

<b>Semestrální práce ke kurzu 4IT421 Zlepšování procesů budování IS</b>	
<b>Semestr</b>	<b>LS 2017/2018</b>
<b>Autoři</b>	<b>Jakub Menda (xmenj14) Ondřej Schrek (scho03)</b>
<b>Téma</b>	<b>Scrumban</b>

## Abstrakt

Cílem této práce je popsání a vysvětlení základních principů metodiky (či rámce) agilního řízení Scrumban. V práci je stručně nastíněno pozadí vzniku Scrumbanu, objasňující jeho původní charakter jakožto přechodného stavu mezi metodikami. Jsou popsány základní principy a praktiky Scrumbanu a zdůrazněny jsou i zamýšlené přínosy ve srovnání s použitím rámce Scrum. Je zhodnoceno používání Scrumbanu v praxi a k závěru práce jsou uvedeny i nástroje pro podporu metodiky.

## Klíčová slova

Scrumban, Scrum, Kanban, Agilní metodiky, Projektový management

## Obsah

Abstrakt .....	1
Klíčová slova .....	1
1. Úvod .....	2
1.1 Co je to Scrumban a proč bychom mu měli věnovat pozornost .....	2
1.2 Cíl práce .....	2
1.3 Koncepce práce .....	2
2. Vznik a historie Scrumbanu .....	3
3. Kuchařka Scrumbanu .....	3
3.1 Iterace versus pull princip .....	3
3.2 Kanban tabule .....	4
3.3 Vlastníci úkolů .....	6
3.4 Plánovací versus release intervaly .....	7
3.5 Řízení událostmi .....	7
3.6 Cumulative Flow Diagram namísto burndown diagramu .....	8
3.7 Shrnutí .....	9
4. Nástroje na podporu metodiky .....	10
Závěr .....	12
5. Zdroje .....	13

# 1. Úvod

## 1.1 Co je to Scrumban a proč bychom mu měli věnovat pozornost

Agilní přístupy patří stále nepochybně mezi paradigmaty, která zásadně mění projektové řízení, a to zejména v oblasti vývoje softwaru. Přestože agilní manifest (Agile Manifesto) vznikl téměř před dvaceti lety (Beck, Kent et al., 2001) a oblíbený rámec Scrum je ještě o mnoho let starší, v dnešní době jsou tyto přístupy a metodiky stále považovány za velmi moderní přístupy pro řízení projektů a v mnoha podnicích jsou stále neznámé a jsou teprve objevovány a zaváděny.

Agilní řízení projektů samozřejmě není řešením každého problému. Implementace agilní metodiky nebo frameworku znamená poměrně rozsáhlé změny v celém chápání projektu a jeho řízení, a to napříč vývojovým týmem i celým podnikem. Ačkoli někteří tvrdí opak, stále převládá názor, že agilní metodiky nejsou použitelné v každém projektu, ať už kvůli povaze projektu nebo složení týmu. Agilní metodiky stále narážejí na zavedené mechanismy v businessu, jako je například veřejné soutěžení nebo tradiční rozvětvené byrokratické organizační struktury. Kvůli tomu také mnohdy narážejí na nepochopení při jejich implementaci.

Agilní přístupy jsou proto něčím, co sice v současné době výrazně mění projektový management, ale velmi pravděpodobně se budou samy dále zdokonalovat a měnit, protože to bude nutné pro jejich zavádění a používání v praxi. Je součástí zcela přirozeného vývoje, že vznikají nové přístupy a metodiky, které přidávají k těm stávajícím zcela nové koncepty a modifikují je s jinými praktikami managementu.

Scrumban je jedním z takových přístupů. Není doslova metodikou a jeho výklad je oproti metodikám v klasickém pojetí poněkud volnější, nicméně je to minimálně koncept, který si zasluhuje pozornost.

## 1.2 Cíl práce

Cílem této práce je popsat principy Scrumbanu a impulsy, které v první řadě vedly k obohacování Scrumu o metody Kanbanu a obecně o implementaci tažného (pull) systému v prostředí Scrumu, čímž v zásadě Scrumban je. Společně s tím budou v práci popsány i potenciální přínosy využití Scrumbanu ve srovnání s klasickým Scrumem založeným na časově ohraničených iteracích.

## 1.3 Koncepce práce

Práce je strukturována tak, aby byly zřejmé i impulsy pro vývoj konceptů jako Scrumban, proto bude nejprve velmi stručně nastíněn jeho vznik. V další kapitole bude pozornost věnována především principům Scrumbanu – tedy do jaké míry se jedná o Scrum a co je naopak přejato z kanbanu (a proč). Uvedeny budou praktiky týmu, myšlenky a za nimi a zamýšlené přínosy, stejně jako způsoby měření efektivity alternativní ke metodám Scrumu. V následující kapitole budou opět stručně popsány nástroje na podporu metodiky.

