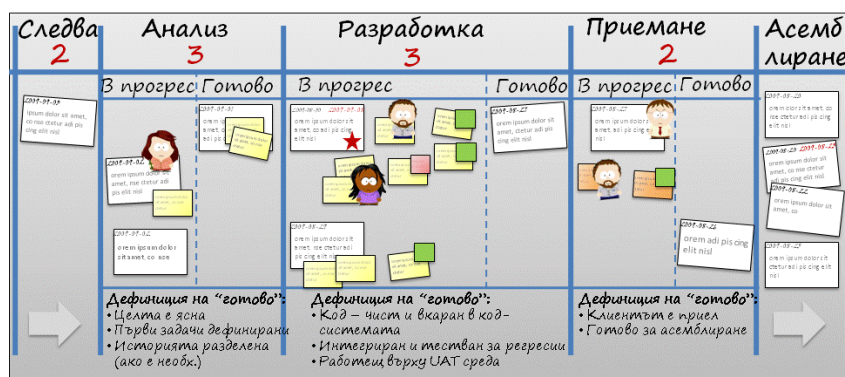


Канбан и Скръм - как да вземем най- доброто и от двете



Хенрик Книберг & Матиас Скарин
Предговор от Мери Попендийк & Дейвид Андерсън

© 2010 C4Media Inc.
Всички права запазени.

C4Media, издавателство на InfoQ.com.

Тази книга е част от серията книги на InfoQ Enterprise Software Development.

За информация или поръчка на тази или други книги на InfoQ, пишете на адрес books@c4media.com.

Никаква част от тази публикация не може да бъде възпроизвеждана или разпространявана в каквато и да е форма – електронна, механична, чрез фотокопиране, сканиране, или по друг начин, освен описаните в раздел 107 и 108 от закона за авторското право на САЩ от 1976 г., без предварителното писмено съгласие на издателя.

Обозначенията, използвани от компаниите за техните продукти, в много случаи са считани за търговски марки. Във всички случаи, където на C4Media Inc. е известно за съществуваща търговска марка, имената на продуктите са изписани с Начални Главни Букви или ИЗЦЯЛО ГЛАВНИ БУКВИ. Въпреки това, читателите трябва да се свържат със съответната компания за по-цялостна информация относно търговски марки и регистрации.

Главен редактор: Диана Плеса
Оформление на корицата : Bistran IOSIP
Предпечат: Assurance

Каталожен номер в Библиотека на Конгреса, САЩ:
ISBN: 978-0-557-13832-6

Превод:
Зорница Николова – zornitsa.yankulova@gmail.com
Любомир Пашов – lpashov@gmail.com
Моника Ковачка-Димитрова – monika.kovachka@gmail.com
Цветелина Петева – tsvetelina.peteva@gmail.com

 *Leanify* - <http://leanify.com>

Съдържание

ПРЕДГОВОР ОТ МЕРИ ПОПЕНДИЙК.....	v
ПРЕДГОВОР ОТ ДЕЙВИД АНДЕРСЪН	vii
ВЪВЕДЕНИЕ	xi
ЧАСТ 1 – СЪПОСТАВКА	1
1. Какво все пак представляват Скръм и Канбан?.....	3
2. Каква е връзката между Скръм и Канбан?.....	7
3. Скръм дефинира роли	12
4. Скръм налага итерации, ограничени във времето.....	13
5. Канбан ограничава броя недовършени задачи чрез лимит на всяка фаза от работния поток, а Скръм – чрез итерация	15
6. И двата са емпирични	18
7. Скръм се противопоставя на промяната по време на итерация ..	25
8. Скръм дъската се изчиства преди всяка итерация	27
9. Скръм препоръчва мултифункционални екипи	29
10. Списъкът със задачи в Скръм трябва да пасне в спринта	31
11. Скръм изисква оценяване и темпо.....	33
12. И двата позволяват работа по няколко продукта паралелно.....	35
13. И двата са Лийн и Аджайл.....	37
14. Малки разлики	39
15. Скръм дъска срещу Канбан дъска – нетривиален пример.....	43
16. Обобщение на сравнението на Скръм с Канбан.....	51
ЧАСТ 2 – ПРИМЕР ОТ ПРАКТИКАТА.....	55
17. Природата на администраторската работа.....	57
18. Защо, за Бога, да променяме нещата?	59
19. Откъде да започнем?	61
20. Да накараме нещата да работят	63
21. Да стартираме с екипите.....	65
22. Да се обърнем към заинтересованите лица.....	67
23. Да създадем първата дъска.....	69

24. Да поставим първите ограничения за недовършените задачи (WIP).....	73
25. Да съблюдаваме ограниченията заброя недовършени задачи (WIP).....	75
26. Кой задачи стигат до дъската?.....	77
27. Как да правим оценки?.....	79
28. И така, как работехме в действителност?.....	81
29. Да намерим концепция за планиране, която работи.....	85
30. Какво да мерим?.....	89
31. Как започнаха да се променят нещата.....	93
32. Общи изводи и поуки.....	101
ФИНАЛНИ ТОЧКИ, КОИТО ДА ЗАПОМНИТЕ	104
ЗА АВТОРИТЕ	107
РЕЧНИК	109

Предговор от Мери Попендийк

Хенрик Книберг е от онези рядко срещани хора, които могат да извлекат същността от всяка сложна ситуация, да отсеят основните идеи от случайните хрумвания и да предоставят кристално ясно и невероятно лесно за разбиране обяснение. В тази книга Хенрик брилянтно обяснява разликата между Скръм и Канбан. Той подчертава, че те са само инструменти, а това, което наистина ви трябва, е пълен набор от такива, разбиране за силните и слабите страни на всеки от инструментите и знание как да ги използвате.

От тази книга ще научите какво е Канбан, силните и слабите му страни и кога да го ползвате. Ще получите и добри насоки как и кога да надграждате Скръм или друг инструмент, който може би ползвате. Хенрик обяснява, че важното е не с кой инструмент ще започнете, а начинът, по който непрекъснато усъвършенствате употребата му и разширявате набора инструменти с течение на времето.

Втората част на книгата, написана от Матиас Скарин, я прави още по-ефективна, като ви превежда през процеса на прилагане на Скръм и Канбан в реална ситуация. Тук ще видите пример за това как инструментите се използват както поотделно, така и заедно, за да подобрят процеса на софтуерна разработка. Ще забележите, че няма един „най-добър” начин да се направят нещата; вие сами ще трябва да помислите и в зависимост от вашата ситуация да намерите следващата си стъпка към по-добър процес.

