

Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky

Katedra informačních technologií



Kanban a jeho využití při vývoji softwaru

5. 12. 2011

Jiří Krejčí (xkrej44@vse.cz)

Abstrakt

Cílem této práce je představit metodiku vývoje softwaru Kanban a následně jí porovnat s jinou agilní metodikou pro vývoj softwaru - Scrum.

Abstract

To introduce agile method for software development Kanban and compare Kanban with method Scrum is the main aim of this thesis.

Obsah

1	Úvod	3
2	Historie Kanbanu	5
3	Popis metodiky	7
4	Porovnání s metodikou Scrum - 4 strany	9
4.1	Scrum ve zkratce	9
4.2	Kanban ve zkratce	10
4.3	Odlišnosti metodik Scrum a Kanban.....	11
5	Závěr.....	17
6	Literatura	18
	Seznam obrázků.....	19

1 Úvod

Cílem této práce je představit Kanban jako nástroj pro vývoj softwaru a následně jej porovnat s jinou metodikou, která se využívá na vývoj softwaru – Scrum. Obě tyto metodiky řadíme mezi takzvané metodiky agilní, které se vyznačují odlišným přístupem k vývoji softwaru než metodiky rigidní. Mezi tradiční, rigidní (tuhé) metodiky můžeme zařadit Vodopádový model, Spirálový model nebo IBM metodiku RUP. Nový přístup v podobě agilních metodik se začal prosazovat kolem roku 2000. Základní premisou pro vznik nových metodik byla naše neustále se měnící a hlavně zrychlující se doba. Dnešní doba si vyžaduje rychlou reakci na vzniklé problémy a to platí zejména při vývoji softwaru. Agilní metodiky právě toto oproti metodikám rigidním splňují. Podstatou agilních metodik je rychlá reakce na změny a jsou postaveny na těchto principech, které jsou společné pro všechny druhy metodik. Jsou to:

- Iterativní a inkrementální vývoj s krátkými iteracemi
- Komunikace mezi zákazníkem a vývojovým týmem
- Průběžné automatizované testování

Za nejznámější typy agilních metodik můžeme považovat tyto:

- Extrémní programování
- Feature-Driven Development – Vývoj řízený vlastnostmi
- SCRUM
- Adaptivní vývoj software
- Lean Development
- Vývoj řízený testy
- Crystal metodiky

V této práci se podrobněji zaměříme na agilní metodiku Scrum a budeme jí porovnávat s metodikou Kanban, která je známá zejména při výrobě automobilů v japonské továrně Toyota. Vyznačuje se zejména dodávkou materiálu Just-in-time. Tato metoda se však postupem času dostala i do oblasti vývoje softwaru a rozhodně jí můžeme zařadit právě mezi metodiky agilní.

Představení metodiky Kanban a její verze pro vývoj softwaru, její výhody a nevýhody, je dílčím cílem této práce. Hlavním cílem této práce je pak porovnání této metodiky s metodikou s agilní metodikou Scrum. K těmto cílům chce autor dojít za pomoci této struktury:

Ve druhé kapitole si představíme metodiku Kanban z jejího historického hlediska, podíváme se, co stálo za vznikem této metodiky a kde vlastně vznikla, Zjistíme, jaký dopad měla na podnikové výsledky a celý další průmysl.

Ve třetí kapitole se podíváme na metodiku Kanban při vývoji softwaru, zjistíme, co patří mezi klíčové prvky této metodiky a na jakých základech je postavena z hlediska vývoje softwaru. Představíme si její klady a zápory, tedy její výhody a nevýhody.

Čtvrtá kapitola bude zaměřena na porovnávání mezi metodikou Kanban a metodikou Scrum. Zjistíme, v čem se tyto metodiky liší zásadně a v čem jen okrajově. Zamyslíme se také nad tím, co by se z obou metodik dalo vybrat a sloučit je do nové metodiky, která by pak mohla být třeba ještě efektivnější.

V závěrečné kapitole bude provedeno zhodnocení celé práce. Posoudíme, zdali bylo dosaženo námi vytýčených cílů a jaký přínos by tato práce mohla mít pro společnosti zabývající se vývojem softwaru pomocí agilních metodik.

2 Historie Kanbanu

V japonštině slovo „Kan“ znamená signál a slovo „Ban“ znamená karta nebo tabule. (Joyce, 2009)

Japonské slovo Kanban, které můžeme volně přeložit jako „tabule se štítky“ se stalo synonymem pro plánování požadavků. Vznik Kanbanu je spojen s prvními roky vzniku společnosti Toyota. Mluvíme o období pozdních 40. a brzkých 50. letech minulého století. Taiichi Onho vyvinul Kanban ke kontrole výroby mezi procesy a k implementaci Just In Time¹ (JIT) výrobních zdrojů v továrně Toyota. Tyto ideje nezískaly celosvětové přijetí až do globálního ekonomického poklesu v 70. letech minulého století. Při využívání Kanbanu, minimalizoval práci v samotných procesech, mezi procesy a redukoval náklady spojené s držetím zásob. (John M. Gross, 2003)