## 2. Vznik a historie Scrumbanu

Jako první použil pojem „Scrum-ban“ Corey Ladas (2008). Ve svém článku mířil k rozšíření agilního chápání projektového řízení o přístupy lean managementu, logickým vyústěním tedy pro něj bylo spojení Scrumu, který byl a stále je jedním z nejrozšířenějších rámců agilního projektového řízení, s Kanbanem. Kanban měl obohatit Scrum nejen o silnou vizualizaci workflow, ale hlavně narušit klasické paradigma časových iterací, známých ve Scrumu jako Sprints, tažným (pull) systémem. Tedy systémem, podle kterého plánování probíhá jen tehdy, je-li to potřeba. Scrumban tedy není o prvoplánovém přidání kanban tabule do zaběhnutého Scrumu, ale především o implementaci pull systému.

Ladas (2008) původně o Scrumbanu mluvil jako o způsobu přechodu ze Scrumu k více „vyvinutým“ metodikám, tedy prostředek, který umožní fungujícím Scrum týmům zdokonalit plánování a případně se posunout ke zcela jiné metodice. Celý Ladasův článek tedy není koncipován jako popis principů metodiky, ale jako výklad několika úrovní, ve kterých je kanban postupně začleňován do klasického Scrumu. Tento princip byl rozšířen v rozsáhlejší publikaci (Ladas, 2009).

Ve skutečnosti se ale Scrumban sám vyvinul v plnohodnotný a v praxi použitelný rámec<sup>1</sup>, nebo spíše sadu principů přejatých ze Scrumu či z Kanbanu, zkombinovaných takovým způsobem, který v závěru poskytuje potenciál výrazně posílit agilitu celého řešení. Podle portálu Agile Alliance (Pahuja) je Scrum obecně vhodnější spíše pro produktový vývoj, zatímco Kanban je systém vhodný pro údržbu. Pokud to ale zobecníme, údržba není správným slovem. Kanban je vhodný tehdy, může-li práce být řízena událostmi – reakce musí být co nejrychlejší a „odpad“, neboli zbytečně provedená práce, co nejnížší.

## 3. Kuchařka Scrumbanu

### 3.1 Iterace versus pull princip

Ideální plánovací proces by měl poskytnout vývojovému týmu pouze práci (úkoly), které jsou z byznys hlediska momentálně prioritní. Ne více, ne méně. To je problém, který se Scrumban snaží řešit. To znamená, že ať je Sprint klasického Scrumu jakkoli krátký, stále se jedná o časové plánování. Práce je pro Sprint vybírána dávkově a případné změny se realizují až v dalším Sprintu (pokud se změny realizují v této iteraci, jedná se velmi pravděpodobně o špatně implementovaný Scrum). Jinak řečeno, plánování Sprint Backlogu v klasickém Scrumu vytváří podstatně větší objem naplánované práce, než je třeba. Ladas (2008) tento přebytek označuje za zbytečný inventář a tím pádem odpad (resp. množství zbytečné práce).

---

<sup>1</sup> Názvosloví v tomto případě (stejně jako v případě Scrumu) není sjednocené. Scrumban tedy někteří označují jako metodiku, někteří jako rámec (framework)

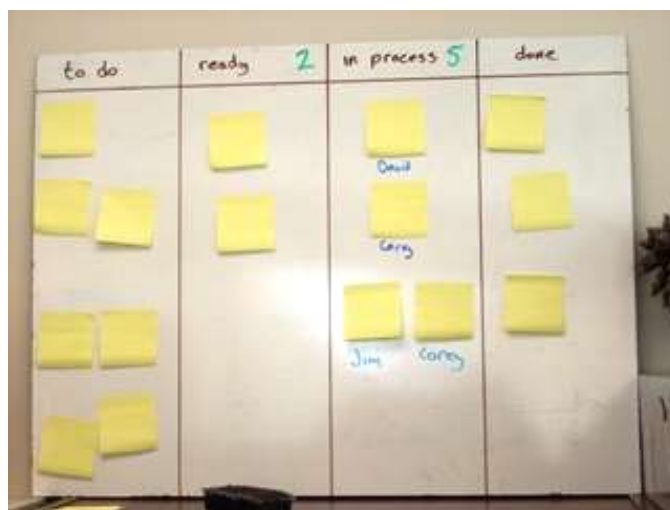
Kanban naopak funguje na tažném (pull) principu, a pull princip je logicky neaplikovatelný s časově ohraničenou iterací. Jedinou limitací objemu práce v kanbanu je velikost jednotlivých front (sloupců) na kanban tabuli – tedy limit WIP (Work In Progress).

