sprint backlog. Zatímco product backlog je sdílený pro všechny týmy podílející se na projektu, sprint backlog je plně spravován samostatným týmem. Většinou lze očekávat, že sprint backlogů bude v rámci projektu více a jejich množství bude shodné s množstvím spolupracujících týmů.

Takový přístup implikuje, že tabuli může editovat pouze tým, kterému náleží. Ostatní členové organizace sice mohou vidět úkoly jiných týmů, ale nemohou do nich zasahovat. Navíc jsou tabule po každém sprintu vymazány a úkoly odstraněny. Kanban nedefinuje tabule na úrovni týmů, ani je nemaže. Defínuje jednu kontinuální tabuli, kterou sdílí všechny týmy a probíhá přes ni veškerý tok úkolů. Tabule se tak neváže k týmu, ale ke stanovenému workflow celého projektu. Autor SCHIFFER (2011) například přímo doporučuje upravit metodiku Scrum a využívat zde tabuli ve stylu Kanbanu pro správu problémů v projektu. S tímto přístupem souhlasím, neboť si myslím, že týmy by měly mít možnost zasáhnout do prací v rámci celého projektu.

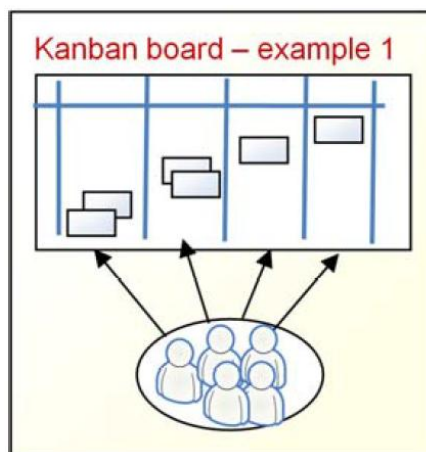
4 ROZDĚLENÍ DO TÝMŮ

Na téma rozdělení členů do týmů narazila již předchozí kapitola v souvislosti s tabulí zobrazující seznam úkolů. Tabule ve Scrumu je editována pouze jedním týmem, ten ale musí být složen ze všech různých profesí.

Účelem Scrumu tedy není definovat jeden tým specializující se například pouze na testování a jiný na vývoj. Cílem je mít naprosto univerzální týmy, které mají schopnosti (prostřednictvím svých členů) realizovat všechny typy činností, aby dokončili část výsledného systému. Pokud se nám podaří takové týmy postavit, lze si pak i představit, že se opravdu budou soustředit

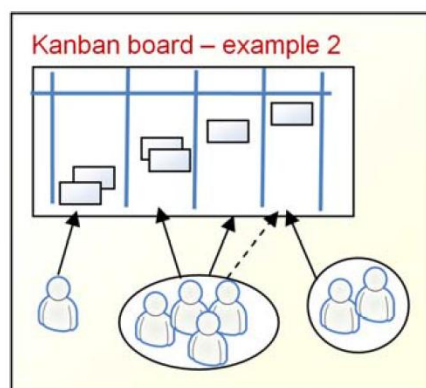
pouze na svou množinu úkolů a nebudou mít příliš velkou potřebu ovlivňovat činnosti jiných týmů.

Kanban žádné takové pravidlo nemá a znovu je tedy volnějším. Složení týmů tak většinou bude záviset na aktuálních možnostech a potřebách projektu. Existují dva základní přístupy, jak můžeme týmy rozdělovat. První definuje tým jako univerzální, stejně jako ve Scrumu. Oblast působnosti takového týmu se pak dotýká všech fází vývojového cyklu. Příklad je znázorněn na Obrázek 4-1.



OBRÁZEK 4-1 UNIVERZÁLNÍ TÝM V KANBANU

Druhý přístup k nastavení týmů v Kanbanu může být takový, že definujeme několik menších specializovaných týmů, kde se každý tým soustředí na vlastní činnost (tedy fázi). Takové rozložení můžeme použít například ve chvíli, kdy víme, že jedna z fází je několikanásobně náročnější, než ty ostatní. Například v případě již zmiňované uživatelské podpory budeme potřebovat mnohem silnější zázemí u servisních techniků. Jejich tým tak může být specializovaný a věnovat se pouze činnosti servisních zásahů. Příklad takového uspořádání ukazuje Obrázek 4-2.



