

Vysoká škola ekonomická v Praze



# Kanban a jeho využití při vývoji SW

Semestrální práce pro předmět 4IT421 - Zlepšování procesů budování IS

**Bc. Zdeněk Filipčík**

## Obsah

Úvod .....	2
1. Kanban obecně .....	2
1.1. Popis .....	2
2. Kanban v oblasti vývoje softwaru.....	4
2.1. Důvody pro zavedení.....	4
2.2. Stanovení limitů pro práci .....	6
2.3. Rozdrobování činností .....	8
2.4. „Pull“ systém .....	8
3. Jak začít s Kanbanem .....	10
3.1. Základní kroky.....	10
3.1.1. Definujte a zobrazte vaše procesy.....	10
3.1.2. Nastavte pracovní v procesu (WIP) .....	10
3.1.3. Používejte metodu „Push“, ne „Pull“ .....	11
4. Závěr .....	11
Použité zdroje.....	12
Příloha: Ukázka reálných Kanban boards .....	14

## Úvod

Kanban se stal jednou z inspirativních technik, které IT svět dlouhou dobu opomíjel. Nyní si z ní přebírá to nejlepší a snaží se jí uplatnit i při vývoji softwaru. Dnešní snaha o Agilní vývoj, orientace na zákazníka, omezení rozpracovanosti, v důsledku velké konkurence omezování nákladů a mnohé další, dalo za vznik specializované verze Kanbanu pro vývoj softwaru. O této konkrétní verzi bude pojednávat tato seminární práce, ve které bych rád z důvodu lepšího pochopení celé koncepce představil i původní verzi Kanbanu. Spolu s tím se budu snažit představit i rozdíly v Kanbanu používaném ve výrobních a vývojových podnicích. Dále bych se chtěl věnovat již hlavním myšlenkám a postupům používaným systémem Kanban při vývoji softwaru. V závěru práce bych se rád dotknul tématu jak začít s tímto druhem Kanbanu. Nastínit hlavní předpoklady pro implementaci a základní kroky při jeho zavádění a používání. V této práci využívám především internetových zdrojů v podobě odborných článků a zveřejněných akademických prací.

## 1. Kanban obecně

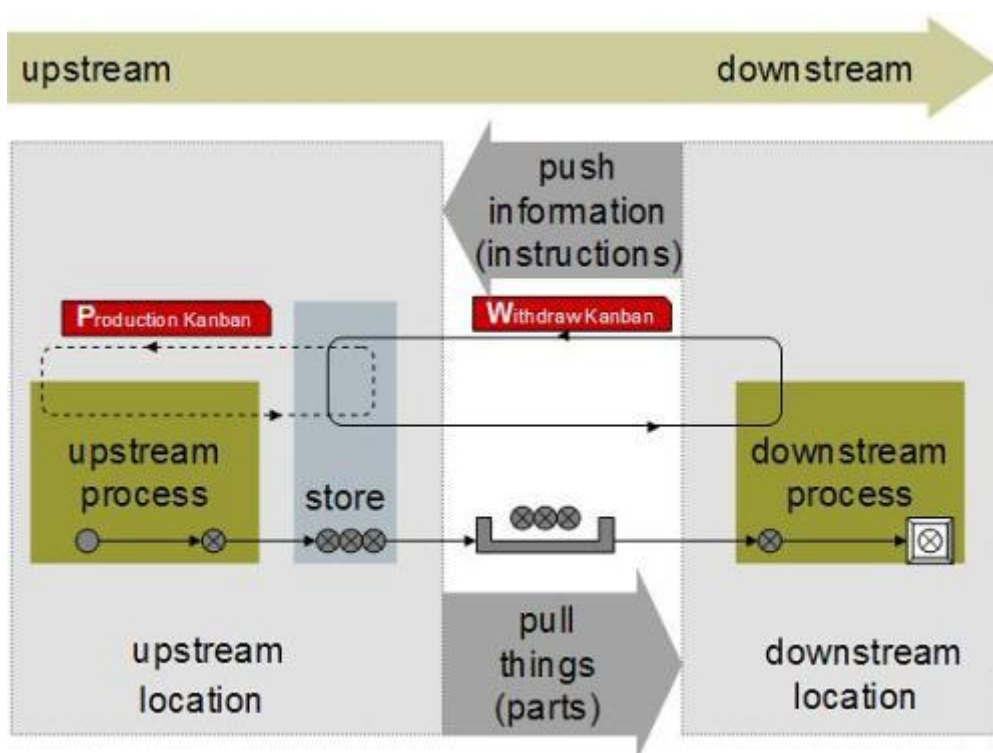
V pozdních čtyřicátých letech dvacátého století začala automobilka Toyota studovat supermarket a jejich úložnou plochu, systém a dobu skladování. Zabývala se především myšlenkou toho, aby zákazníci dostali to, co chtějí a ve správný okamžik a množství, které oni vyžadují. Konkrétně viděla to, že supermarket a naskladňují jen to, co předpokládají, že zákazníci budou potřebovat a tím si zajistí nepřetržitou dodávku zboží. V tomto viděla firma Toyota budoucnost, a snažila se tyto postupy aplikovat do vnitřních procesů firmy. Podobně jako v supermarketech byly vývěsní štítky, dle kterých se zboží opět naskladňovalo do regálu, používá Kanban karty dle kterých se řídí nejenom tok materiálu ale i práce. Toyota tento systém poprvé zavedla již v roce 1953 jako součást Toyota Production System známý pod zkratkou TPS (Taiichi, 1988).