Limitace WIP je spolu s vizualizací workflow jedním z hlavních principů kanbanu, jde o mechanismus zabraňující hromadění nedokončené práce. Pro každý stav kanban kartičky – tedy řekněme konkrétního úkolu nebo funkcionality – definuje maximální počet kartiček, které se v tomto stavu mohou nacházet. Jinak řečeno, s počtem kartiček na tabuli pracujeme jako s oběživem a oběh kontrolujeme. Předchází se tím případům, kdy se kartičky hromadí v určitém stavu, ale nejsou dokončeny. A tím je zajištěn zmíněný pull princip – práce jen plyne, neexistuje začátek ani konec iterace.

Jak již bylo zmíněno výše, Ladas (2008) představil Scrumban jako pozvolný přechod. Logicky tedy nedoporučuje hned na první úrovni zrušit iterace. Navrhuje ale mechanismus, který podporuje dokončení rozpracované práce před započítáním nové – tedy zavedení kanban tabule s definovaným WIP limitem do klasického Scrumu. WIP limit by měl být nastaven optimálně pro tým, tak, aby blokový člen mohl začít na jiném úkolu, ale v ideálním případě ne na více než jednom.

### 3.2 Kanban tabule

Mezi základní stavební kameny Scrumbanu tedy patří kanban tabule, přičemž jako první kroky výše zmíněného přechodu od klasického Scrumu Ladas (2008) navrhuje rozšiřování této tabule, tedy specifikace konkrétnějších stavů. Do jednoduchého tříslopcového rozložení „to do“, „in process“ a „done“ tedy zařazuje sloupec „ready“, a to mezi sloupce „to do“ (tedy produktový Backlog) a „in process“. Sloupec „ready“, ať už si ho každý tým nazve dle svého, je logickým předpokladem aplikace tažného principu. Pokud tým dokončí práci v „in process“, „ready“ je sloupec, ze kterého se bude brát další kartička. Teoreticky bychom mohli brát další kartičku i z „to do“, neboli produktového Backlogu, problém je ale v tom, že produktový Backlog neodpovídá prioritám dle byznys požadavků.



**Obr. 1: Klasická kanban tabule doplněná o sloupec „ready“ (Ladas, 2008)**

Už tato drobná změna výrazně přispívá k celkové agilitě, protože rozhodování o dále provedené práci není prováděno jednou za iteraci, ale kontinuálně. Jakákoli další kartička tedy bude z byznysového hlediska vždy prioritní a na změny lze reagovat prakticky okamžitě.

Další štěpení sloupců dle Ladase (2008) spočívá ve vyčlenění sloupce „specify“ ze sloupce „in process“. Takový stav zahrnuje především objevování činností, které je třeba vykonat, aby mohla být kartička přesunuta do „done“. Na druhou stranu potom další nový sloupec „execute“ bude o samotném vykonávání právě těchto činností. Takovým rozdělením sloupce „in process“ se pochopitelně bude dělit i WIP.



**Obr. 2: Sloupce „specify“ a „execute“ (Ladas, 2008)**

Nakonec Ladas (2008) ještě přidává sloupec „specify – complete“, tedy sloupec, odkud tým tahá již dostatečně specifikované kartičky do sloupce „execute“.



**Obr. 3: Sloupec „specify-complete“ (Ladas, 2008)**

Již tento přístup napovídá tomu, že Scrumban v tomto ohledu více nahrává specializaci členů týmu. Dobře definované workflow umožňuje členům týmu zaměřit se pouze na část procesu, kterou ovládají velmi dobře, a zbytek přenechat dalšímu. Na jednu stranu jde toto ruku v ruce s implementací pull systému a opravdu to do jisté míry připomíná maximálně efektivní výrobní linku, na druhou stranu, schopnost každého člena rozumět každé části procesu je jedním z osvědčených principů Scrumu (důraz na znalosti členů, viz subkapitulu 3.3). Je ovšem dobré vědět, že použitím Scrumbanu má tým tuto možnost, a záleží pouze na týmu, jak s ní naloží.

Na konci máme stále poměrně jednoduchou kanban tabuli, závisí tedy na týmu a konkrétní implementaci, jak bude rozšířena. Přístup, který Ladas (2008) při rozšiřování sloupců aplikoval, bylo v zásadě vytváření jakýchsi meziprocesních nárazníků, za jejichž pomoci, v kombinaci s Cumulative Flow Diagramu (viz níže), lze identifikovat slabá místa celého workflow. Zásadní je při vytváření nových sloupců jasně definovat podmínky a politiku, podle kterých se mohou stavy kartiček měnit (tedy za jakých podmínek můžeme kartičku přesunout do dalšího sloupce).