OBRÁZEK 4-2 SPECIALIZOVANÉ TÝMY

5 OMEZENÍ MNOŽSTVÍ PARALELNÍCH ÚKOLŮ (WIP³)

Přístup Scrumu a Kanbanu k tomuto aspektu je zřejmý z předchozích definic. Zatímco Kanban omezuje zpracovávané položky explicitně v rámci dané fáze, Scrum toto množství omezuje pouze nepřímým, tím, že definuje délku trvání iterace (neboli sprintu).

Jednoduše shrnuto tedy:

- Scrum limituje WIP dle jednotky času.
- Kanban limituje WIP pomocí nastavení fáze vývoje.

Jaký přístup je ale vhodnější? Pravděpodobně ten, který nám nabízí větší možnosti přesného řízení průběhu projektu. Máme-li nastavené limity po vzoru Kanbanu (tedy absolutní množství úkolů v rámci fáze) a předpokládáme-li přibližně stejně náročné úkoly, můžeme přesně stanovit, jakou dobu trvá, než úkol projde celým cyklem. Tento čas bývá nazýván „lead time“ a označuje hlavní vlastnost celého systému – jak rychle jsme schopni realizovat jeden úkol.) Naopak pokud máme omezení projektu, které říká, že musíme každých 30 dní vydat hotový prototyp, je výhodné využití Scrumu a sladění sprintů s těmito milníky.

Z pohledu flexibility a lean principů je tedy přístup Kanbanu vhodnější, což dokazuje i KNIBERG (2009), který uvádí, že mnoho Scrum týmů zavádí omezení množství souběžných úkolů. Je samozřejmě ale možné narazit i na situace, kdy budou pravidelné stejné iterace rovněž výhodou. Nicméně samotná informace o výkonnosti systému je pro organizace velice důležitá, protože právě na jejím základě je pak možné uzavírat smlouvy o úrovni poskytovaných služeb a vytvářet tak realistické plány do budoucna. Je tedy cílem obou metodik znát kadenci řešení jednotlivých úkolů.

6 VOLBA METODIKY

Porovnal jsem základní vlastnosti obou metodik a vyzdvihl ty, o kterých si myslím, že jsou pro úspěšné řízení projektů důležité. Nyní se pokusím formulovat dílčí závěr a vyvodit doporučení volby metodiky dle typu projektu.

V agilních přístupech není možné jednoznačně kategorizovat a říci na typ projektu „ABC“ je potřeba použít metodiku „XYZ“. I z vlastní praxe vím, že je výhodnější brát metodiky, jako souhrn doporučení a nápadů a sám si sestavit systém řízení na míru projektu, organizaci a týmům. I přesto je ale možné stanovit, která metodika reaguje lépe na určité aspekty projektů.

Z předchozí práce jednoznačně vyplývá, že Kanban je oproti Scrumu mnohem flexibilnější a volnější. To jak z pohledu množství pravidel, která definuje, tak i z pohledu rigoróznosti

³ WIP, neboli Work In Progress označuje množství práce vykonávané souběžně, obě metodiky kvantifikují množství práce na úrovni jednotlivých úkolů.

těchto pravidel. Nelze říci, že je takový přístup horší nebo lepší, záleží na typu projektu a nahodilosti událostí. Obecně může být přínosnější, pokud máme přesně definovaný proces vývoje, ale pouze do té míry, kdy samotný proces nebrzdí výkon práce. S ohledem na tyto předpoklady tedy definuji 3 základní faktory, které ovlivní volbu metodiky.