Toyota tuto metodu využila k úspoře nákladů a řízení využití strojů. V dnešní době již Toyota nepokračuje ve využívání tohoto systému pouze z hlediska úspor nákladů, ale také k identifikování překážek toků a příležitostí pro kontinuální zlepšení. Je však nutné podotknout, že JIT výrobních prostor byla koncipována p. Kiichero Toyoda, zakladatelem společnosti Toyota Motor. Přesto to byl p. Onho kdo vyvinul strategii Kanban, která se stala jedním z klíčových pilířů úspěšné implementace JIT výrobních zdrojů v Toyotě Motors. (John M. Gross, 2003)

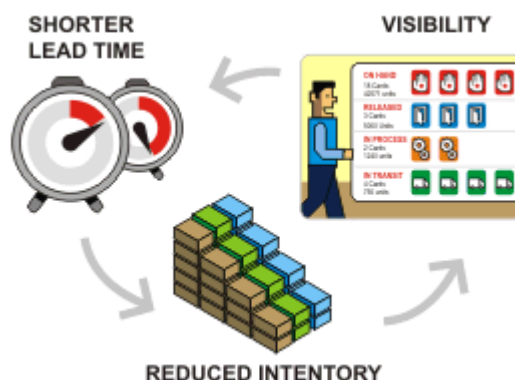
Při organizaci práce v Kanbanu pracovníci využívají vizuální signály k determinování aktuální fáze práce, fáze ukončení nebo fáze zachycující změny v průběhu. Kanban pravidla také nařizují operace během výskytu nějakého problému a na koho se obrátit když tento problém nastane. Konečně dobře naplánovaný Kanban má vizuální indikátory které umožňují manažerům a nadřízeným vidět na první pohled stav pracovního plánu. (John M. Gross, 2003)

V procesech kontrolovaných Kanbanem, pracovníci produkují produkty zaměřené na aktuálním užití než na předpokládaném využití. Kanban program nahrazuje tradiční týdenní nebo denní výrobní program, který většina za nás znali ve výrobních operacích. Tento stávající rozvrh prací je nahrazen vizuálními signály a

¹ Just In Time – logistická technologie využívaná při výrobě

předefinovanými rozhodovacími pravidly, které umožňují výrobní operace provádět v jedné linii. (John M. Gross, 2003)

Kanban, který může mít mnoho forem, řídí operace procesu na denní bázi. Kanban nenahrazuje plánování materiálu, ale raději bere v potaz informaci o materiálové plánování a využívá ji k vytvoření Kanbanu. (John M. Gross, 2003)



Obrázek 1 - Kanban

Fungování metody Just-in-Time (JIT) lze vysvětlit v zásadě dvojitým způsobem. Modernější přístup ji charakterizuje jako systém, který vede k úspoře času v celé průběžné době výrobku. Tím přináší výrazné snížení nákladů a zvýšení produktivity práce. Druhé, tradiční pojetí spočívá ve využití JIT pro řízení jednotlivých stupňů výroby či mezi jednotlivými provozny. Obecně je pak v případě jednotlivých pracovišť možno jako JIT chápat i princip Kanban. (Tuček, 2004)

Nejčastěji zdůrazňovaný přínos je snížení zásob. Dále zajištění systémového toku informací v celém procesu výroby a dodávek dílů, založeného na sledování předem určeného stavu zásob. Podpora plynulosti výroby při nárůstu sortimentu, zmenšení pracnosti plánování (tzn. tvorby plánu i jeho kontroly). V neposlední řadě se jedná se o otevřený systém pro řízení, který umožňuje lepší přehledu o stavu výroby a zásob rozpracované výroby i úsporu přepravních nákladů. (Tuček, 2004)

Kanban přinesl do výroby obrovské zefektivnění a podpořil tak výrazné snížení nákladů výrobních podniků. S nástupem agilních metodik při vývoji softwaru se metodika Kanban stala velmi zajímavou i pro vývojáře softwaru. Vysvětlení, jakým způsobem funguje Kanban při vývoji softwaru, je předmětem následující kapitoly.

3 Popis metodiky

Všeobecně můžeme Kanban považovat za tzv. „pull systém“, který je na rozdíl od „push systému“ založen na sledování potřeb trhu. Pull systém produkuje pouze požadované množství čehokoliv. V ideálním případě najdeme zákazníka, vyprodukujeme mu, co on potřebuje a prodáme mu to. V každém případě pull systém redukuje zásoby. (Dubakov, 2008)

Dle (Anderson, 2010) je metoda Kanban určený postavená na těchto pěti základních premisách:

1. Zobrazení pracovního toku

Dnes pracovní tok práce ve své podstatě není vidět, protože je "skrytý" v informačních systémech. Vizualizace pracovního toku a jeho zviditelnění je jádrem k porozumění a pochopení celé práce. Bez pochopení workflow je těžší dělat správné změny. Obyčejný způsob, jak představit pracovní postup je za použití karet nebo kartiček na tabuli ve sloupcích. Sloupce představující různé fáze nebo kroky v pracovním postupu.

2. Omezení práce v průběhu

Omezení práce v procesu vyjadřuje, že tento „Pull systém“² je implementován po částech nebo jako celý pracovní tok. Pull systém bude fungovat jako jeden z hlavních stimulů pro průběžné, přírůstkové a vývojové změny v systému. Pull systém může být implementován jako systém Kanban, CONWIP systém, systém DBR, nebo nějaká jiná varianta. Důležité prvky, které pracují v průběhu výrobního procesu v každé fázi pracovního postupu, jsou limitovány. Nová činnost je "táhnuta" až když je k dispozici kapacita v rámci místních WIP limitu.