Мери Попендийк

Предговор от Дейвид Андерсън

Канбан се базира на много проста идея. Броят на започнати и недовършени задачи (WIP) трябва да се ограничи и нещо ново може да бъде започнато , само когато съществуващите задачи са завършени или предадени на следващата стъпка от поточната линия. Канбан (или сигнална карта) означава визуален сигнал, който показва, че може да се вземе нова задача, защото съществуващите не надминава определеният лимит. Не звучи много революционно, нито пък фундаментално влияещо върху производителността, културата, възможностите и зрелостта на екипа и заобикалящата го организация. Впечатляващото е, че всъщност оказва влияние! Канбан наглед е такава малка стъпка, а променя всичко в един бизнес.

Това, което осъзнахме за Канбан е, че той е подход за управление на промяната. Не е само цикъл или процес за разработка на софтуер или управление на проекти. Канбан е подход за промяна на съществуващия производствен процес или методология за водене на проектите. Принципът на Канбан е да започнете с това, което правите в момента. Разбирате процеса си, като изграждате картата на стойностния поток (value map stream) и дефинирате ограничения за броя недовършени задачи за всяка фаза от този процес. След това започвате да прекарвате задачите през тази система, като вземате всяка следваща, когато се появи канбан сигнал.

Канбан е доказал ефективността си при екипи, които прилагат гъвкав процес за разработка на софтуер, но работи също така добре и при екипи с по-традиционен подход. Канбан се прилага като част от Лийн инициативи (от англ. lean – букв. *слаб*; в този контекст става дума за *оптимизирано производство*) целящи да променят културата на организациите и да насърчат непрекъснатото усъвършенстване.

Тъй като броят недовършени задачи е ограничен в Канбан, всичко, което по някаква причина е блокирано, заплашва да задръсти системата. Ако достатъчно много работни задачи биват блокирани, целият процес спира. Ефектът от това е, че целият екип, както и организацията извън екипа, се фокусират върху решаването на проблема, за да освободят блокираните задачи и да възстановят потока.

Канбан използва механизъм за визуален контрол за проследяване на работата в процеса на изпълнението ѝ през различните етапи на стойностния поток. Обикновено се използва бяла дъска със стикери или електронна стена. Вероятно най-добре е да се използват и двете. Това генерира прозрачност, която допринася за промяна в културата. Гъвките методи са добри за предоставяне на прозрачност по отношение на започнатите и недовършени задачи, завършената работа и метриките, като например темпо (количеството работа, завършено в рамките на една итерация). Но Канбан отива една стъпка по-нататък и осигурява прозрачност по отношение на процеса и неговото протичане. Канбан изкарва наяве затрудненията, опашките, вариациите и излишната работа – всички те оказват влияние върху производителността на организацията, измерена чрез количеството стойностна работа и времето за завършването ѝ. Канбан позволява както на участниците в екипа, така и на външните заинтересовани лица да видят резултата от действията (или бездействията) си. Ранни проучвания показват, че по този начин Канбан променя поведението и насърчава по-силно сътрудничество между въвлечените в процеса. Прозрачността и влиянието върху затрудненията, излишната работа и вариациите насърчава и дискусиите, и екипите бързо започват да въвеждат подобрения в процеса си.

Като резултат, Канбан създава предпоставки за постепенна еволюция на съществуващите процеси - еволюция, свързана с ценностите от Лийн и Аджайл (от англ. agile – *гъвкав*). Канбан не изисква радикална революция на начина, по който работят хората, а насърчава поэтапна промяна. Това е промяна, която е осъзната и приета с консенсус от служителите и техните сътрудници.

Чрез същността на системата на изтегляне (pull system), Канбан предполага и забавяне на обещанията, както за приоритизиране на нови задачи, така и за завършването на съществуващи такива. Обикновено екипите определят ритъм на приоритизация, според който се срещат със заинтересованите лица и да решат по какво ще продължат да работят. Тези срещи могат да се провеждат често, тъй като обикновено са много кратки. Трябва да се отговори на един прост въпрос – например: „От последната ни среща има две свободни места за нови задачи. Времето за изпълнение на задача е 6 седмици до окончателно завършване. Кои са двете неща, които най-много искате да ви бъдат доставени след 6 седмици?“. Това има двоен ефект – простият въпрос обикновено получава бърз и качествен отговор, а срещата остава кратка. Същността на въпроса е, че обещанието по какво ще се работи е забавено до последния възможен момент. Така гъвкавостта се повишава чрез управление на очакванията, намаляване на времето от обещанието до предоставянето на завършената функционалност и елиминиране на преработката, тъй като шансът приоритетите да се сменят в рамките на периода е минимален.

И последно за Канбан – ефектът от ограничаването на броят започнати и недовършени задачи прави времето за завършването им предвидимо, а доставянето на готова функционалност – по-надеждно. Подходът „спри линията“ за отстраняване на пречки и дефекти също насърчава стремеж към много високо ниво на качеството и рязък спад в преработването.

Въпреки, че всичко това ще стане ясно от чудесните и ясни обяснения в тази книга, как се стига дотам ще остане скрито. Канбан не е създаден за един следобед в някакъв невероятен акт на сътворение – той е еволюирал постепенно за няколко години. Много от дълбоките психологически и социологически ефекти, които променят културата, възможностите и зрелостта на организациите, не са били подозирани. Те са открити. Много от резултатите на Канбан са контра-интуитивни. Това, което изглежда като много механичен подход – ограничаване на броя недовършени задачи и изтеглянето на задачи от предходната стъпка, всъщност има дълбок ефект върху хората и начина им на комуникация и съвместна работа.

Нито аз, нито някой от другите хора, занимаващи се с Канбан в ранните му дни, очакваше това.

Търсех в това, което се превърна в Канбан, подход за промяна, който трябваше да се посрещне с минимална съпротива. Беше ми ясно още през 2003 г., че го търсех и заради механичните ползи. Както открих по това време с прилагането на Лийн техники, ако в управлението на текущата работа имаше смисъл, то в ограничаването ѝ имаше още повече: то премахваше усилията за управлението ѝ от процеса. И така, през 2004 г. реших да реализирам изтегляща система, базирана на основните принципи. Получих тази възможност, когато един мениджър от Майкрософт дойде при мен и ме помоли да му помогна с управлението на промяната в неговия екип, който се занимаваше с поддръжката на вътрешните ИТ приложения. Първата реализация на изтеглящата система беше базирана на теорията на ограниченията, позната като „барабан-буфер-въже”. Тя пожъна голям успех – времето за изпълнение спадна с 92%, дебитът се увеличи повече от 3 пъти, а предвидимостта (производителността с оглед на датите за завършване) стигна до много приемливите 98%.