1. **Nahodilost požadavků** – Způsob jakým vstupují požadavky do systému a to, jak potřebují být vyřešeny je zřejmě hlavním faktorem pro volbu metodiky. Požadavky mohou přicházet buď v daných intervalech a ne příliš často. Takovým případem je klasický vývojový projekt, který se implementuje na základě funkční specifikace a nové požadavky vstupují do systému zřídka, nebo na základě vydání nové verze. Pro takový systém je přístup metodiky Scrum vhodný, protože časově omezené iterace nebudou nové požadavky příliš omezovat. Druhým příkladem může být uživatelská podpora. Její běžnou náplní činnosti je, že v průběhu dne neustále přichází tzv. tickety (požadavky) a podpora je řeší. Pro takový systém není vhodné definovat omezené časové iterace, ale spíše zvolíme přístup Kanbanu s nekonečným proudem požadavků řešených průběžně. Nic tak nebude bránit co nejrychlejšímu zpracování každého nového ticketu.
2. **Ukázky software v definovaných intervalech** – Druhým faktorem může být požadavek zadavatele na vydání určité verze/funkcionality systému každých X dní. V takovém případě je výhodné sladit délku sprintů ve Scrumu s tímto požadavkem a lépe se tak přizpůsobit termínům.
3. **Velikost týmu** – Pro úplně malé týmy nemá velký smysl definovat odděleně role Scrum mastera a Product ownera, ale spíše vystačíme s jedním vedoucím týmu. Nebo dokonce může tento vedoucí vést několik týmů zároveň. Obecně pak lze říci, že pro větší týmy je vhodnější více formální metodika (v rámci této práce to je metodika Scrum) s více pravidly. Naopak malé týmy je možné řídit více ad hoc přístupem (v rámci této práce to je metodika Scrum).

Je tedy na rozhodnutí konkrétního managementu, jakou metodiku zvolí, neboť pouze vedení firmy může znát potřeby vlastních projektů. I přesto si ale myslím, že výše uvedené faktory mohou takové rozhodnutí ulehčit.

7 SOUHRNNÉ SROVNÁNÍ

V této kapitole uvedu závěrečné souhrnné srovnání obou metodik, dle jednotlivých, většinou již dříve uvedených aspektů. Tabulka vychází z KNIBERG (2009).

Scrum	Kanban
Napevno nastavená délka trvání iterace (sprintu).	Ohraničené iterace jsou pouze volitelné. Přístup je spíše řízen událostmi.

Tým se zavazuje ke splnění určitého objemu práce v rámci jedné iterace.	Neexistují iterace, týmy pracují průběžně.
Základní metrikou je rychlost vyjádřená v množství práce vykonané v rámci sprintu. Používá se pro měření činnosti a hodnocení zlepšení procesů.	Základní metrikou je doba jednoho cyklu. Používá se pro měření činnosti a hodnocení zlepšení procesů.
Scrum předepisuje takové složení jednotlivých týmů, aby každý tým obsahoval všechny role specializací. (např. návrh, vývoj, testování, nasazení)	Namíchané týmy jsou volitelné. Zároveň Kanban ani nezakazuje čistě specializované týmy.
Jednotlivé úkoly musí být rozděleny, aby mohli být realizovány v rámci jednoho sprintu.	Nepředepisuje konkrétní velikost úkolu.
Vizualizace práce pomocí burndown grafů.	Nepředepisuje typ grafu, doporučuje se ale využití tokových diagramů. (viz kapitola 2.3)
Množství souběžné práce limituje nepřímo, prostřednictvím délky sprintu a kvality týmů.	Množství souběžné práce omezuje přímo prostřednictvím limitu úkolů v rámci každé fáze.
Tým by měl provádět odhady času.	Přímo nedefinuje.
Nelze přidat položku do již probíhající iterace.	Zpracování nové položky lze zahájit, kdykoliv je volná kapacita.
Sprint backlog je v režii příslušného týmu.	Tabule s úkoly může být editována jakýmkoliv týmem/členem týmu.
Definuje 3 základní role.	Nedefinuje žádné role.
Tabule s úkoly je vymazána po každém sprintu.	Na tabuli zůstávají všechny úkoly, pouze se přesouvají mezi fázemi.
Product backlog definuje jako seznam s prioritami.	Nastavení priorit je volitelné, ale doporučované.