3. Správa toku

Každý tok v práci by měl být v každé fázi monitorován, měřen a reportován. Prostřednictvím aktivního průběžného řízení toku, postupné a vývojové změny mohou být vyhodnoceny jako pozitivní nebo negativní na systém.

² Pull systém – je systém, který je tažen na základě požadavků trhu, neboli v našem případě na základě požadavků na software

4. Vytváření explicitních procesních politik

Do té doby, než je mechanismus procesů jasně explicitně vyjádřený, je často těžké nebo nemožné udržet diskusi o vylepšení. Bez jasného porozumění věci a toho, jak si práce aktuálně stojí, vede k vypjatým, emočním a subjektivní diskusím o problémech. Pouze s jasně daným vysvětlením je možné se posunout racionálně a objektivně v projednávání otázek. Je více pravděpodobné že tak získáme konsenzus o vylepšení našeho návrhu řešení.

5. Zlepšení vzájemné spolupráce (využívání modelů & vědeckých metod)

Metodika Kanban podporuje menší kontinuální, inkrementální a vývojové změny. Když jednotlivé týmy chápou teorii problému o práci, pracovním toku, procesech a rizicích stejně, jsou pravděpodobně více ochotni sdílet pochopení o problému a navrhnou nová řešení, která mohou být odsouhlasena za pomoci konsenzu. Metodika Kanban naznačuje, že vědeckého přístupu je použito nepřetržitě, postupně a za pomoci vývojových změn. Tato metoda nepředepisuje konkrétní vědeckou metodu použití.

4 Porovnání s metodikou Scrum

V následující kapitole se zaměříme na rozdíly a odlišnosti těchto agilních metodik a podíváme se na možnosti jejich vzájemného doplnění, tak abychom dostali ještě lepší metodiku pro vývoj našeho nového softwaru.

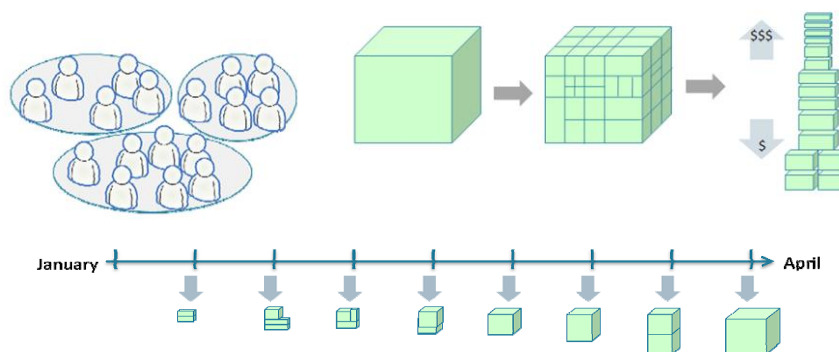
Raději než se soustředit na to být Agilní, což může vést k úspěchu se Kanban zaměřuje na to být úspěšný, což by mohlo vést k tomu, abychom byli Agilní. (Joyce, 2009)

V následujících 2 kapitolách (Kniberg, 2009) zhodnotil obě tyto metodiky ve zkratce.

4.1 Scrum ve zkratce

- Rozděl svoji organizaci do malých, samostatných celků napříč všemi funkcemi.
- Rozděl svůj úkol na malé a konkrétní výstupy. Seřaď je podle priorit a odhadni relativní náročnost každé z nich.
- Rozděl čas na krátké a časově neměnné iterace (1-4 týdny), kde každá iterace představuje kód, který je okamžitě připravený k demonstrování.
- Optimalizuj plán výstupů a aktualizuj priority ve spolupráci se zákazníkem
- Optimalizuj proces retrospektivně po každé iteraci

Všechny tyto pravidla jsou zobrazeny na následujícím obrázku. Viz Obrázek 2 - Pravidla Scrum



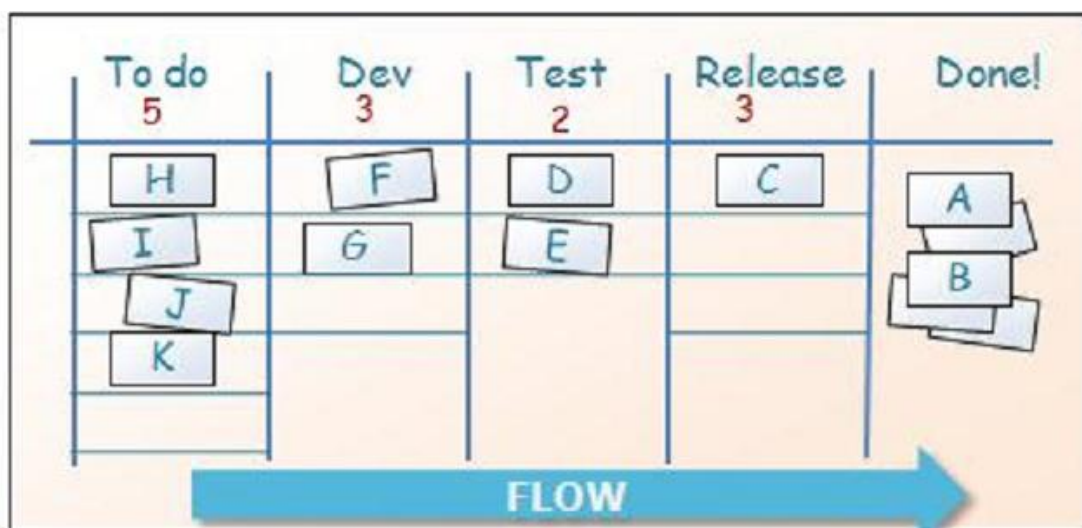
Obrázek 2 - Pravidla Scrum

Zdroj: (Kniberg, 2009)