През 2005 г. Доналд Райнертсен ме убеди да рализирам цялостна Канбан система. Получих тази възможност през 2006, когато поех софтуерния отдел в Corbis, Сиатъл. През 2007 г. започнах да говоря за резултатите. Първата презентация беше на Lean New Product Development Summit в Чикаго през май 2007 г. Последва я представяне на Agile 2007 във Вашингтон през август същата година. От 25-те присъстващи трима бяха от Yahoo!: Аарън Сандърс, Карл Скотланд и Джо Арнолд. Те се върнаха в Калифорния, Индия и Великобритания и приложиха Канбан в техните екипи, които вече имаха проблеми със Скръм. Започнах и дискусийна група в Yahoo!, която по времето на написване на този предговор имаше почти 800 членове. Канбан започваше да се разпространява и хората, които го прилагаха, говореха за своя опит.

Сега през 2009 г. Канбан се разпространява бързо и пристига все повече информация. Научихме много за Канбан през последните 5 години и всички продължаваме да учим всеки ден. Моят фокус е да практикувам Канбан, да пиша за Канбан, да говоря за Канбан и да мисля за Канбан, за да мога да го разбера по-добре и да го обясня на

другите. Умишлено се отказах от сравненията на Канбан с други Аджайл методологии, въпреки че положих известни усилия през 2008 г. да обясня защо той трябва да бъде смятан за подход, съвместим с Аджайл.

Оставих на други хора с по-богат опит да отговорят на въпроси, като: „Какви са приликите и разликите между Скръм и Канбан?“. Много се радвам, че Хенрик Книберг и Матиас Скарин се наложиха като лидери в тази област. Вие, търсещите знание в тази сфера, имате нужда от информация, за да вземате информирани решения и да напредвате в работата си. Хенрик и Матиас ви помагат по начин, по който аз не бих могъл. Особено съм впечатлен от задълбочения подход на Хенрик при сравнението и с начина, по който го е представил балансирано и обективно. Илюстрациите му са изпълнени с информация и често ви спестяват четенето на страници текст. Практическият пример на Матиас е важен, защото демонстрира, че Канбан е повече от теория, и ви показва как той може да помогне на вас и вашата организация.

Надявам се тази книга, сравняваща Канбан и Скръм, да ви хареса и да ви даде задълбочени познания за Аджайл като цяло и за Канбан и Скръм в частност. Ако искате да научите повече за Канбан, моля посетете уеб страницата на нашата общност, The Limited WIP Society, <http://www.limitedwipsociety.org/>.

Дейвид Дж. Андерсън

Скуим, Вашингтон, САЩ
8 юли 2009 г.

Въведение

Обикновено не пишем книги. Предпочитаме да уплътним времето си, като помагаме на клиентите да оптимизират и анализират процесите и организацията си. Напоследък, обаче, се забелязва една тенденция и ние бихме искали да споделим какво мислим по този повод. Ето ви една типична ситуация:

- **Джим:** „Най-последно преминахме изцяло към Скръм!”
- **Фред:** „Е, как върви?”
- **Джим:** „Ами, доста по-добре е от преди...”
- **Фред:** „...но?”
- **Джим:** „...но виж - ние сме екип по поддръжка.”
- **Фред:** „Да, и?”
- **Джим:** „Ами, ние харесваме цялата история с приоритизиране на продуктивния списък, със самоорганизиращите се екипи, ежедневния Скръм, ретроспекцията и т.н....”
- **Фред:** „Е добре, какъв е проблемът тогава?”
- **Джим:** „Постоянно се проваляме в спринтовете”
- **Фред:** „Защо?”
- **Джим:** „Защото ни е много трудно да обещаем, че ще свършим определена работа за 2 седмици. Итерациите не са много смислени за нас, ние просто работим по най-важното за деня. Може би трябва да правим едноседмични итерации?”
- **Фред:** „А можете ли да се ангажирате с едноседмичен план? Ще ви позволят ли да се съсредоточите и да се фокусирате на спокойствие за 1 седмица?”
- **Джим:** „Не съвсем, при нас изникват проблеми ежедневно, може би ако правехме еднодневни спринтове...”

- **Фред:** „Вашите проблеми за по-малко от ден ли се решават?”
- **Джим:** „Не, понякога ни отнемат по няколко дена.”
- **Фред:** „Тогава едnodневни спринтове също няма да ви помогнат. Мислили ли сте да се отървете от спринтовете напълно?”
- **Джим:** „Ами, честно казано, бихме харесали това. Но не е ли против правилата на Скръм?”
- **Фред:** „Скръм е просто инструмент. Ти избираш кога и как да го използваш. Не му бъди роб!“
- **Джим:** „Е добре, какво да правим в такъв случай?”
- **Фред:** „Чувал ли си за Канбан?”
- **Джим:** „Какво е това? Каква е разликата между това и Скръм?”
- **Фред:** „Ето, прочети тази книга.”
- **Джим:** „Но аз наистина харесвам останалото от Скръм. Трябва ли да зарежа всичко?”
- **Фред:** „Не, можеш да комбинираш техники.”
- **Джим:** „Какво? Как?”
- **Фред:** „Просто чети...”

Целта на тази книга

Ако се интересувате от гъвкави методологии за разработка на софтуер, най-вероятно сте чували за Скръм, а може би сте чували и за Канбан. Въпросът, който чуваме все по-често, е „какво е Канбан и как може да се сравни със Скръм”. Как се допълват взаимно? Има ли някакви конфликти?

Целта на тази книга е да разсее мъглата, така че да се изясни как Канбан и Скръм могат да се използват във вашия случай.

Моля, кажете ни, дали сме успели!

Част 1 – Съпоставка

Първата част на книгата е опит да се направи обективно и практическо сравнение на Скръм и Канбан. Това е леко обновена версия на оригиналната статия "Канбан срещу Скръм" от април 2009 година. Този статия стана популярна, затова реших да я превърна в книга и помолих моя колега Матиас да я допълни с примери „от фронтовата линия” от един от нашите клиенти. Чудесна работа! Чувствайте се свободни да прескочите направо към част II, ако предпочитате да започнете с примера. Няма да се обидя.