### 1.1. Popis

Slovo Kanban vzniklo spojením „Kan“, volně přeloženo jako signál a „ban“, což znamená karta nebo přímo tabule (Patton, 2009). Volný překlad tedy může znamenat vizuální tabule či karty. A na těchto dvou prvcích Kanban staví. Jde především o vizualizaci výroby pomocí takzvaných Kanban karet a tabulí. Je to jednoduchá technologie koordinující pohyb materiálu a práce při využití vizuálních prvků, usnadňující řízení plánování výroby a řízení zásob. Systém klasického Kanbanu vidíte na obrázku 1. Cílem Kanbanu je minimalizovat WIP (Work-in-proces) nebo zásob mezi procesy tím, že zajišťuje informace o požadovaných dílech, které má proces proti proudu výroby vyrobit.

Systém Kanban pracuje na několika principech (Šifra, 2011):

- Fungují zde tzv. samořídící regulační okruhy, které tvoří dvojice článků (dodavatel a odběratel) vzájemně propojené na základě tzv. „pull principu“ (tažného principu).
- Objednací množství je zde obsah jednoho nebo více přepravních prostředků, které jsou plně naloženy stále stejným množstvím materiálu.
- Dodavatel zde ručí za kvalitu a odběratel má povinnost svoji objednávku vždy převzít.
- Kapacity odběratele a dodavatele jsou vyvážené a jejich činnosti jsou synchronní.
- Spotřeba materiálu je rovnoměrná bez velkých výkyvů a sortimentních změn.
- Dodavatel ani odběratel nevytváří žádné zásoby.



Obrázek 1: Procesy umístěné níže ve výrobním procesu táhnou poptávku po meziproduktech z výše položených procesů. (Hiranabe, 2008)

Nyní si na obrázku 1 ukažme na názorném příkladu jak tedy Kanban funguje. Představte si sami sebe ve výrobním závodě Toyota při montáži například palubních desek nového Avensisu. Máte k dispozici 12 palubních desek, které postupně ubývají, jak jsou montovány na auta přijíždějící po lince. Pro zajištění plynulé výroby je u 6 desky přiložená Kanban karta s požadavkem pro zkompletování dalších 12 palubních desek, kterou odevzdáte na Vaše sběrné místo Kanban kartiček a můžete se vrátit zpět ke své práci. Během toho si kompetentní pracovník vyzvedne váš požadavek a započte kompletování dalších palubních desek, přitom opět používá svoje karty pro dodávky komponent potřebných ke kompletaci a takto proces pokračuje dál až k výrobním linkám jednotlivých

částí palubních desek. Takže v našem příkladu je kompletace desek upstream proces a jejich montáž do vozidel downstream proces. Vše je v ideálním případě nastaveno tak, aby při dokončování montáže vaší poslední palubní desky přijel pracovník s dodávkou dalších 12 palubních desek a výroba tak mohla nepřetržitě pokračovat dál. Pokud je tento systém správně nastaven, značně se sníží rozpracovanost a omezí přebytky ve výrobě, které by se tímto musely skladovat, což ve štíhlé výrobě označuje jako odpad. Dále takto stavěný proces umožňuje pružně reagovat na vznik problémů. Pokud by například nastal problém s robotem, který asistuje při montáži palubních desek, nikdy nemůže vzniknout to, že by se na pracovišti hromadilo více než 18 desek, protože by se do oběhu nedostaly karty s požadavkem na jejich další kompletaci. Pracovníci starající se o kompletaci desek se tímto nemusí zabývat výrobou aktuálně nepotřebných desek a můžou se věnovat například požadavkům z jiných stanovišť či urychlit opravu robota a tím snížit prostoje ve výrobě. Dá se říct že v tomto případě zapůsobily Kanbanové kartičky jako měna. Podobně jako existuje pouze fixní množství peněz v systému, existuje pouze fixní efektivní množství zásob, které mohou být tímto omezeny. Systém Kanban karet také umožňuje pružně reagovat na zlepšení jedné části výrobního procesu. Pokud se například inovacemi výrazně zrychlí kompletace palubních desek, avšak nevznikne jejich požadavek na jejich větší dodávky, je možné, aby si toto stanoviště vzalo i jiné karty například na kompletaci středových panelů řadicí páky a tyto panely kompletovalo, pokud na to existuje požadavek. Tím se omezí prostoje ve výrobě, a uvedou v praxi všechny kladné efekty uvedené v definici.

V průmyslu je Kanban jako součást Lean Manufacturing již poměrně rozšířený. Takže se nejedná o žádnou novinku, spíše je zajímavé, že trvalo více než 30 let, než si svět informačních technologií všimnul, že v průmyslu mají výborný a inspirativní nástroj, který lze aplikovat i na oblast vývoje softwaru (Vallo, 2010). Ale o tom již následující kapitola.