4.2 Kanban ve zkratce

- Zobraz pracovní tok
 - Rozděl práci na části, napiš každou položku na kartu a dej ji na tabuli
 - Použij názvy sloupců k ilustrování, kde se daná položka v pracovním toku aktuálně nachází
- Omez práci v jednotlivých stavech (WIP) – explicitně přiřaď limity na počet položek, které mohou být v každém stavu pracovního toku
- Měř potřebný čas (průměrný čas k dokončení jedné položky, optimalizuj proces tak, aby potřebný čas k dokončení byl malý a předvídatelný, jak jen to možné.

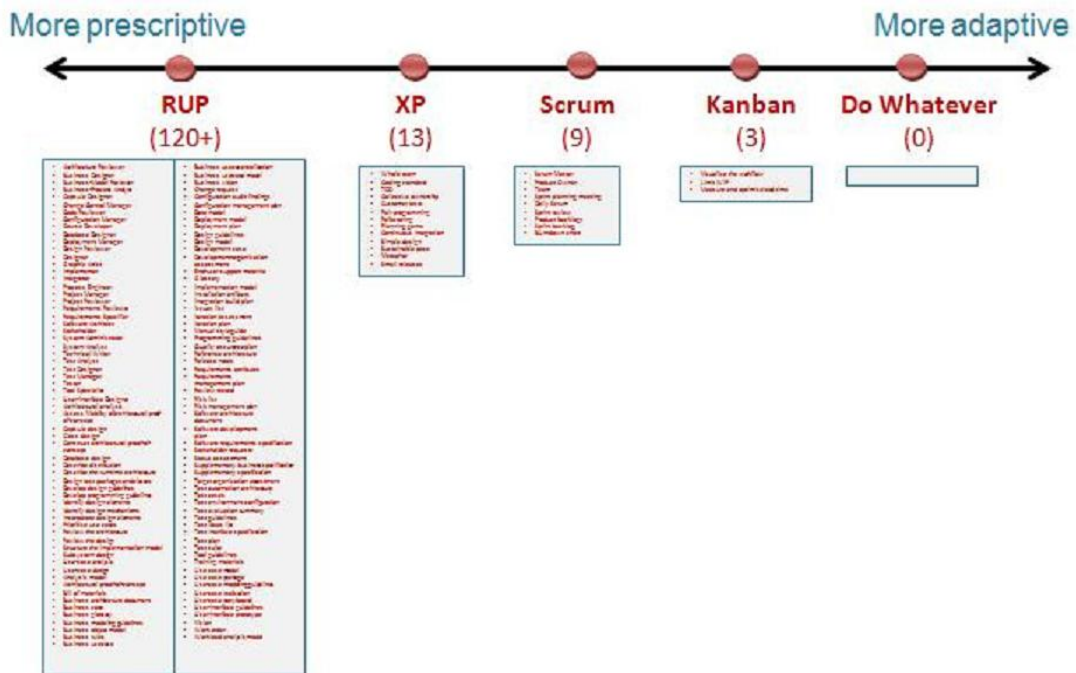
Všechny tyto pravidla jsou zobrazeny na následujícím obrázku. Viz **Chyba!**
Nenalezen zdroj odkazů.



Obrázek 3 - Pravidla Kanban

Zdroj: (Kniberg, 2009)

Z celkového pohledu je Kanban méně přísný, než Scrum. Například Scrum nařizuje využití časově ohraničených iterací, zatímco Kanban ne. V porovnání s ostatními metodikami, jsou tyto dva přístupy k vývoji softwaru mnohem „lehčí“ Viz



Obrázek 4 - Počet omezení jednotlivých metodik

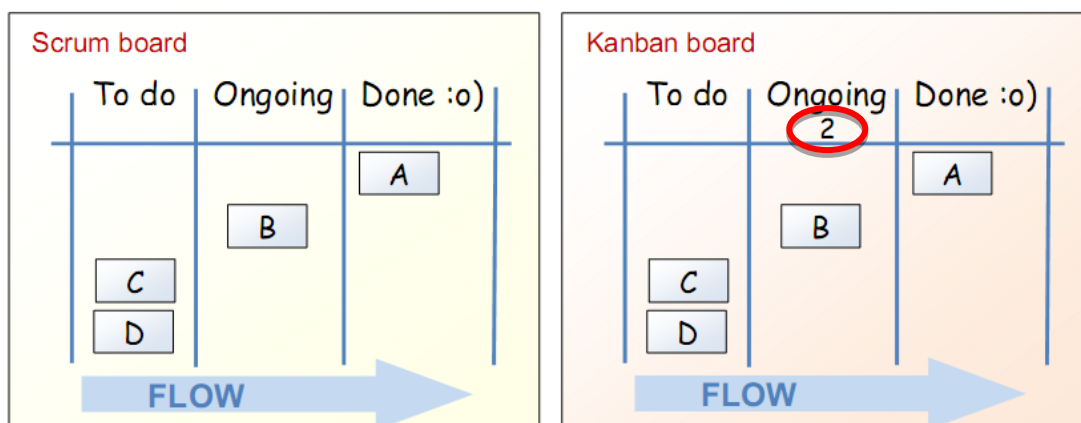
Zdroj: (Kniberg, 2009)