Е, може би само малко.

/Хенрик Книберг/

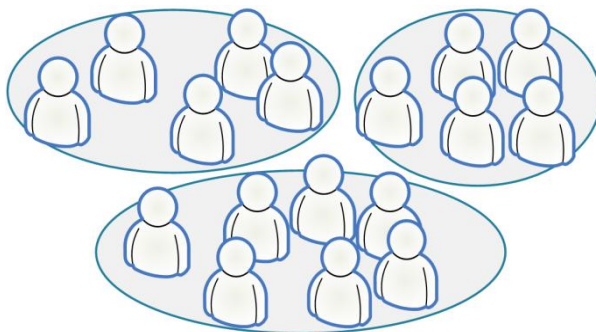
1

Какво, все пак, представляват Скръм и Канбан?

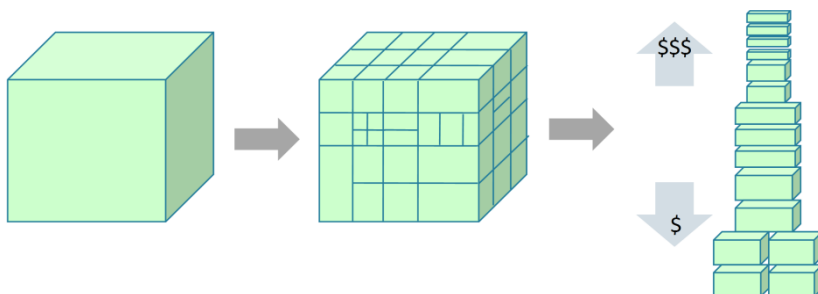
Добре, нека да се опитаме да обобщим Скръм и Канбан с по-малко от 100 думи всеки.

Скръм накратко

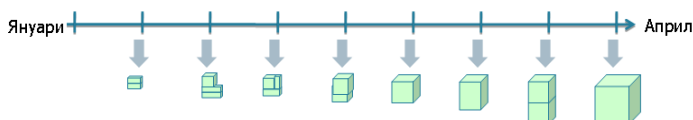
- **Разделете вашата организация** на малки, мултифункционални, самоорганизиращи се екипи.



- **Разделете вашата работа и я организирате** в списък от малки задачи, водещи до конкретни резултати. Сортирайте списъка по приоритети и оценете приблизително усилието за всяка една задача.



- **Разделете времето** на кратки интервали с фиксирана дължина (обикновено 1-4 седмици), с възможност за демонстрация в края на всяка итерация на работеща функционалност, потенциално готова за предаване на клиента.



- **Оптимизирайте плана си за продуктова версия** и приоритизирайте отново заедно с клиента на базата на идеите, събрани при анализа на резултата след всяка итерация.
- **Оптимизирайте процеса**, провеждайки ретроспекция след всяка итерация.

И така вместо **голяма група**, която прекарва **дълго време** изграждайки **нещо голямо**, вие имате **малък екип**, който за **кратко време** изгражда **нещо малко**. Но **интегрира постоянно**, за да се види цяла картинка.

121 думи... справих се добре. (133 в българският превод – Б.П.)

За повече детайли проверете “Скръм и ХР от фронтовата линия”. Книгата е безплатна и може да се прочете онлайн. Познавам автора – готин е ☺

<http://www.crisp.se/ScrumAndXpFromTheTrenches.html>

Повече линкове за Скръм можете да намерите на <http://www.crisp.se/scrum>

Канбан накратко

- **Визуализирайте работния поток**
 - Разделете работата на парчета, напишете всяко парче на карта и поставете на стената.
 - Използвайте колони с имена, за да илюстрирате фазата, в която се намира всяка една задача в работния поток.

- **Ограничете броят недовършени задачи (Work In Progress - WIP)** – определете лимит за това колко започнати задачи може да има във всяка една графа/колона на работния поток.
- **Измервайте време за изпълнение на целия процес (lead time; средното време, за което се изпълнява дадена задача, понякога се нарича *cycle time*), оптимизирайте процеса, така че времето за изпълнение да е възможно най-малко и предвидимо.**



Ние събираме полезни Канбан линкове тук:
<http://www.crisp.se/kanban>.

2

Каква е връзката между Скръм и Канбан?

И Скръм, и Канбан са инструменти за процеси

Инструмент = всяко нещо използвано като средство за изпълнение на задача или цел

Процес = начинът на работа

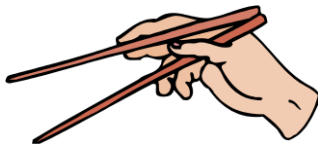
Скръм и Канбан са *инструменти за процес*, които ви помагат да работите по-ефективно, като в известна степен ви казват какво да правите. Java също е инструмент, който ви дава по-лесен начин да създавате програми за компютър. Четката за зъби е инструмент, който ви помага да достигнете зъбите си, така че да ги изчеткате добре.

Сравнете инструментите, за да ги разберете, а не за да ги съдите

Вилица или нож – кой инструмент е по-добър?



Доста безсмислен въпрос, нали? Защото отговорът зависи от контекста. Ако ядете кюфтета, вероятно вилицата е по-добрия избор. За да нарежете гъби, вероятно ножът е по-подходящ. За барабанене по масата стават и двете. При ядене на пържола може би бихте предпочели да ползвате и двата инструмента. За ядене на ориз... ами... някои предпочитат вилица, а други предпочитат клечки.



Затова, когато сравняваме инструменти, трябва да сме внимателни. Сравнявайте за да разберете, а не за да съдите.

Никой инструмент не е нито всеобхватен, нито перфектен

Като всеки инструмент, Скръм и Канбан не са нито перфектни, нито всеобхватни. Те не ви казват изрично *всичко*, което трябва да направите, те просто ви дават някои ограничения и насоки. Например Скръм ви налага ограничение във времето (итерации) и мултифункционални екипи. Канбан ви налага да използвате нагледно дъски и ограничава размера на опашките.