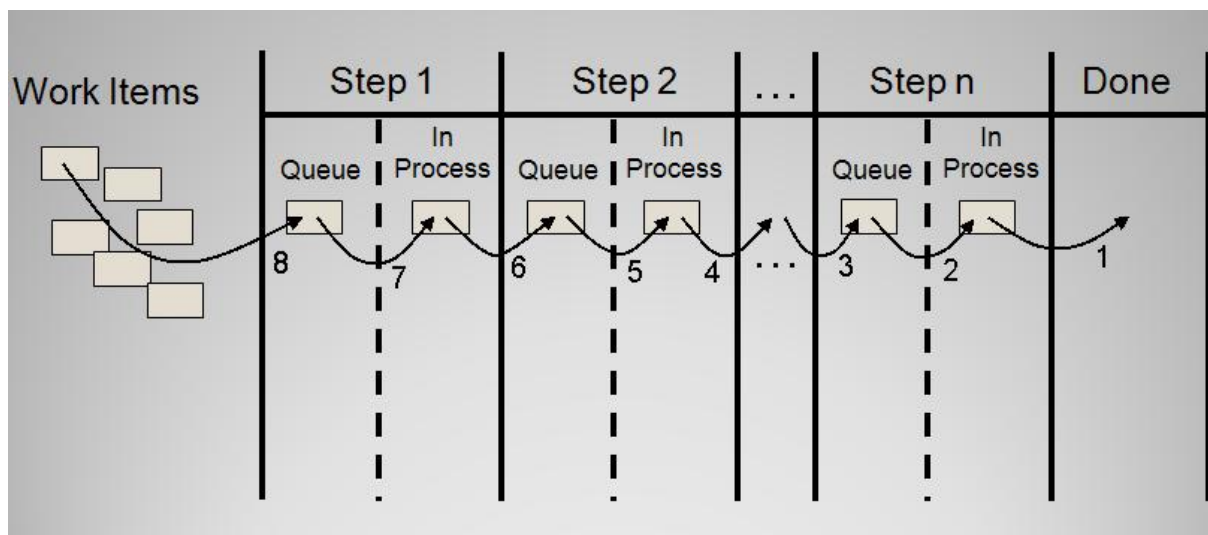
## 2. Kanban v oblasti vývoje softwaru

Vývoj softwaru a výrobní činnost není to samé. Z tohoto vyplývají i rozdíly mezi Kanbanem výrobním a vývojovým. Softwaroví inženýři vytváří pokaždé jiné věci, zatím co při výrobě produkujeme stejné věci pokaždé dokola. Marně jsem se snažil najít nějakou přesnou definici Kanbanu pro IT avšak neuspěl jsem. Časem mi došlo, že Kanban pro IT vlastně ani definici mít nemůže, protože přebírá jen určité techniky jak organizovat práci v týmech a odproštuje se od materiálových požadavků a soustředí se především na vizualizaci průběhu procesu.

### 2.1. Důvody pro zavedení

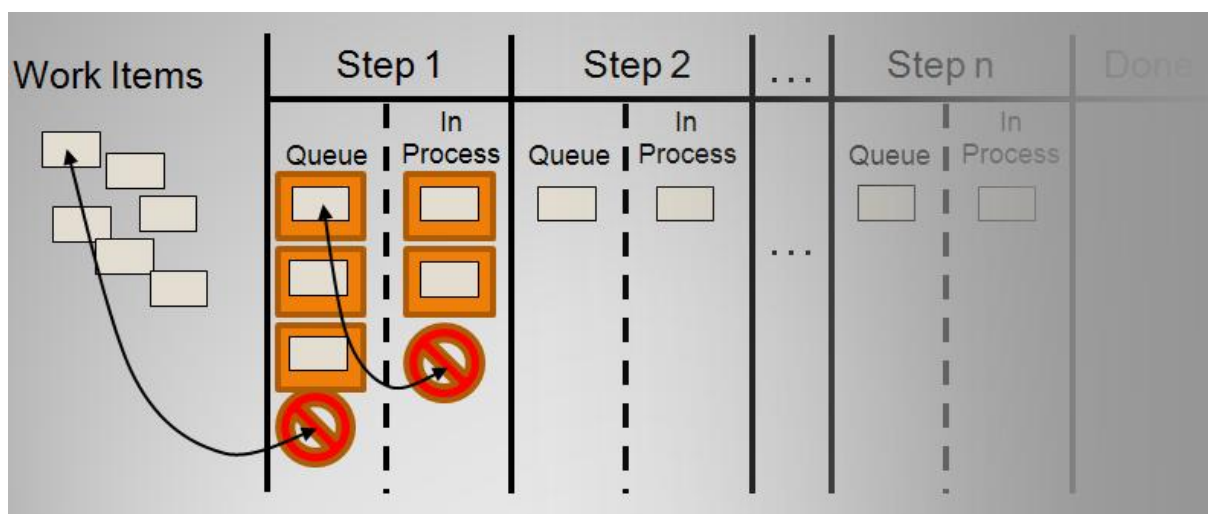
Mnohé z vás by mohla napadnout otázka, proč tedy vlastně zavádíme systém Kanban do vývoje softwaru, když zde neoperujeme se zásobami? Na to je jednoduchá odpověď. Představte si výrobu auta a všechny jeho procesy. Většina dílů jako například brzdové kotouče, skla, budíky a mnohé další musí být nejdříve vyrobeny, samostatně alespoň hrubě otestovány (jak kvalita, tak například i funkčnost) a poté složeny do větších celků, polotovarů a zkompletovány do podoby osobního automobilu. A to se opakuje stále dokola. V případě vývoje softwaru se to dá analogicky

převést. Každá jednotlivá funkcionálita výsledného programu musí být nejdříve zanalyzována, navrhována. Po těchto krocích následuje samotné naprogramování, testování a implementace do většího celku. V obou dvou případech lze jednotlivé úkony generalizovat a řetězce rozdrobit na jednotlivé činnosti. Tyto činnosti jako například vytvoření GUI, možnost importu v XLS či jiné představují jednotlivé Kanban karty a generalizované fáze mohou být například analýza a návrh, vývoj, testování, implementace, což představují sloupce v Kanban nástěnce, na které se následně zobrazuje fronta práce. Jednotlivé činnosti pak procházejí fázemi, až jsou hotovy (Scotland, 2008). Tento systém je možné vidět na obrázku 2.