4.3 Odišnosti metodik Scrum a Kanban

Scrum předepisuje 3 role. (Vlastník, Tým, a Scrum Master) Zatímco Kanban žádné nepředepisuje. To však neznamená, že nemůžete mít vlastníka, ale prostě nemusíte. Pokud chceme, můžeme do procesu vývoje začlenit nějaké nové role, avšak každá nová role, by měla přinášet nějakou novou hodnotu pro daný projekt. Měli bychom se řídit pravidlem, že někdy méně je více. (Kniberg, 2009)

Scrum je založen na časově ohraničených iteracích. Délku iterace si můžeme vybrat, ale všeobecně bychom měli dodržovat její stejnou délku. Na začátku iterace jsou definovány týmy, a co budou dělat na základě priorit vlastníka. Na iteraci pracují až po konečnou prezentaci zainteresovaným stranám. V metodice Kanban nejsou žádné časově ohraničené iterace předepsány. Můžeme si vybrat, kdy budeme plánovat, vylepšovat procesy nebo přestavovat konečný produkt. Můžeme tyto aktivity nastavit na pravidelné bázi (např. každé pondělí) nebo na požadavek (vydat když máme něco hotové) (Kniberg, 2009)

Kanban omezuje práci v jednotlivých stavech Scrum nikoliv. Jinými slovy Scrum je omezuje nepřímou, zatímco Kanban přímo. Vysvětlení bude patrné z následujícího obrázku. Viz. Obrázek 5 - Porovnání WIP mezi Kanban a Scrum

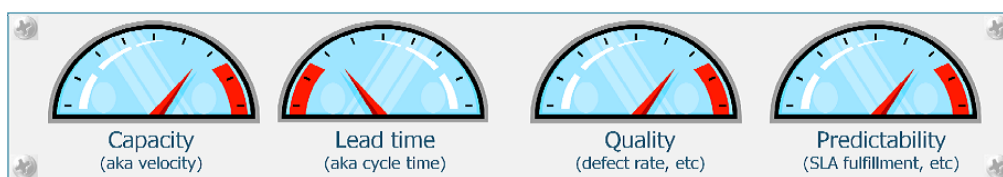


Obrázek 5 - Porovnání WIP mezi Kanban a Scrum

Zdroj: (Kniberg, 2009)

V obou případech sledujeme pohyb svazku položek v pracovním toku. Vybrali jsme 3 stavy.(To Do, Ongoing a Done) Můžeme zvolit jakékoliv stavy chceme, ale měli bychom mít na zřeteli, že méně je více. Jediný malý rozdíl mezi těmito dvěma obrázky je malá dvojka ve stavu „Ongoing“ na tabuli Kanban. Ta značí, že v tomto sloupci nemohou být více než 2 položky. V metodice Scrum není žádné pravidlo bránící týmu vložit všechny položky do jednoho sloupce ve stejný čas. Přesto je zde implicitní omezení poněvadž iterace sama o sobě má fixní rámec položek. V tomto případě je tento implicitní limit roven 4 položkám, poněvadž zde máme pouze 4 položky na celé tabuli. Proto Scrum omezuje nepřímou, zatímco Kanban přímo. Metodika Scrum omezuje WIP jednotkou času, Kanban je omezen počtem ve stavech. K lepšímu odhadu potřebného času, je potřeba omezit každý stav pracovního toku. Pokud máme jasně předvídatelný čas, můžeme tím lépe vyhovět požadavkům na SLA a vytvořit realistické plány. (Kniberg, 2009)

Kanban a Scrum jsou empirické metodiky ve smyslu, že se očekává Váš experiment s procesy a upravení podle Vašeho prostředí. V ideálním vývoji softwaru, bychom chtěli, abychom všechny tyto proměnné měli na maximum. Viz Obrázek 6



Obrázek 6 - Měřené proměnné

(Kniberg, 2009)

Bohužel, takhle to u obou metodik nefunguje a musíme se spolehnout pouze na vlastní zkušenosti.

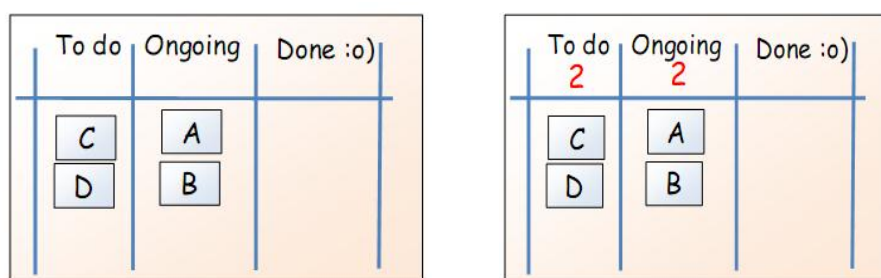
- Scrum ukládá, že bychom měli mít týmy, které mají pracovníky všech funkcí. Takže kdo by měl být v jakém týmu? Nevíme, experiment
- Scrum říká, že týmy by se měly rozhodnout, kolik práce si vezmou do jednotlivých sprintů. Jaké je optimum? Experiment
- Kanban ukládá, že bychom měli limitovat WIP, ale jak velký limit by to měl být, zjistíme až na základě zkušeností