Изненадващо е, че ползата от един инструмент е в това, че *ограничава възможностите*. Един инструмент за процес, който ви позволява да правите всичко, не е много полезен. Можем да наречем такъв процес „Прави каквото и да е“ или например „Прави правилното нещо“. Процесът „Прави правилното нещо“ гарантирано работи, той е решение на всички проблеми! Защото ако не работи, вие очевидно не спазвате процеса :o)

Използването на правилните инструменти ще ви помогне да успеете, но не ви гарантира успех. Много е лесно да се обърка успехът/провалът на проекта с успеха/провала на инструмента, който се ползва.

- Един проект може да успее заради добър инструмент.
- Един проект може да успее въпреки лош инструмент.
- Един проект може да се провали заради лош инструмент.

RUP (от англ., Rational Unified Process – *унифициран процес на Rational*) е силно ограничаващ – има над 30 роли, над 20 дейности, 70 артефакта; доста голямо количество информация, която трябва да се научи. В действителност не се налага да ползвате всичко това, по-скоро се очаква да изберете подходящия набор за вашия проект. За съжаление, на практика се оказва трудно. „Хмммм...“, дали ще ни трябва артефакта „Протокол от проверка на конфигурация“ (*Configuration audit findings*)? Ще имаме ли нужда от ролята „Мениджър по контрол на промените“ (*Change control manager*)? Тъй като не е сигурно, за всеки случай ще е по-добре да ги запазим. Най-вероятно това е една от причините внедряването на RUP да е доста тежко в сравнение с гъвкавите методологии като Скръм и XP.

XP (от англ., eXtreme Programming – *екстремно програмиране*) е с доста повече правила в сравнение със Скръм. Включва се част от Скръм плюс набор от строго специфицирани практики при разработването на софтуер като програмиране, базирано на тестове (*test-driven development*) и програмиране по двойки (*pair programming*).

Скръм е по-малко ограничаващ от XP, тъй като не препоръчва конкретни практики за програмиране. Скръм е по-ограничаващ от Канбан, защото налага практики като итерациите и мултифункционалните екипи.

Една от основните разлики между Скръм и RUP е, че при RUP получавате твърде много и се налага да премахвате излишни неща. В Скръм получавате малко, но се налага да добавите това, което липсва.

Канбан оставя доста отворени врати. Единствените правила са **Визуализирай работния поток** и **Ограничи броя на недовършените задачи**. На сантиметри от **Прави каквото и да е**, но в същото време изненадващо силни.

Не се ограничавайте с един инструмент

Смесвайте и напасвайте инструментите според вашите нужди! Трудно си представям успешен Скръм екип, който не прилага повечето от практиките на XP. Много от Канбан екипите имат ежедневни правостоящи срещи (практика, идваща от Скръм). Някои Скръм екипи описват задачите си като сценарии или „варианти на използване“ (*use cases*, практика от RUP) или ограничават размера на опашките си (практика от Канбан). Правете това, което работи във вашия случай.

Миямото Мусаши, самурай от 17-и век, известен с техниката си за бой с два меча, го казва простичко:



Не се привързвайте към нито
едно оръжие или бойна школа.

- Миямото Мусаши

Обърнете внимание на правилата при всеки един инструмент. Например ако използвате Скръм, но решите да спрете използването на итерации (или който и да е ключов елемент от Скръм), не казвайте, че използвате Скръм. Скръм е достатъчно минималистичен сам по себе си и ако продължите да го наричате Скръм, след като премахнете елемент от него, би било безсмислено и объркващо. Наречете го „Скръм – вдъхновение” или „частичен Скръм”, или какво ще кажете за „ала Скръм” :o)

3

Скръм дефинира роли

Скръм дефинира 3 роли: Продукт Оунър (от англ. Product Owner – *собственик на продукта*, определя визията и приоритетите), Екип (изработва продукта) и Скръм Мастър (от англ. Scrum Master – *„майстор“ на Скръм*, премахва препятствията и отговаря за процеса).

Канбан не дефинира никакви роли.

Това не означава, че не можете да имате Продукт Оунър в Канбан! Означава само, че не сте задължени да имате. И в Скръм, и в Канбан сте свободни да си добавите ролите, от които имате нужда.

Бъдете внимателни при добавянето на роли, уверете се, че допълнителните роли носят стойност и не са в противоречие с други елементи от процеса. Сигурни ли сте, че имате нужда от роля Мениджър на проекта? В голям проект това сигурно е добра идея, ако това е човекът, който помага за синхронизирането на различните екипи и Продукт Оунърите помежду им. В малък проект такава роля е само загуба на ресурси, или още по-лошо, може да доведе до субоптимизация и микромениджмънт.

Основното съзнание и в Скръм, и в Канбан е „по-малкото е повече”. Така че, когато се колебаете, започнете с по-малко.

В останалата част от книгата ще използвам термина „Продукт Оунър”, за да представя ролята на човека, който определя приоритетите на екипа, независимо от процеса, който се използва.

4

Скръм налага итерации, ограничени във времето

В основата на Скръм са итерациите, ограничени във времето. Вие можете да изберете дължината на итерацията, но основната идея е да има повторемост на итерациите, така че след време да се установи някакъв *такт*.

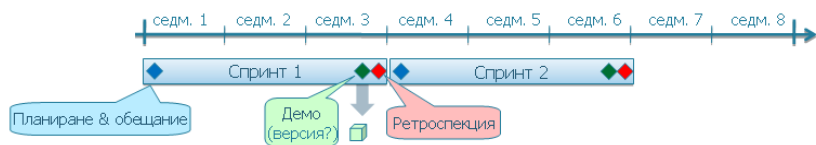
- **Началото на итерацията:** Планът за итерацията е създаден, екипът избира определен брой задачи от продуктивния списък според приоритетите, определени от Продукт Оунъра и според това, колко може да свърши екипа за една итерация.
- **По време на итерацията:** Екипът се фокусира върху изпълнението на поетите задачи. Обхватът на итерацията е фиксиран.
- **Край на итерацията:** Екипът демонстрира работещия код на заинтересованите лица като в идеалния случай кодът е готов за използване от клиента (т.е. тестван и с достатъчно добро качество). След това екипът прави ретроспекция с цел дискусия и оптимизиране на процеса.

И така, една Скръм итерация е един такт, ограничен във времето, комбиниращ 3 дейности: планиране, подобряване на процеса и (в идеалния случай) издадена версия на продукта.