### 3.3 Vlastníci úkolů

S časově plánovanými iteracemi úzce souvisí také problematika tzv. vlastníků úkolů, neboli přiřazování práce konkrétním členům týmu. V některých případech a implementacích Scrumu dochází k přiřazování úkolů konkrétním lidem ještě před skutečným započítáním Sprintu – tedy při Sprint Planningu. Tento přístup je pochopitelný – pokud je odhadnuta náročnost (effort) konkrétního úkolu, a pokud je naplánovaný objem práce v tomto Sprintu, je nasnadě přiřadit úkoly jednotlivcům, kteří mají například s danou problematikou více zkušeností.

Tento přístup ale trpí minimálně dvěma zásadními nedostatky. Zaprvé, tým se se neučí. Scrum mimo jiné zakládá na týmu s tzv. „T-shaped skills“ (představíme-li si graf znalostí členů, z grafu o tvaru písmena „T“ můžeme vyčíst, že obecnou znalost o všem, ale podstatně hlubší znalost o jedné konkrétní věci). Ve skutečnosti přístup, kdy rozdělíme práci jednotlivcům už při plánování Sprintu, tuto myšlenku

příliš nepodporuje, spíše nahrává specializaci jednotlivých členů (což nám vytváří zajímavou kontradikci k předchozí subkapitole).

Zadruhé, a co je pravděpodobně závažnější, tímto přístupem si uměle vytváříme kritickou cestu v rámci Sprintu. Potom může jednoduše dojít k tak banálním případům, kdy polovina týmu nemá nic na práci, ale Sprint se nestihne.

Odhady náročnosti by proto měly sloužit jako metrika především týmu, nikoli jako podklad pro plánování práce jednotlivců.

Scrumban právě díky pull principu odstraňuje jakákoli pokušení plánování práce jednotlivců. Jestliže neplánujeme časovou iteraci, nemůžeme přiřazovat úkoly jednotlivcům a pull princip zajistí, že kdokoli z týmu je volný, začne pracovat na kartičce s nejvyšší prioritou. Pull princip tedy přímo vynucuje, aby vývojový tým vystupoval jako jedna role.

### 3.4 Plánovací versus release intervaly

Scrum a jeho časově plánované iterace jsou pevně spjaty i s release intervalem, tedy v ideálním případě je koncem Sprintu nová funkcionality otestována a vydána. Jestliže ale existuje nějaký vhodný interval pro release a nějaký jiný interval pro plánování iterace, vynucování souladu těchto dvou je umělé. A pokud již aplikujeme na náš proces pull princip, nemáme ani jinou možnost, než release interval oddělit a vydávat tehdy, kdy je to vhodné. Na té nejvyšší úrovni Scrumbanu bychom tak měli release interval, stejně jako plánování, řízený specifickou událostí – fungovala by tedy specifická on-demand funkce, která by vyžádala vydání pouze ve chvíli, kdy jsou splněny určité podmínky a má k němu tedy dojít.

### 3.5 Řízení událostmi

Jedním ze základních principů Agilního manifestu (Beck, 2001) je vítání změnových požadavků i v pozdějších stádiích vývoje, tedy jinak řečeno schopnost rychlé reakce na změnu byznys požadavků. Backlog by se tedy měl neustále měnit, aby hodnotu pro byznys vždy adekvátně odrážel. Je tedy řízený událostmi (event-driven). V časovém plánování můžeme říct, že tou událostí je právě nějaký časovač – jinak řečeno před začátkem Sprintu udělujeme práci priority a definujeme Sprint Backlog. Jestliže ale aplikujeme pull princip, není těžké si představit další a vhodnější druhy událostí, což vede k lepším reakcím na změny v prioritách.