Obrázek 2: Task Board (Scotland, 2008)

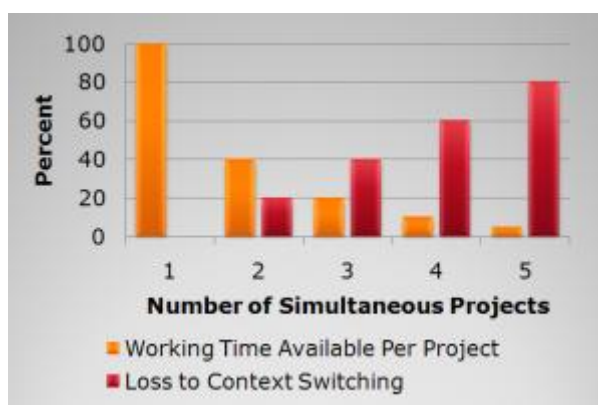
Někomu se může zdát, že tato tabulka je velmi podobná typické agilní tabulce úkolů (Agile Task Board), která obrazuje všechny úkoly pro každou iteraci. Z části je to pravda, avšak je tu však jeden velmi důležitý prvek, který v obrázku 2 chybí. Tím jsou limity. Na limitech staví celý systém Kanban. Obrázek 3 nám již ukazuje pravou Kanban Board s nastavenými limity, o nichž si více povíme v následující podkapitole.



Obrázek 3: Kanban Board (Scotland, 2008)

## 2.2. Stanovení limitů pro práci

Jedním z důležitých parametrů při použití metody Kanban je stanovení limitu práce. Existují dva základní limity, limit ve frontě čekající na zpracování a limit „work in progress“. Limity fronty čekající na zpracování jsou navrženy tak, aby zabránily předčasnému započetí práce ale také s ohledem na to, aby celý tým mohl pracovat bez zbytečných prostojů. Správným nastavením tohoto limitu je zajištěno dokončení všech úkolů právě včas a byl primárně uplatněn princip just-in-time (JIT). V ideálním případě by se na tuto frontu měl uplatnit princip FIFO, i když to spíše může být vodítko, než pravidlo. Záleží totiž na daných okolnostech a dovednostech volných pracovníků, tím je uplatněn i měkký princip řízení, kdy se sám pracovník rozhodne, který z dostupných úkolů ve frontě si vybere s ohledem na další zdroje, či jeho zkušenosti a dovednosti. Limity ve frontě na zpracování nám pomáhají k uplatnění principu JIT. Ruku v ruce s tím jsou nastaveny i limity WIP snažící o maximální redukci multitaskingu, či chcete-li rozpracovanosti. Rozpracovanost má totiž za následek znatelné snížení reálného času věnovanému danému projektu. Největší časové ztráty vznikají při přecházení mezi projekty, uvedení se do problematiky a zapojení se aktivně do práce. Obrázek 4 názorně ukazuje závislost počtu simultánně probíhajících projektů na práci na nich skutečně vykonané.<sup>1</sup>

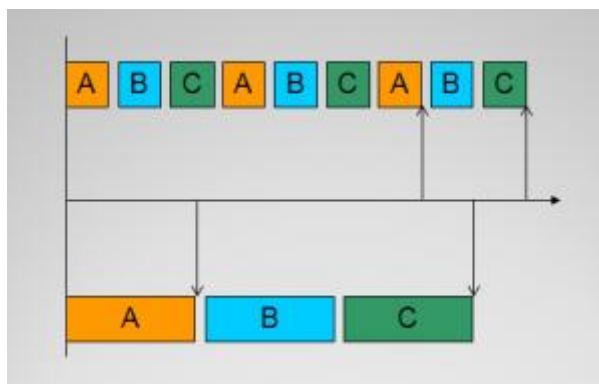


Obrázek 4: Ztráty způsobené rozmělnění práce na více projektech. (Scotland, 2008)

Množství času skutečně věnovanému projektu se při multitaskingu snižuje, a to velmi výrazně. Už při práci na dvou projektech je skutečně vykonaná práce na jeden projekt ne 50 ale přibližně 40 procent. Při pěti projektech už značnou část pracovní doby pracovník věnuje přeskakováním mezi projekty. Proto se systém Kanban snaží o co největší eliminaci multitaskingu. Další věc související s multitaskingem je fakt, že při multitaskingu se nejen prodlouží doba potřebná na vyřešení daných úloh, či prací na projektech (dle mého názoru to platí obecně), ale i všechny úlohy budou vyřešeny daleko později než při jejich postupném zpracování. Obrázek 5 ukazuje zpracování úloh A, B, C. Horní řádek ukazuje zpracování úloh při použití multitaskingu. Délka všech tří obdélníků u každé úlohy je stejně dlouhá jako u druhého řádku, kde jsou úlohy zpracovávány sekvenčně. Z obrázku 5 tedy jasně vyplývá nejenom delší doba zpracování všech úloh, ale i to, že jsou dokončeny daleko později než při klasickém zpracování. Stejně jako se tomu děje například při čtení dat

<sup>1</sup> Zdroj: Gerald Weinberg, Quality Software Management: Systems Thinking, 1991, 978-0932633224

z klasického harddisku, kde při velké fragmentaci trvá načítání souboru daleko déle než při sekvenčním čtení.



Obrázek 5: Sekvenční vs. multitaskové zpracování úloh

Přílišné snižování WIP se však může jevit nevhodné pro velké týmy. Často panují obavy, že při nízko nastavené WIP budou někteří členové týmu nevytíženi a nebudou mít co na práci, protože jejich upstream proces nebude připraven. Na to však má Kanban jednoduchou příručku pro případ, že nemáte připravenou žádnou práci (Scotland, 2008):

1. **Můžete pomoci při řešení nějakého již existujícího úkolu?**  
*Pracuje na tom.*
2. **Nemáte na to správné vědomosti?**  
*Pokuste na Kanban Board se najít úzké místo a vyřešte ho.*
3. **Nemáte na to správné vědomosti?**  
*Pomozte při řešení upstream procesu.*
4. **Nemůžete začít řešit žádnou práci z upstream procesu?**  
*Jsou zde nějaké úlohy s nižší prioritou, které se mohou začít řešit?*
5. **Není tu nic s nižší prioritou?**  
*Najděte si jinou zajímavou práci.*