В Канбан итерациите ограничени във времето не са задължителни. Вие можете да изберете кога да правите планиране, подобрения, версия на продукта. Можете да изберете да извършвате тези дейности на регулярен принцип („версия на продукта всеки понеделник”) или при нужда („версия на продукта когато имаме нещо полезно за клиента”)

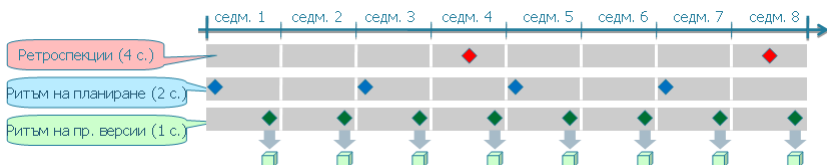
Екип #1 (един такт)

“Ние правим Скръм итерации”



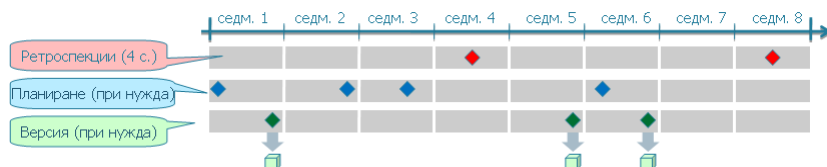
Екип #2 (три такта)

“Ние имаме 3 различни такта. Всяка седмица имаме версия на продукта с това, което е готово. Всяка втора седмица имаме среща за планиране и съответно обновяваме приоритетите си и плана. Всяка четвърта седмица си правим ретроспекция, за да оптимизираме процеса си.



Екип #3 (предимно управляван от събитията)

“Правим среща за планиране, когато стигнем до момента, в който нямаме задачи за вършене. Издаваме версия на продукта, когато имаме набор от минимални „продаваеми“ свойства/функционалности (от англ. *Minimum Marketable Features (MMFs)*). Правим спонтанно цикъл за подобряване на качеството, когато се сблъскаме с един и същи проблем втори път. Също така правим задълбочена ретроспекция всяка четвърта седмица.”



5

Канбан ограничава броят на недовършените задачи чрез лимит на всяка фаза от работния поток, а Скръм – чрез итерация

В Скръм продуктивният списък за спринта съдържа задачите, които трябва да се изпълнят по време на текущата итерация (= „спринт” в Скръм термини). Често това се визуализира чрез карти на стената и се нарича Скръм дъска или дъска за задачи.

И така, каква е разликата между Скръм дъската и Канбан дъската?

Нека да започнем с тривиален и прост проект и да ги сравним:



И в двата случая следим задачите чрез прогреса им в работния поток. Избрали сме три фази: „Да се направи”, „В прогрес” и „Готово”. Вие можете да изберете каквито фази си поискате – някои екипи добавят фази, като например „Интеграция”, „Тестване”, „Издаване на версия”, и т.н. Но не забравяйте принципа *„По-малкото е повече”*.

И така, каква е разликата между тези два примера за дъски. Да – малкото число 2 по средата на колоната на Канбан дъската. Това е всичко. Това 2 означава „Не може да има повече от 2 задачи в тази колона в даден момент”.

В Скръм няма правило, което да пречи на екипа да сложи по едно и също време всички задачи в колоната „В прогрес“! Все пак има

косвено ограничение, тъй като итерацията има фиксирана дължина. В този случай косвеното ограничение е 4, тъй като има общо само 4 задачи на цялата дъска. Така Скръм ограничава броя на недовършените задачи (WIP) косвено, докато Канбан ограничава броя на недовършените задачи директно.

Повечето Скръм екипи се научават, че е лоша идея да има прекалено много започнати и недовършени задачи и изграждат култура, при която първо се довършват текущите задачи преди да се поемат нови. Някои екипи дори решават да ограничат броя на недовършените задачи и тогава – та-да-дааа! – Скръм дъската се превръща в Канбан дъска.

Както се вижда, и Скръм, и Канбан ограничават броя на започнати и недовършени задачи, но по различен начин. Скръм екипът използва понятието „темпо“ – колко задачи (или съответстващите „сюжетни точки“) са завършени за една итерация. Веднъж разбрали темпото си, това е тяхното ограничение за започнати и недовършени задачи (или поне това им е някаква насока). Екип, имащ темпо от по 10 задачи, няма да вземе повече от 10 задачи (или сюжетни точки) за един спринт.

В Скръм броят на започнатите и недовършени задачи (WIP) се ограничава в рамките на единица време.

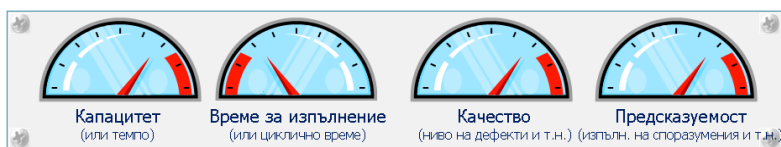
В Канбан броят на започнатите и недовършени задачи (WIP) се ограничава в рамките на всяка фаза от работния поток.

В по-горния пример за Канбан най-много 2 задачи могат да са в работния поток във фаза „В прогрес“ по всяко едно време, независимо от дължината на такта. Вие трябва да изберете какво ограничение да поставите на всяка фаза от работния поток, но основната идея е да ограничите WIP за всичките му положения, започвайки възможно най-рано и свършвайки възможно най-късно в потока. В примера по-горе е добре да помислим за поставяне на ограничение и на фазата „да се направи“ (или както наричате входната опашка). Веднъж имайки ограничение на броя на започнатите и недовършени задачи, ние можем да започнем да измерваме и да предсказваме времето за изпълнение, т.е. средното време, което е необходимо на дадена задача да се придвижи по целия път на дъската. Имайки предвидимо време за завършване, можем да поемем отговорност за споразуменията за нивото на услугите (от англ. *Service level agreements (SLAs)*) и да направим реалистичен план за весриите на продука и тяхното издаване.

Ако размера на задачите варира драстично, можете да помислите за ограничение на количеството започната работа в сюжетни точки или в каквато мерна единица използвате. Някои екипи инвестират време в разбиването на задачите на съизмерими задачи, за да избегнат колебанията и да намалят времето за оценка на нещата (вие дори можете да решите, че оценката е излишна). Във всеки случай е много по-лесно да се направи гладко вървяща система, ако задачите са съизмерими.

6

И двата са емпирични



Представете си, че имате копчета за управление на тези показатели и че можете да си настроите процеса чрез въртене на копчетата. „Искам висок капацитет, кратко време за изпълнение, високо качество и висока степен на предвидимост. За целта ще завъртя копчетата на 10, 1, 10, 10 съответно.”