Řekněme, že zredukujeme WIP limit na základě hypotézy, že tato změna vylepší náš proces. Poté sledujeme proměnné jako kapacita, dodací lhůta, kvalita a předvídatelnost změny. Vyvodíme závěr z výsledků a něco málo zase změníme, takhle neustále náš proces vylepšujeme. K tomu slouží řada metodik jako např. Kaizen, Empirické řízení procesu nebo Kontrola & Adaptace. Nejdůležitějším elementem jak pak zpětná vazba. Změna → zjištění jak to šlo → poučení. Všeobecně chceme tyto zpětné vazby v co nejkratší době. U metodiky Scrum, je tato vazba v podobě jednotlivých sprintů. (Kniberg, 2009)

Dle (Kniberg, 2009) u Kanbanu můžeme měřit tyto proměnné:

- Průměrnou dodací lhůtu aktualizovanou pokaždé, když naše položka dojde do stavu „*DONE*“
- Překážky. Typickým symptomem je, že sloupec X je přeplněn položkami, zatímco sloupec X+1 je prázdný. Hledej hluchá místa

Scrum je nepřizpůsobivý změnám během iterace. Vycházejme z následujícího obrázku. Viz Obrázek 7.

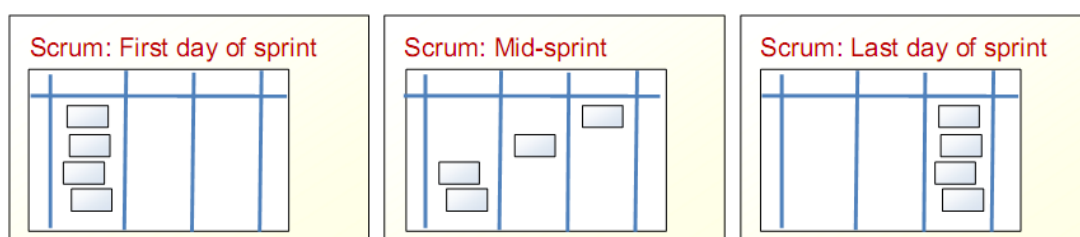


Obrázek 7 – Scrum a Kanban

(Kniberg, 2009)

Pokud budeme chtít přidat do průběhu procesu položku *E*. Scrum team nám řekne, že odsouhlasili „ $A+B+C+B$ “ pro tento sprint. Pokud na této položce budeme trvat, můžeme jí vložit do procesu v dalším sprintu. Což však bude trvat nějaký čas. Viz. levý obrázek U Kanbanu můžeme říci, že není problém vložit položku *E* ale limit ve sloupcích je 2, takže budeme muset odstranit položky *C* nebo *D* v tomto případě. Takže výsledný čas záleží na volných kapacitách v Kanban týmu, zatímco u metodiky Scrum je délka odezvy rovna v průměru polovině délky sprintu. (Kniberg, 2009)

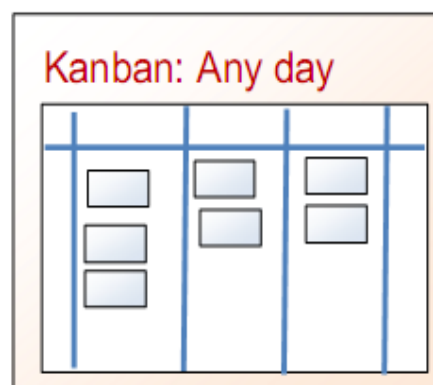
Scrum tabule se smaže po skončení každé iterace. Během iterace vypadá Scrum tabule jako následující obrázek. Obrázek 8 - Průběh metodiky Scrum.



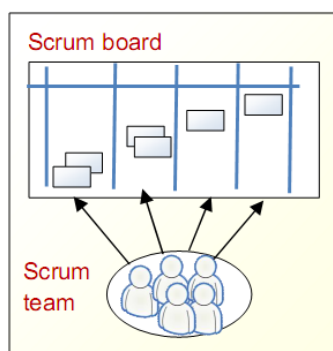
Obrázek 8 - Průběh metodiky Scrum

(Kniberg, 2009)

Když sprint skončí, tabule je prázdná, všechny položky se smažou. Zatímco u metodiky Kanban je tabule za normálních okolností stabilní věc – nepotřebujeme jí resetovat nebo začínat s ní znovu. (Kniberg, 2009) Viz. Obrázek 9 - Průběh Kanban



Obrázek 9 - Průběh Kanban

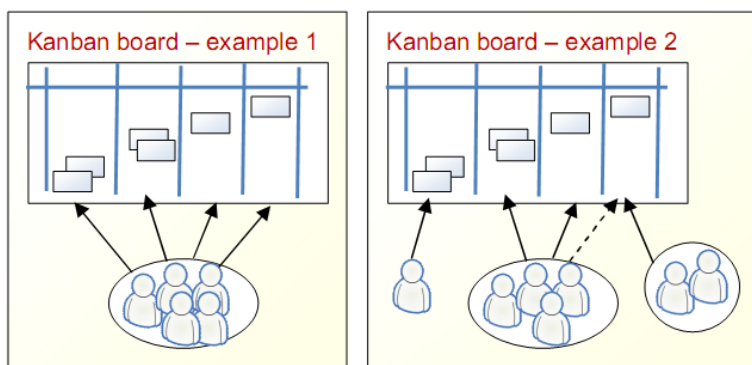


Obrázek 10 - Scrum tým

(Kniberg, 2009)