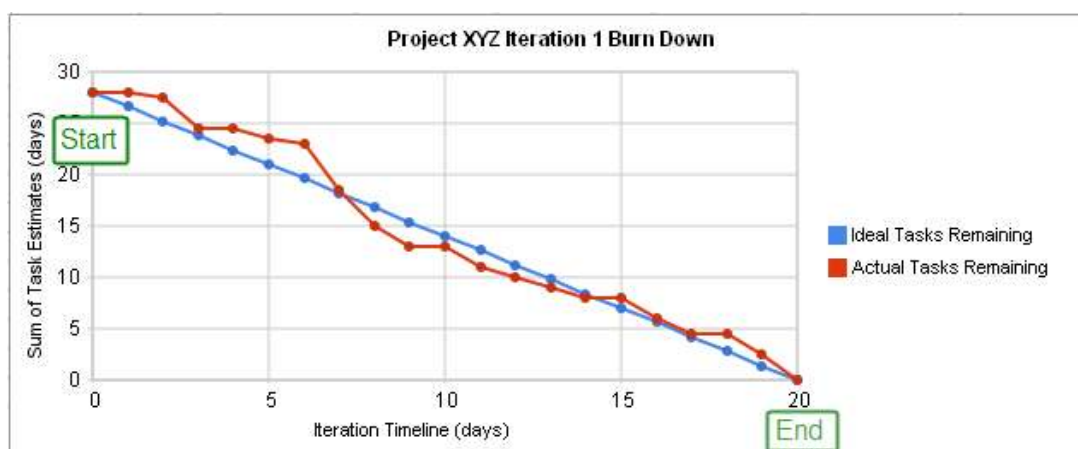
Poslední stupeň Scrumbanu časovou událost nahrazuje konceptem tzv. bodu objednávky („order point“) (Ladas, 2008). Pokud taháme kartičky z Backlogu do sloupce „in-process“, snižujeme tím počet nevyřízených položek. Bod objednávky není nic jiného než pevná hranice tohoto počtu nevyřízených položek. A pokud počet nevyřízených položek klesne pod tuto hranici, máme hledanou událost, která oznámí odpovědným stranám potřebu dalšího plánování (naplňování „to do“ prioritními položkami). Výše bodu objednávky bude samozřejmě záležet na specifikách společnosti – tedy na schopnosti týmu

(například kolik položek průměrně řeší denně) nebo i na schopnosti osob zodpovědných za plánování Backlogu rychle reagovat. Jinak řečeno, nedává smysl stanovit příliš nízký bod objednávky, pokud projektový manažer není schopen do dne zareagovat, přestože ideální stav je samozřejmě bod objednávky o výši jediné položky.

Pokud se tým dostane do této fáze, udržovat časově ohraničené iterace už postrádá smysl a projekt je řízen dle zásad pull principu. Jak již bylo zmíněno v předchozí subkapitole, zbývá už jen stejný princip aplikovat pro release intervaly – tedy nalézt vhodné události, které budou spouštět nasazení. Tím je zcela odděleno plánování práce od plánování vydání a projekt je zbaven časového plánování.

### 3.6 Cumulative Flow Diagram namísto burndown diagramu

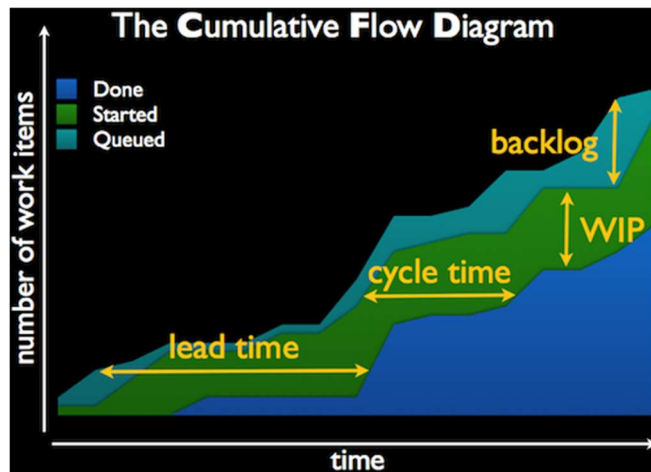
Pro sledování postupu projektu se ve Scrumu používá burndown chart, znázorňující na svislé ose hodnotu, která se může podle potřeb Scrum mastera lišit, běžně se používá počet mandays z odhadů, story pointy, případně počet stories, které jsou spotřebovány či vyřešeny. Na vodorovné ose jsou dny sprintu. Způsob kontroly ve Scrumu je poměrně přímočarý, kontrolujeme průběh reálné křivky proti lineární klesající funkci, která nám indikuje ideální průběh sprintu. Pokud jsme nad křivkou, jsme ve zpoždění, pokud pod ní tak jsme v předstihu. V Kanbanu se standardně podobné grafy nepoužívají právě z důvodu neexistence uzavřených cyklů, v rámci kterých by se měřily hodnoty.



Obr. 4: Ukázka Scrum burn-down chart (CC BY-SA 3.0, 2011)

Avšak v rámci Scrumbanu byl zaveden kumulativní graf, pomocí kterého můžeme sledovat množství tasků v jednotlivých stádiích a v čase. Jedná se o vhodnou metriku pro zjištění „úzkých hrdel“ v rámci projektu a umožňuje kontinuální zlepšování procesů. Tento kumulativní graf podporuje řízení projektu, a například proti klasickému Kanbanu, který standardně statistické grafy nemá, nám pomáhá aplikovat všeobecnou poučku: „co nemůžeme změřit, nemůžeme efektivně řídit“.