Нямаше ли да е прекрасно? За съжаление, няма такива директни начини на контролиране. Или поне не такива, за които да знам. Ако намерите начин, моля ви, обадете ми се.

Вместо това ние разполагаме с няколко индиректни начина за контрол.



Скръм и Канбан са емпирични, в смисъла че от вас се очаква да експериментирате с процеса и да го адаптирате към вашите условия. В действителност вие *трябва* да експериментирате. Нито Скръм, нито Канбан дават всички отговори - те просто ви дават един базов набор от ограничения, с които да управлявате подобренията на процесите си.

- Скръм казва, че трябва да имате мултифункционални екипи. И така – кой в кой екип ще бъде? Не знам – експериментирайте.
- Скръм казва, че екипът избира колко работа да вземе за спринта. И така – колко точно работа трябва да вземат? Не знам, експериментирайте.
- Канбан казва, че трябва да ограничите броят на започнати и недовършени задачи. И така – какво трябва да е ограничението? Не знам – експериментирайте.

Както споменах по-рано, Канбан налага по-малко ограничения от Скръм. Това означава, че имате повече параметри, по които да мислите, повече копчета, които да въртите. Това може да е както предимство, така и недостатък, в зависимост от контекста. Когато отворите конфигурационен диалог на някое приложение, дали предпочитате да имате 3 възможности за настройки или 100? Може би нещо по средата. Зависи от това колко обичате да настройвате и колко добре разбирате приложението.

И така, нека да кажем, че намалим броя на недовършените задачи, базирайки се на хипотезата, че това ще подобри нашия процес. Тогава наблюдаваме как неща като капацитет, време за изпълнение, качество и предвидимост се променят. Правим изводи от резултатите и после променяме още неща, постоянно подобрявайки процеса.

Има много имена за това действие на подобряване. Kaizen (постоянно подобрене в термините на Лийн), Инспектирай и адаптирай (в термините на Скръм), Емпиричен контрол на процеса или защо не Научният метод.

Най-критичният елемент е цикълът на *обратната връзка*. Промени нещо => Виж как се случва => Научи се от него => Промени нещо отново. Нормално е вие да искате възможно най-кратък цикъл на обратната връзка, за да адаптирате процеса си бързо.

В Скръм основния цикъл на обратната връзка е спринтът. Разбира се има и други, например ако комбинирате с Екстремно програмиране (XP).



Когато се прави коректно, Скърм + Екстремно програмиране ви поднася букет от изключително полезни цикли за обратна връзка.

Вътрешният цикъл за обратна връзка, програмирането по двойки, е даване на обратна връзка в период от няколко секунди. Дефекти се откриват и оправят в рамките на секунди („Хей, не трябва ли тази променлива да е 3“). Това е цикъл на обратна връзка от типа „наистина ли правим нещата **правилно**?”.

Външният цикъл на обратната връзка, спринтът, има продължителност от няколко седмици. Този цикъл е от типа „наистина ли правим **правилните** неща?”.

Какво да кажем за Канбан? Разбира се, на първо място вие можете (и най-вероятно трябва) да поставите описаните цикли на обратна връзка във вашия процес, независимо дали използвате Канбан. Канбан ви дава допълнително няколко много полезни метрики:

- Средна продължителност на цикъл (cycle time или lead time). Обновява се всеки път, когато задача достигне до фаза „Завършено” (или както и да сте кръстили крайната си дясна колона).
- Затруднения („тесни места“, bottlenecks). Типичен симптом е, когато колона X е препълнена със задачи, докато колона X+1 е празна. Потърсете празнини на вашата дъска.

Хубавото нещо на метриките в реално време е, че вие можете сами да решите каква е дължината на цикъла на обратната връзка, в зависимост от това колко често искате да анализирате метриките и да правите промени. Прекалено дълги цикли водят до бавно подобряване на процеса. При прекалено кратки цикли може да не е

имало достатъчно време, за да се стабилизира вашият процес между промените, което може да доведе до тъпчене на едно място.

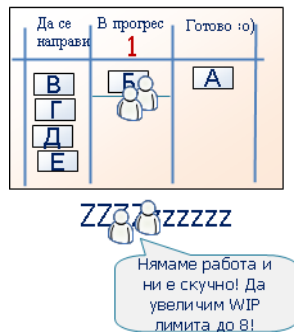
Всъщност, дължината на цикъла на обратна връзка е едно от нещата, с които можете да експериментирате... нещо като мета-цикъл за обратна връзка.

Добре, ще спра.

Пример: Експериментиране с ограниченията на броя недовършени задачи в Канбан

Един от типичните аспекти на Канбан, които могат да се настройват, е ограничаването на броя недовършени задачи. Е, как да разберем дали го правим правилно?

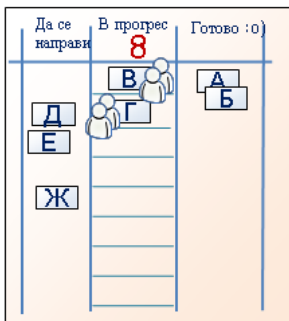
Нека да предположим, че имаме екип от 4 човека и решаваме да започнем с ограничение на броя на започнати и недовършени задачи равен на 1.



Когато започнем работа по една задача, не можем да започнем нова, докато първата не е готова. Е, тя ще е готова наистина бързо.

Супер! Но тогава става ясно, че не е много разумно всичките четирима човека да работят по една и съща задача (в този прост случай) и имаме хора, които просто безделничат. Ако това се случва рядко, не е голям проблем, но ако се случва редовно, последствието е, че средното време за изпълнение ще се увеличи. В действителност, ограничаване на броят на започнати и недовършените задачи с 1 означава, че дадена задача бързо ще премине през фазата „В прогрес“, но задачите ще заседнат във фазата „Да се направи“ по-дълго от необходимото, така че общото време за изпълнение за целия работен поток ще е ненужно високо.

Е, ако 1 е прекалено малко за ограничение за текущата работа, какво ще кажете да го увеличим на 8.