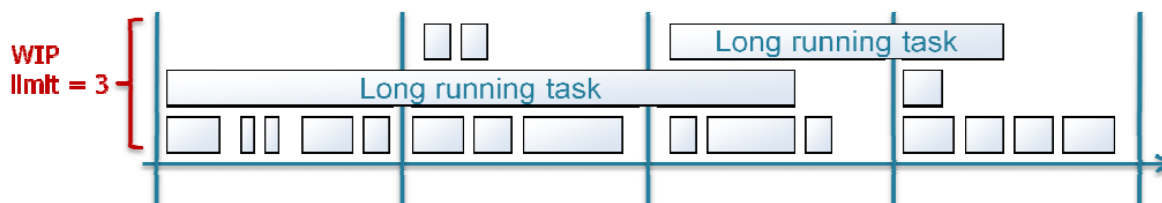
Základní otázka u této příručky je i co přesně znamená práce s nižší prioritou a jiná zajímavá práce. Práce s nižší prioritou je taková, který nevytváří žádné downstream úlohy a spíše pomůže zlepšit propustnost celého procesu vývoje nebo může být přerušena, jakmile bude další úloha (Kanban karta) k dispozici. Může to být také analýza pro jinou důležitější úlohu. Jiná zajímavější práce se být například i osobní vývoj nějaké části systému či jeho inovace. Všechny tyto kroky dále podporují práci v týmu a zlepšují řešení ať již úzkých míst, tak práci na složitějších a náročnějších úlohách. Je zde vyloučeno, aby pracovník odpovědný například za GUI si předem vypracoval kompletní grafické rozhraní, aniž by byly hotovy věci související s funkcionalitou, které v konečné míře změni požadavky na grafické rozhraní, které by se tímto muselo dodělovat. Místo toho tento pracovník pomáhat řešit počáteční úlohy a tímto urychlit celý proces. Naopak na konci vývoje již budou někteří programátoři zabývající se zpočátku funkcionalitou volní pro práci i na GUI a tím opět dojde k urychlení práce na celém projektu (Scotland, 2008).



## 2.3. Rozdrobování činností

Rozdrobování činností je další důležitou součástí jak agilního myšlení, tak i Kanbanu. Toto dělení nám slouží především k lepší správě, kontrole, zjištění úzkých míst. Je důležité si uvědomit, že v oblasti softwarového vývoje je daleko více uplatněno to, že primární cíle jsou dosahovány lidmi a nikoliv stroji. To klade daleko větší požadavky na jejich řízení. Systém Kanban umožňuje pružně reagovat na rozdílnou rychlost práce jednotlivých pracovníků, zohledňovat jejich zkušenosti. Pokud se tedy práce na jednom s upstreamů zpozdí či začne komplikovat, mohou pracovníci z downstreamů při dořešení svých úkolů pomoci při tohoto problému. Zaměstnanci, kteří již dokončili svoji práci si taky mohou z volných Kanbanových karet vybírat, ty, na které mají větší specializaci a tak se pružně a nejvhodněji začlenit do workflow.

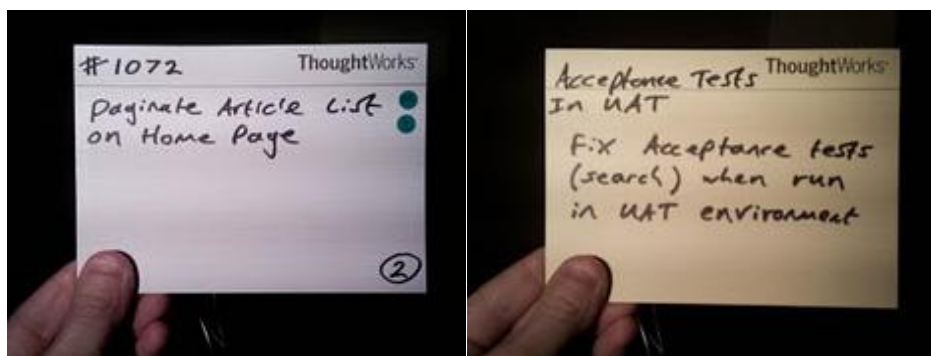
Při rozdělování na konkrétní úlohy, nebo chcete-li Kanban karty, je nutné správně určit, jakou úroveň podrobnosti si zvolíme. Na rozdíl od Scrumu či jiných, kde jsme omezeni neustále se opakujícími meetingy či eventy, je Kanban v tomto ohledu daleko volnější a umožňuje vedle sebe mít jak krátkodobé i dlouhodobé trvající úlohy bez potřeby je jakkoliv přerušovat kontrolami či meetingem. Tento fakt znázorňuje obrázek 6.



Obrázek 6: Různě velké položky na Kanban Board (Kniberg, 2009)

## 2.4. „Pull“ systém

Pull (tažený) systém nám oproti push (tlačený) způsobu práce umožňuje soustředit se pouze na úlohy, které jsou aktuálně třeba. Tento princip nám umožňuje předcházet tomu, abychom vyvíjeli části kódu, které nakonec nebudou potřeba, nepsali více specifikací a dokumentace než je třeba, nepsali více kódu, než budeme testovat a netestovali více kódu, který nakonec nebude nasazen. Princip pull systému je založen na signálech, které nám vytvářejí Kanban karty. Ty oproti jejich ekvivalentům z výrobních podniků obsahují daleko méně informací a již z podstaty velké odlišnosti těchto dvou činností vyplývá, že tyto karty budou velmi odlišné. Na obrázcích 7 a 8 můžete porovnat odlišnosti jednotlivých Kanban karet. Kanban z oblasti vývoje však velmi často používá karty barevně odlišené. Jednotlivé barvy většinou znamenají rozdílný typ akce, které se tato karta týká. Jejich barevné značení závisí pouze na kreativitě implementátora Kanban systému. Většinou se však pro klasické úlohy používají žluté nebo bílé karty, modré pro analýzu, a červené pro nějaký chybový stav či závadu. Je zde důležité poznamenat, že design karet není jednotný a záleží jen na jeho tvůrci.