Scrum tabule je vlastněna jedním týmem, který zahrnuje všechny potřebné funkce, které vedou ke splnění celé iterace. Tato tabule je viditelná pro všechny, ale pouze vlastníci jí mohou editovat. Kanban má nepovinné multifunkční týmy a tabule nemusí být vlastněna specifickým týmem. Tabule je ve vztahu k jednomu pracovnímu toku, nikoliv k týmu. Viz Obrázek 11 - Kanban tým

(Kniberg, 2009)



Obrázek 11 - Kanban tým

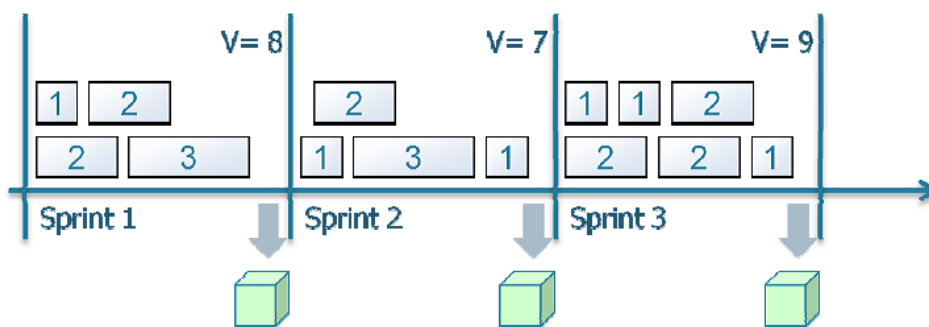
Zdroj: (Kniberg, 2009)

V příkladu 1, je to stejné jako u metodiky Scrum. V příkladu 2 vlastník produktu nastaví priority ve sloupci 1, multifunkční tým vyvíjí software a testuje jej. Závěrečné testování

je úkolem speciálního týmu. Může docházet i k vzájemné výpomoci. U Kanbanu je tedy důležité nastavit základní pravidla pro užití tabule a poté s těmito pravidly experimentovat k optimalizaci pracovního toku. (Kniberg, 2009)

Scrum musí splnit všechny položky v rámci sprintu. Během sprintu musí všechny položky dojít do sloupce „DONE“. Proto je důležité i plánování sprintů, je potřeba si je rozdělit na malé celky, tak abychom všechno zvládli. Kanban tým se snaží minimalizovat dodací lhůtu a úroveň toku, tak tedy nepřímo vytváří pobídku k rozdělení položek do relativně malých kusů. Ale není zde explicitně dané pravidlo určující, že položky musí být malé vzhledem ke splnění nějakého časového horizontu. Na stejné přímce můžeme mít položku, která má k dokončení 1 měsíc anebo jinou která má pouze 1 den. (Kniberg, 2009)

Scrum předepisuje odhad a rychlost. Scrum týmy se snaží zjistit náročnost jednotlivých položek. Součtem jednotlivých položek dokončených na konci sprintu dostaneme rychlost. Zjistíme, kolik věcí můžeme dodat na jeden sprint. Viz Obrázek 12 - Odhad času u Scrum

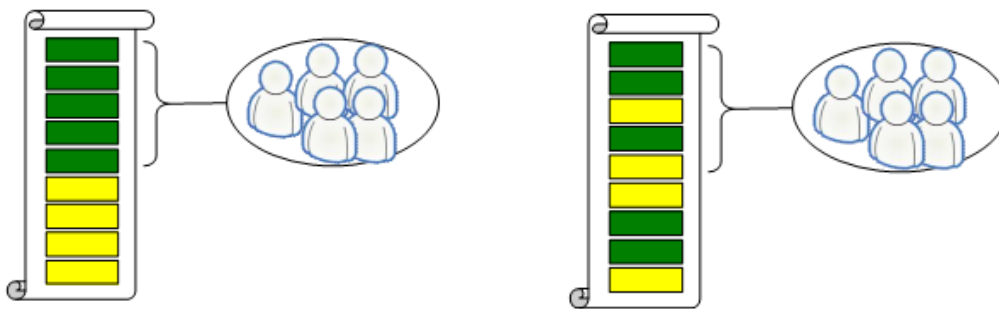


Obrázek 12 - Odhad času u Scrum

Zdroj: (Kniberg, 2009)