Obr. 5: Ukázka Cumulative Flow Diagram (Zsolt Fabók, 2011)

Ve zjednodušeném grafu jsou tři stádia tasku, a to backlog (tyrkysová), in progress (zelená), done (modrá). Svislá osa zobrazuje počet tasků, vodorovná čas ve dnech. První hodnota, kterou můžeme sledovat, je WIP (výše zmiňovaný limit Work In Progress), tedy počet tasků, které se aktuálně vykonávají. Stejně tak můžeme sledovat, kolik tasků se hromadí v backlogu. V rámci každého stadia můžeme sledovat čas, který stráví tasky v jednotlivém stádiu (cycle time). Analogicky můžeme sledovat napříč stádii, tj. od backlogu po done, který ukazuje čas průchodu přes všechna stádia tasku (lead time). Díky těmto metrikám můžeme sledovat průchody celým stadiem a současně i časy v jednotlivých stádiích. Metriky nám pomohou zlepšovat konkrétní procesy v rámci stádií a také v případě potřeby dynamicky upravovat limit tasků pro jednotlivá stádia.

### 3.7 Shrnutí

Tato část práce shrnuje nejdůležitější principy Scrumbanu, které je nutné při zavádění brát na vědomí. Prvním podstatným principem je udržovat optimální velikost backlogu, tak, aby měl každý člen týmu práci a zároveň, aby backlog nepřetékal tasky k řešení, které vytváří zmatek a především vedou k neefektivitě, potenciálnímu zbytečnému či špatně prioritizovanému plánování. Samotný backlog je řízen událostmi, tj. plnění je na základě aktuální potřeby, přidávání není omezeno časovými úseky. Tasky jsou vždy prováděny podle priority, tj. prioritní předbíhají méně důležité.

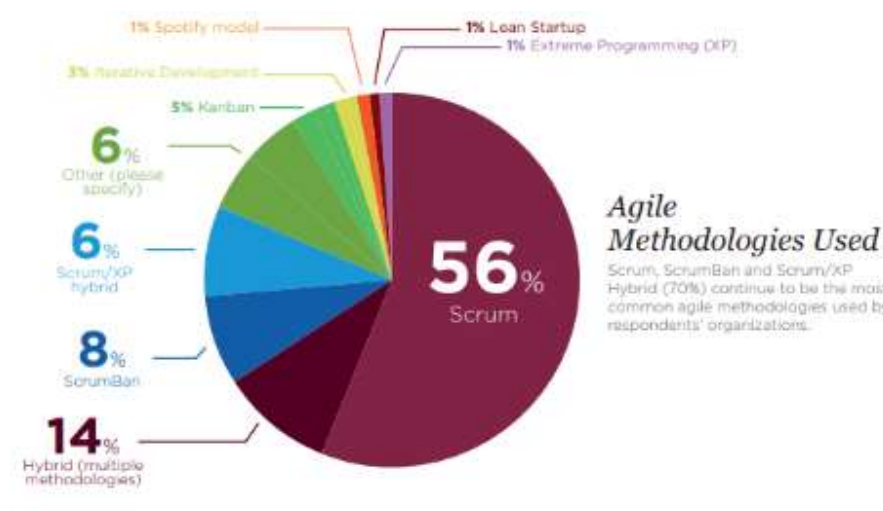
Nezbytným krokem je nastavení WIP limitů sloupců, tj. každý sloupec (stavy tasků) je kapacitně omezen. Při stanovování používáme CFD, diagram, který nám v čase indikuje problémové úseky při zpracovávání tasků (úzká hrdla) a pomůže efektivně nastavit WIP limit pro sloupec.

Je vhodné implementovat standardní zvyky ze Scrumu jako jsou role v týmu, daily meetingy (tzv. standupy) a také retrospektivu, která je prováděna za určitý časový úsek. Naopak proti Scrumu, využíváme kontinuální planning, dle potřeby. Vypouštíme také Sprint Backlog, který se redukuje na backlog na boardu, nepoužívají se časové iterace a většinou ani nepoužíváme odhady tzv. Planning

Poker, používáme průměrný lead time (díky průběžnému vedení CFD se odhady zpřesňují). Oproti Scrumu podporujeme specializace členů týmu, dbáme ale ideálně na tzv. T-shaped skills. Změny v procesech (řídících, výkonných) se mohou dělat okamžitě, dle potřeby.

Podle průzkumů organizace State of Agile z roku 2018 (VersionOne, © 2018), v praxi používá metodiku Scrumban 8% uživatelů agilního přístupu. Zároveň Scrumban předčil samotný Kanban, který používá 5% uživatelů. To svědčí o jeho kvalitách, které jsou ověřovány jeho uživateli v praxi.