8 ZÁVĚREM

Práce krátce představila agilní metodiky Scrum a Kanban. Stěžejní část této práce pak byla věnovaná porovnání těchto metodik dle jednotlivých aspektů a snažila se vyvodit určité doporučení, kdy použít více prvků z metodiky Scrum a kdy z metodiky Kanban.

Jako hlavní faktor rozhodující o volbě metodiky jsem označil charakter vzniku požadavků. Pokud přicházejí nové požadavky v malé míře a v alespoň trochu předpověditelných okamžicích, dobře se hodí Scrum, neboť svými iteracemi nebude příliš omezovat zpracování úkolů. Naopak například v případě uživatelské podpory (helpdesku) přicházejí nové

požadavky velmi náhodně a ve vysoké frekvenci a je vhodnější je zpracovávat přístupem Kanbanu, který naprosto zastupuje kategorii „pull“ systémů. Mohli bychom použít i řešení Scrumu, ale bylo by zapotřebí dostatečně zkrátit iterace. Pokud však ve Scrumu zkrátíme iterace na minimální míru, dostáváme se v podstatě na specifika Kanbanu.

Závěrem několik slov o možnostech selhání projektu. Jak uvádí četné studie (SUTHERLAND, 2010) přímo metodika sama selhat nemůže. Většinou je selhání na straně vedení projektu nebo rovnou celé firmy. Tyto potíže se mohou projevit ve špatné komunikaci, nesprávném pochopení pravidel metodiky nebo například špatné spolupráci týmu. Výběr jednotlivých pravidel z množiny agilních metodik by tak neměl být podceněn, protože bude ovlivňovat průběh celého projektu.

POUŽITÉ ZDROJE

KNIBERG, Henrik. Kanban vs Scrum. [online]. 2009 [cit. 2012-11-24]. Dostupné z: <http://www.crisp.se/henrik.kniberg/Kanban-vs-Scrum.pdf>

KNESL, Jiří. Agilní vývoj: Scrum. [online]. 2009 [cit. 2012-11-24]. Dostupné z: <http://www.zdrojak.cz/clanky/agilni-vyvoj-scrum/>

SCHWABER, Ken. 2004. Agile Project Management with Scrum. Microsoft Press. ISBN 978-0-7356-1993-7.

DUBAKOV, Michael. Kanban in Software Development. [online]. 2009 [cit. 2012-11-24]. Dostupné z: <http://www.targetprocess.com/blog/2008/09/kanban-in-software-development.html>

GROSS, John M., MCINNIS, Kenneth R. 2003. Kanban made simple: demystifying and applying Toyota's legendary manufacturing process. New York: AMACOM Div American Mgmt Assn, 2003.

SUTHERLAND, Jeff. Scrum handbook. [online]. 2010 [cit. 2012-11-24]. Dostupné z: <http://jeffsutherland.com/scrumhandbook.pdf>

CRISP AB. *Crisp: Kanban* [online]. [cit. 2012-12-03]. Dostupné z: <http://www.crisp.se/gratis-material-och-guider/kanban>

SCHIFFER, Bernd. 7 Steps to Build a Kanban Board for a Scrum Team's Impediments. [online]. 2011 [cit. 2012-11-24]. Dostupné z: <http://agiletrail.com/2011/09/05/7-steps-to-build-a-kanban-board-for-a-scrum-teams-impediments/>

SEZNAM OBRÁZKŮ

Obrázek 1-1 Schéma rozdělení fází v metodice Kanban	4
Obrázek 2-1 Množství pravidel u agilních metodik	5
Obrázek 2-2 Burndown graf v metodice Scrum.....	6
Obrázek 2-3 Kumulativní tokový diagram v metodice Kanban	7
Obrázek 3-1 Tabule s úkoly ve scrumu.....	8
Obrázek 3-2 Souhrnná tabule v metodice Kanban.....	9