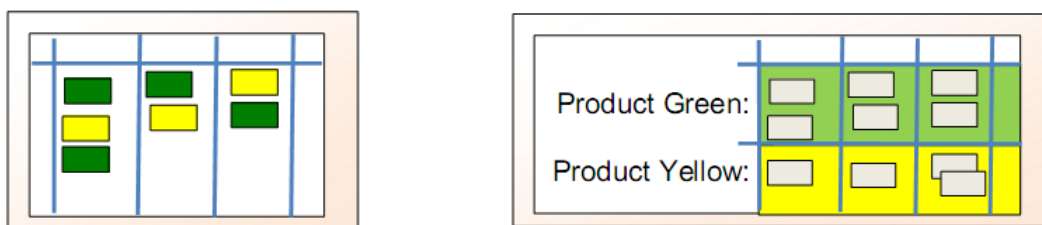
Znalost, že průměrná rychlost je 8 je pěkná, neboť můžeme realisticky předpovídat rychlost položek následujícího sprintu. V metodice Kanban nejsou žádné odhady předepsány. Takže pokud je potřeba nějaká predikce, musíme se rozhodnout, jak k ní dospějeme. Některé týmy k tomu využívají Scrum. Některé odmítají odhad, ale snaží se rozdělit každou položku na části zhruba stejně složité. Poté mohou měřit rychlost jednoduše, z hlediska toho, kolik bylo položek dokončeno za jednotku času. Je zde několik technik, které se k tomuto účelu mohou použít, ale do budoucna jich bude ještě více. (Kniberg, 2009)

Jednou z posledních vlastností je, že obě metodiky umožňují pracovat na více produktech zároveň. U metodiky Scrum můžeme použít 2 strategie při řešení 2 produktů. Viz. Obrázek 13 - Scrum práce na 2 produktech. Řešení pro Kanban, viz. Obrázek 14 - Kanban práce 2 produktech



Obrázek 13 - Scrum práce na 2 produktech

(Kniberg, 2009)



Obrázek 14 - Kanban práce 2 produktech

(Kniberg, 2009)

5 Závěr

Cílem této práce bylo představit metodiku Kanban pro vývoj softwaru. Vznik této metodiky a její vliv na průmysl 20. Století byl popsán ve 2.kapitole. V následující kapitole 3 byly popsány základní prvky, na kterých metodika Kanban stojí. Stěžejní část, této práce se věnovala porovnání metodiky Kanban s další agilní metodikou Scrum. Tato stěžejní část práce je popsána v kapitole 4.

Tato práce si dala jako hlavní úkol popsat rozdíly mezi metodikami Kanban a Scrum. Obě tyto agilní metodiky se však v mnoha ohledech shodují, neodporují si. Vývojové softwarové týmy, by se tedy měly nad podstatou těchto metodik zamyslet a využívat je společně tak, aby docházelo k co nejlepším výsledkům při vývoji softwaru. Vzájemné ideální doplnění metodiky Scrum a Kanban, by se tak mohlo stát dalším tématem pro jinou seminární práci.

Z porovnání obou metodik vyplývá, že Kanban je volnější metodika než Scrum. Má pouze tři hlavní omezení, zatímco Scrum devět. Obě tyto metodiky jsou ale založeny na principu kontinuální a empirické procesní optimalizace, která koresponduje s metodou Kaizen „*štíhlého*“ vývoje softwaru. U obou metodik je tedy nesmírně důležité se poučit z minulého vývoje, a tuto zkušenost aplikovat na vývoj budoucí. Jedině tak můžeme dosáhnout kvalitního fungujícího softwaru, který vyhotovíme včas a za minimální náklady. Tyto slova potvrzuje i (Kniberg, 2009):

„Experimentujme, dokud nenajdeme něco, co pro nás funguje. A poté pokračujme v experimentování.“

6 Literatura

Anderson, David J. 2010. *Kanban - Successful Evolutionary Change for your Technology Business.* místo neznámé : Blue Hole Press, 2010. 978-0984521401.

Dubakov, Michael. 2008. Kanban in Software Development. *targetprocess.com.* [Online] 8. 9 2008. <http://www.targetprocess.com/blog/2008/09/kanban-in-software-development.html>.

Hiranabe, Kenji. 2008. Kanban Applied to Software Development: from Agile to Lean. *Info Q development.* [Online] 14. 1 2008. <http://www.infoq.com/articles/hiranabe-lean-agile-kanban>.

John M. Gross, Kenneth R. McInnis. 2003. *Kanban made simple: demystifying and applying Toyota's legendary manufacturing process.* New Yorke : AMACOM Div American Mgmt Assn, 2003.

Joyce, David. 2009. Kanban for software engineering. *wordpress.com.* [Online] 24. 05 2009. <http://leanandkanban.files.wordpress.com/2009/04/kanban-for-software-engineering-apr-242.pdf>.

Kniberg, Henrik. 2009. Kanban vs Scrum. *crisp.se.* [Online] 2009. <http://www.crisp.se/henrik.kniberg/Kanban-vs-Scrum.pdf>.

Tuček, David. 2004. systému, Kanban jako řídicí a integrující metoda v informačním. *cvis.cz.* [Online] 04. 10 2004. <http://www.cvis.cz/hlavni.php?stranka=novinky/clanek.php&id=167>.

Seznam obrázků

Obrázek 1 - Kanban	6
Obrázek 2 - Pravidla Scrum	9
Obrázek 3 - Pravidla Kanban	10
Obrázek 4 - Počet omezení jednotlivých metodik	11
Obrázek 5 - Porovnání WIP mezi Kanban a Scrum.....	12
Obrázek 6 - Měřené proměnné.....	12
Obrázek 7 – Scrum a Kanban.....	13
Obrázek 8 - Průběh metodiky Scrum	14
Obrázek 9 - Průběh Kanban	14
Obrázek 10 - Scrum tým	14
Obrázek 11 - Kanban tým	15
Obrázek 12 - Odhad času u Scrum.....	15
Obrázek 13 - Scrum práce na 2 produktech.....	16
Obrázek 14 - Kanban práce 2 produktech.....	16