Za zvyšující oblibou může stát fakt, že Scrumban je sadou principů, která tříbí výhody ze dvou oblíbených agilních metodik. Navíc je vhodný pro organizace, které by rády využívaly principy a výhody Scrumu, ale nemohou uplatnit všechny principy Scrumu, například ve výrobě, či pro service desk, kde nemůžeme použít například iterace. Zde nalezne Scrumban uplatnění. Samotná metodika je dnes již hojně používaná a není problém dohledat praktické zkušenosti ze zavádění a používání. Také je dostupné množství literatury, která detailně Scrumban popisuje.



**Obr. 7: Použití agilních metodik v praxi 2018 (VersionOne, 2018)**

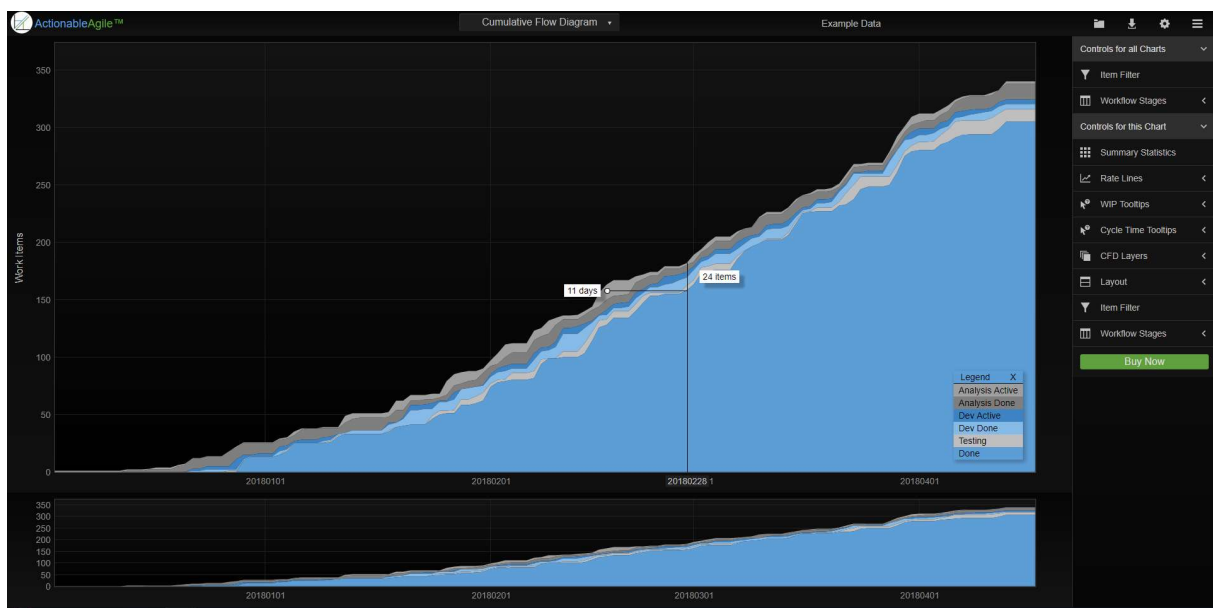
V neposlední řadě je třeba zmínit, že stále vycházíme primárně z frameworku Scrum – čili zachováváme jeho role, rituály a další artefakty, a pouze některé z nich (ty v práci zmíněné) modifikujeme či nahrazujeme implementací pull principu.