Obrázek 7: Kanban karty používané při vývoji SW. (Stanger, 2011)

Z obou obrázků je také patrné, že u karet používaných pro vývoj SW má být většinou hned jasné, jaký úkol obnášejí a obsahují co nejvíce informací sloužící k jeho jednoznačnému určení jako je například zadání, počet lidí který na daném úkolu má pracovat či jak dlouho už se na daném úkolu pracuje (to je například znázorněno pomocí zelených teček na levé kartě v obrázku 7). Podíváme-li se na obrázek 8, není nám zcela jasné k čemu tato Kanban karta slouží. Zde se již bez čtečky čárových kódů již neobejdeme. Je to především díky skutečnosti, že na výrobních linkách již většinou pracovníci přesně uvědí, co dané karty znamenají a automaticky je odevzdávají do správných boxů, kde si je pak vyzvedne kompetentní osoba s čtečkou a načte si potřebná data ať již pro dodavatel například dílu nebo pro odeslání signálu do informačního systému.



Obrázek 8: Kanban Karta z výrobního podniku.

## 3. Jak začít s Kanbanem

Tato kapitola by měla poskytnout důležité informace pro to, jak začít s Kanbanem v oblasti software development. V ní se inspiřuji článkem „How To Get Started With Kanban In Software Development“ od Dericka Baileyho (Bailey, 2009). Autor tento článek několikrát upravoval a promítnul do něj své několikaleté zkušenosti se systémem Kanban. Zavádění Kanbanu je nenáročné, nákup nástěnky a vytváření kartiček je velice levné, oproti tomu změny v řízení mohou být značné.

### 3.1. Základní kroky

Předem je nutné říci, že tyto kroky vůbec nejsou nějaký návod krok za krokem, ale spíše „best practices“ nebo kolekce pravidel jak nejlépe začít. Je nutné brát ohledy na odlišnosti jednotlivých projektů a kulturu v podniku. Nyní již pojďme k následujícím třem základním krokům:

1. Definujte a zobrazte vaše procesy.
2. Nastavte pracovní v procesu (WIP).
3. Používejte metodu „Push“, ne „Pull“.

#### 3.1.1. Definujte a zobrazte vaše procesy

Základním úspěchem pro používání Kanbanu je vizualizace procesů a vytvoření správné Kanban tabule. Je zde mnoho zdrojů, které je pro tento krok použít. Diagram stavů v UML, workflow diagram. Další možností je tvořit tabulku přímo s konkrétním týmem. Má to své výhody, na rozdíl od použití již hotových diagramů může tento postup vytvořit aktuálnější nástěnku, která bude více respektovat aktuální situaci než pouhé dedukování z již publikovaných zdrojů. Je zde velmi důležitá poznámka: **Při používání Kanbanu nemusíte měnit vaše procesy.** Vše se odehrává pouze na úrovni řízení toků.

#### 3.1.2. Nastavte pracovní v procesu (WIP)

Správné nastavení limitů pro práci je jednou z důležitých předpokladů pro správné fungování Kanbanu. Chceme-li se držet opravdu štíhlých metodik, je nutné snížit aktuální počet všech úloh na Kanban board. Na to existuje jednoduché pravidlo pro stanovení počátečních hodnot. Tyto hodnoty se pak mohou měnit v závislosti na tom, jak se osvědčí. Základní vzorec pro součet všech limitů rozpracovanosti je, že sečteme všechny členy týmu ve všech rolích a vydělíme dvěma. Pokud je tedy všech členů 20, dostaneme číslo 10 a toto číslo pak rozdělujeme mezi jednotlivé sloupce. Pro sloupce „fronta“ (queue) se obecně nastavuje polovina hodnoty oproti sdruženému sloupci „probíhající“ (In process). Pro omezení sloupce „probíhající“ (In process) pro danou činnost je limit často nastavován na polovinu lidí, kteří mohou vykonávat tuto práci. Tím podporujeme spolupráci v týmech. Tyto hodnoty jsou pouze orientační a je nutné vše otestovat (Patton, 2009).

### 3.1.3. Používejte metodu „Push“, ne „Pull“

Toto je jeden z lehčích kroků, pokud již máte nastaveny limity rozpracovanosti. Stačí již jen dodržovat Kanban Board. Z tabulky je vždy jasné vidět, kde a jaká práce se nachází. Dle stavu rozpracovaných úloh je možné předvídat, kde bude v nejbližším časovém úseku potřeba nejvíce pracovní síly a tomu se přizpůsobovat. Pokud naopak nějaká skupina zaměstnanců dlouho čeká na práci, je nutné odhalit slabá místa a předcházet jim. Z tohoto důvodu je nutné neustálé monitorování situace.

## 4. Závěr

Snaha o co největší úspory jak ve výrobě, tak nakonec i ve vývoji softwaru podporovala zájem o co nejefektivnější řízení. Jednou z takových cest se vydala Toyota v roce 1953 zavedením Kanbanu. Překvapivé je, že až o mnoho let později si tohoto nástroje všimly i týmy v oblasti vývoje softwaru. A v pozměněné podobě tento systém začaly používat. O této modifikaci byla především tato seminární práce, kde jsem nejdříve popsal základy původního Kanbanu a to i na názorném příkladu. Kanban v oblasti vývoje softwaru se opírá především o názornou vizualizaci probíhajících prací, jejímž základem je Kanban Board. Dalším klíčem k úspěšnému řízení je především stanovení si maximální rozpracovanosti a snížením multitaskingu, který, jak jsem předvedl na přiložených grafech, dokáže zásadně snížit efektivní čas věnovaný konkrétním úkolům. Protože pokud chceme přinést nové hodnoty pro zákazníka rychle, musíme snížit i množství najednou rozpracované práce. To je možné dosáhnout i aplikováním takzvaného „pull“ systému řízení práce, kdy je daná práce tažena odvedenými úkoly. To nám právě umožňuje systém Kanban karet, které fungují jako signální body určující co se má konkrétně dělat. O tomto pojednává celá druhá kapitola. Ty, které metoda Kanban zaujala, jsem ve třetí kapitole lehce nastínil, jak by mohli začít používat Kanban.