#### 4. Nástroje na podporu metodiky

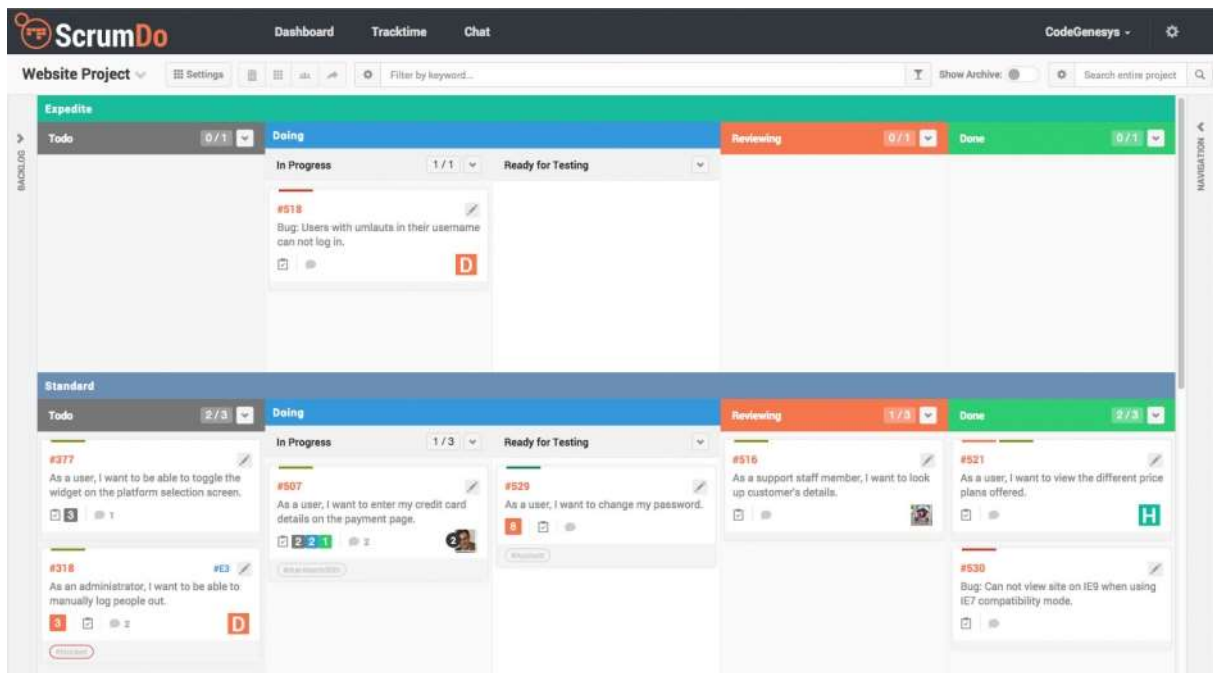
Standardně se pro Scrumban používá klasická tabule. Ladas ve svých článcích upřednostňuje velkou fyzickou tabuli oproti virtuální podobě za pomoci softwaru. Výhodou fyzické tabule je možnost rychlých a snadných změn, všichni mají na očích pouze jednu podobu boardu a mezi další výhody patří její větší přehlednost s možností sledovat v celé perspektivě. Avšak od prvních Ladasových publikací uplynulo

téměř 10 let a technika nám poskytuje lepší možnosti. Výhodou je, že pro Scrumban můžeme používat softwarové nástroje určené pro Kanban nebo Scrum, které se dají dobře customizovat. Avšak nejlepší je takový nástroj, který umožňuje generovat CFD a další statistiky, což většinou nástroje pro Scrum nativně nepodporují, tudíž nemají o moc větší přínos než klasická tabule. Právě podpora automatických statistik dává vysokou přidanou hodnotu, kterou fyzická tabule neumožňuje, ušetří tím agendu, kterou by musel zpracovávat řídicí pracovník.

Jedny mnoha z nástrojů splňující výše uvedené požadavky jsou například Kanbantool, nebo ScrumDo. Pokud však chceme zachovat fyzickou tabuli a pouze ušetřit zpracování analýz, hodí se nám nástroj ActionableAgile, který umožňuje při vložení dat velmi podrobné analýzy.



Obr. 6: Ukázka nástroje ActionableAgile (Schrek O., 2018)



Obr. 6: Ukázka nástroje ScrumDo (Schrek O., 2018)

## Závěr

V práci se podařilo zmapovat historii vývoje, popsat nejdůležitější principy, rozebrat impulsy, které vedly k potřebě vytvoření alternativy pro zavedené agilní metodiky a v neposlední řadě uvést výhody a nevýhody tohoto rámce pro řízení projektu. Byl vysvětlen smysl jednotlivých komponent Scrumbanu i myšlenkové paradigma stojící za jejich vznikem. Během tvoření práce se nepodařilo získat relevantní informace o používání v rámci České Republiky, což byl jediný problém, který se vyskytl během vypracování. Avšak podařilo se nalézt výzkumy, které sledovaly použití ve světě, viz studie společnosti VersionOne (2018).

## 5. Zdroje

BECK, Kent et al., 2001. Principles behind the Agile Manifesto. Manifesto for Agile Software Development [online]. 2001. [cit. 2018-04-03]. Dostupné z: <http://agilemanifesto.org/iso/en/principles.html>

LADAS, Corey, 2008. Scrum-ban. *Lean Software Engineering, Essays on the Continuous Delivery of High Quality Information Systems* [online]. [cit. 2018-04-03]. Dostupné z: <http://leansoftwareengineering.com/ksse/scrum-ban/>

LADAS, Corey, 2009. *Scrumban: Essays on Kanban Systems for Lean Software Development*. Modus Cooperandi Press. ISBN 978-0578002149.

PAHUJA, Savita. What is Scrumban? *Agile Alliance*. [cit. 2018-04-03]. Dostupné z: <https://www.agilealliance.org/what-is-scrumban/>

VERSIONONE, © 2018. 12th Annual State of Agile Report. Versionone.com [online]. [cit. 2018-05-02]. Dostupné z: <https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report>.