Nakonec je zde nutné připomenout, že vlastně Kanban ani není metodika, je to pouze systém postupů jak určitou věc dělat jinak, efektivněji a je jen na nás co si z něj převezmeme. Systém Kanban ve své podstatě může být jak jednoduchá kartička v hromadě balíků papíru u tiskárny sloužící nám k připomenutí, že máme objednat další balíky, jinak nám v dohledné době dojde papír, tak i velmi propracovaná tabule s důmyslně stanovenými limity a postupy přispívající značně ke snížení rozpracovanosti a zvýšení rychlosti se kterou můžeme reagovat na požadavky zákazníka. Je to jen na nás.

## Použité zdroje

**Bailey, Derick. 2009.** How To Get Started With Kanban In Software Development. *LosTechies.com*. [Online] 5. Srpen 2009. [Citace: 2. Listopad 2011.]

<http://lostechies.com/derickbailey/2009/08/05/how-to-get-started-with-kanban-in-software-development/>.

**Hiranabe, Kenji. 2008.** Kanban Applied to Software Development: from Agile to Lean. *InfoQ.com*.

[Online] 14. Leden 2008. [Citace: 2011. Listopadu 2.] <http://www.infoq.com/articles/hiranabe-lean-agile-kanban>.

**Joyce, David. 2009.** Kanban for Software Engineering. *Systems Thinking, Lean and Kanban*. [Online] 24. Duben 2009. [Citace: 2. Listopad 2011.]

<http://leanandkanban.files.wordpress.com/2009/04/kanban-for-software-engineering-apr-242.pdf>.

**Kelly, Allan. 2009.** 10 things to know about Kanban software development . *Agile, Lean, Patterns* .

[Online] 15. Květen 2009. [Citace: 2. Listopadu 2011.] <http://allankelly.blogspot.com/2009/03/10-things-to-know-about-kanban-software.html>.

**Kniberg, Henrik. 2009.** Kanban vs Scrum: How to make the most of both . *Crisp.se*. [Online] 29.

Červen 2009. [Citace: 4. Prosinec 2011.] <http://www.crisp.se/henrik.kniberg/Kanban-vs-Scrum.pdf>.

**Ladas, Corey. 2007.** Kanban systems for software development. *Lean Software Engineering*. [Online]

29. Srpen 2007. [Citace: 2. Listopad 2011.] <http://leansoftwareengineering.com/2007/08/29/kanban-systems-for-software-development/>.

**Patton, Jeff. 2009.** Kanban Development Oversimplified. *AgileProductDesign.com*. [Online] 20. Duben 2009. [Citace: 2. Listopad 2011.]

[http://agileproductdesign.com/blog/2009/kanban\\_over\\_simplified.html](http://agileproductdesign.com/blog/2009/kanban_over_simplified.html).

**Peter Middleton, David Joyce. 2011.** Agile, Lean, and Kanban, Do They Change Management

Thinking? *Systems Thinking, Lean and Kanban*. [Online] 10. Září 2011. [Citace: 2011. Listopadu 2.] <http://leanandkanban.wordpress.com/>.

**Rally Software Development Corp.** AgileZen. *AgileZen.com*. [Online] [Citace: 2. Listopadu 2011.]

<http://www.agilezen.com/>.

**Scotland, Karl. 2008.** Kanban, Flow a Cadence. *AvailAgility.co.uk*. [Online] 28. Říjen 2008. [Citace: 4.

Prosinec 2010.] <http://availagility.co.uk/2008/10/28/kanban-flow-and-cadence/>.

**Stanger, Damian. 2011.** Kanban inspired card wall. *Folding Air, Software Development With Damian Stanger*. [Online] 20. Říjen 2011. [Citace: 4. Prosinec 2011.]

<http://foldingair.blogspot.com/2011/10/kanban-inspired-card-wall-our-example.html>.

**Šifta, Jan. 2011.** *Skladové hospodářství podniku*. Diplomová práce : Masarykova univerzita v Praze, 2011. [http://is.muni.cz/th/212936/esf\\_m/Diplomova\\_prace.txt](http://is.muni.cz/th/212936/esf_m/Diplomova_prace.txt).

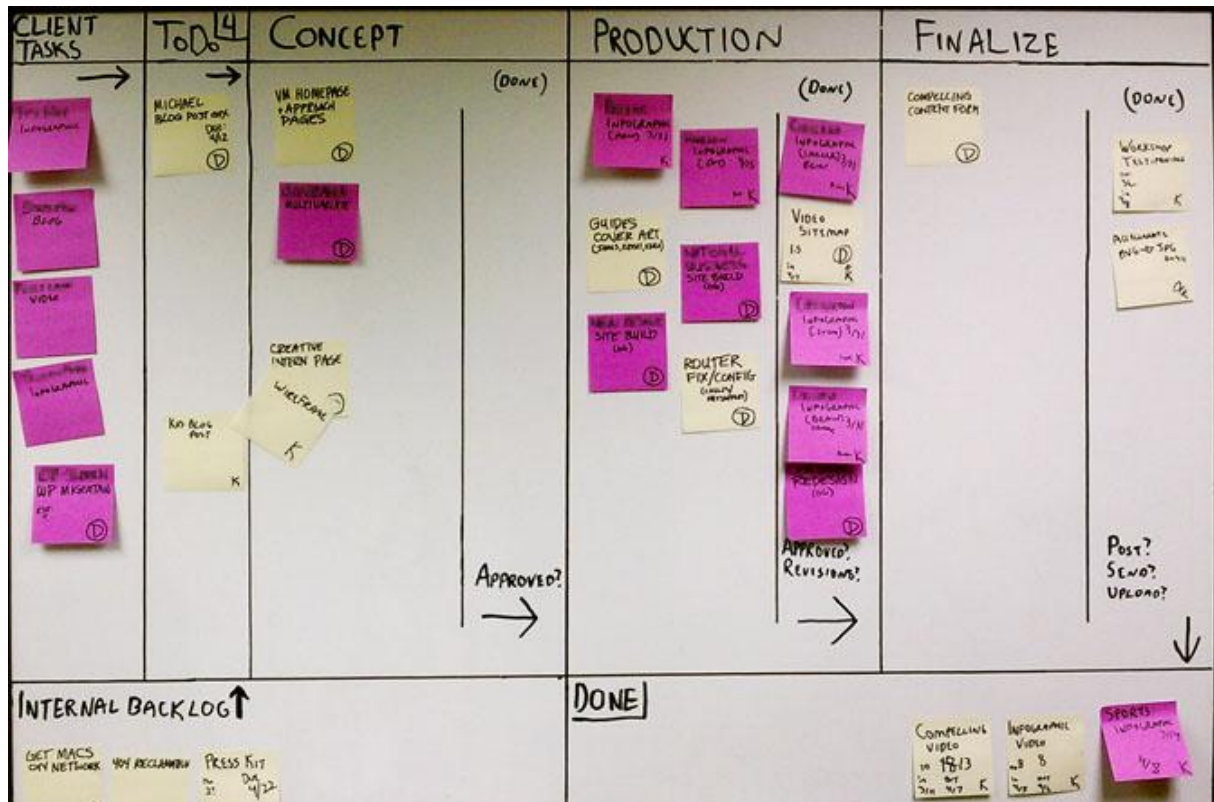
**Taiichi, Ohno. 1988.** *Toyota Production System: Beyond Large-Scale Production*. místo neznámé : Productivity Press, 1988. ISBN: 978-0-915299-14-0.

**Vallo, Michal. 2010.** Kanban – systém vizualizácie vývoja. *zdrojak.cz*. [Online] 2. Srpen 2010. [Citace: 2. Listopad 2011.] <http://zdrojak.root.cz/clanky/kanban-system-vizualizacie-vyvoja/nazory/>.

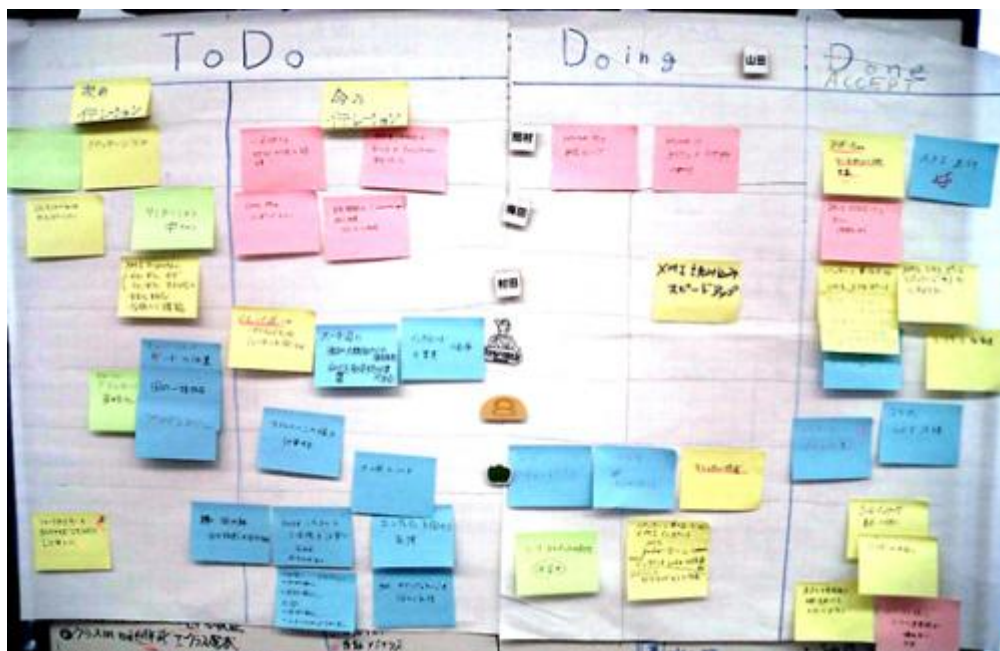
**Wikipedie, Příspěvatelé. 2011.** Kanban. *Wikipedie*. [Online] Wikimedia Foundation, Inc, 2. Prosinec 2011. [Citace: 3. Prosinec 2011.] [http://en.wikipedia.org/wiki/Kanban#cite\\_note-4](http://en.wikipedia.org/wiki/Kanban#cite_note-4).



## Příloha: Ukázka reálných Kanban boards



Obrázek 9: Zdroj <http://www.verticalmeasures.com/wordpress/wp-content/uploads/2011/04/Kanban-Board.jpg>



Obrázek 10: Zdroj [http://www.infoq.com/resource/articles/agile-kanban-boards/en/resources/Fig1\\_task-board.jpg](http://www.infoq.com/resource/articles/agile-kanban-boards/en/resources/Fig1_task-board.jpg)