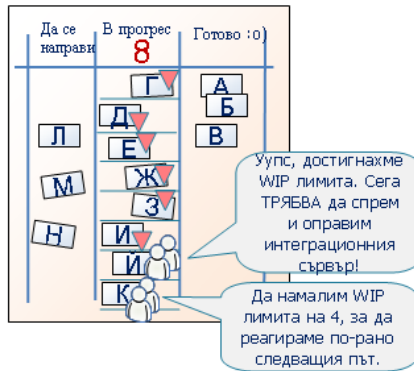


Това работи по-добре за известно време. Разбираме, че средно работата по двойки допринася работата да бъде свършена най-бързо. И в екип от 4 души, обикновено имаме 2 задачи, по които се работи в даден момент. Ограничението за броя недовършени задачи от 8 беше просто горен лимит, така че няколко започнати задачи по-малко е в реда на нещата

Сега си представете, че се натъкнем на проблем с интеграционния сървър и не можем да приключим напълно с която и да е задача (определението ни за „Готово“ включва интеграция). Понякога се случват и такива неща, нали?



При условие, че не можем да довършим задачи Г и Д, започваме работа по Е. Но не можем да интегрираме и Е, така че вземаме нова задача Ж. Скоро достигахме нашия лимит – 8 задачи във фаза „В процес“.



В този момент вече не можем да взимаме повече задачи. Ей, по-добре да оправим този ужасен интеграционен сървър. Ограничението за броя на недовършените задачи ни накара да реагираме и да оправим затрудненията, вместо просто да генерираме куп недовършена работа.

Дотук добре. Но ако ограничението на броя недовършени задачи беше 4, щяхме да реагираме много по-бързо, което щеше да доведе до по-добро средно време за изпълнение. Всичко е въпрос на баланс. Измерваме средното време за изпълнение и оптимизираме лимита на на броя недовършени задачи, за да оптимизираме времето за изпълнение:



След време може да установим, че някои задачи се задържат във фаза „Да се направи“. Може би е време да поставим ограничение на броя задачи и там.

Защо въобще имаме нужда от колона „Да се направи“? Ако клиентът е винаги налице, за да каже на екипа какво следва, винаги когато те попитат, тогава от колоната нямаше да има нужда. Но в случаите, когато клиентът невинаги е налице, колоната „Да се направи“ дава на екипа малък буфер със задачи от които да избира.

Експериментирайте! Или както казват идеолозите на Скърм „Инспектирай и адаптирай”!

7

Скърм се противопоставя на промяната по време на итерация

Нека приемем, че Скърм дъската ни изглежда така:



Какво ще стане, ако някой се появи и пожелае да добави Д на дъската?

Обикновено Скърм екипът ще каже нещо като „Не, съжаляваме, но ние сме обещали да изпълним А+Б+В+Г в този спринт. Но можете да добавите Д към продуктивния списък. Ако Продукт Оунърът прецени, че е с висок приоритет, ще го вземем в следващия спринт”. Спринтовете с правилната дължина дават на екипа достатъчно фокусирано време, за да създаде нещо работещо, и все пак позволяват на Продукт Оунъра да променя приоритетите регулярно между итерациите.

Е, какво ще каже в този случай Канбан екипът?



Канбан може да каже „Свободен си да добавиш Д към колоната „да се направи”. Но лимитът на тази колона е 2, така че в такъв случай ще трябва да премахнеш или В, или Г. Работим по А и Б в момента, но щом освободим капацитет, ще вземем най-горната задача от колоната „Да се направи”.

В този случай времето за реакция (колко време отнема да се откликне на промяна на приоритетите) на Канбан екипа е точно толкова, колкото дълго е освобождаването на капацитет. Следва се основният принцип „Една задача влиза, една излиза” (управляван от ограничаването на броя недовършени задачи).

В Скръм средното време за реакция е половината от дължината на спринта.

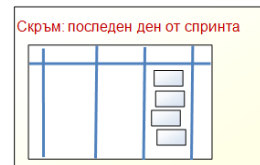
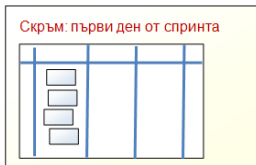
В Скръм Продукт Оунърът не може да променя Скръм дъската, след като екипът е обещал да свърши определен набор от задачи за една итерация. В Канбан вие трябва сами да си определите правила за това кой може да променя по дъската. Обикновено на Продукт Оунъра се дава една колона най-вляво, наречена „Да се направи”, „В готовност”, „Продуктов списък” или „Предложения”, където той може да прави промени когато и както си пожелае.

Тези два подхода, обаче, не се изключват взаимно. Скръм екипът може да реши да позволи на Продукт Оунъра да промени приоритетите по време на спринта (въпреки че обикновено това се счита за изключение). И Канбан екипът може да реши да постави ограничения относно това кога приоритетите могат да се променят. Канбан екипът даже може да реши да използва ограничени по време предефинирани итерации, точно както в Скръм.

8

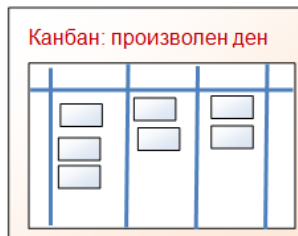
Скръм дъската се изчиства преди всяка итерация

Скръм дъската обикновено изглежда подобно на тези по време на различните фази от спринта.



Когато спринтът свърши, дъската се изтрива – всички задачи се премахват. Започва нов спринт и след планирането му имаме нова Скръм дъска с нови задачи в най-лявата колона. Технически погледнато това е загуба, но за един опитен Скръм екип това не отнема много време, а процесът на изтриване на всичко от дъската дава приятното усещане за успешно изпълнение и приключване. Нещо като измиването на чиниите след вечеря – миенето е мъка, но след това усещането е приятно.

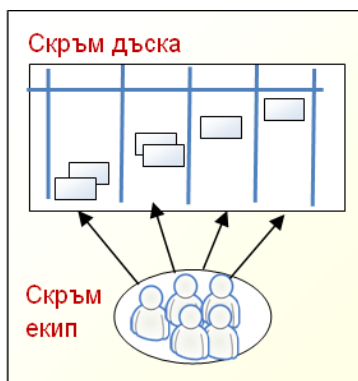
В Канбан е нормално дъската да е постоянна – няма нужда да я изтривате и да започвате наново.



9

Скръм препоръчва мултифункционални екипи

Скръм дъската е притежание на точно един Скръм екип. Скръм екипът е мултифункционален, съдържащ всички умения, нужни за завършването на задачите в итерацията. Скръм дъската се вижда от всички заинтересовани, но може да се обновява само от Скръм екипа, който я притежава – това е техният начин да управляват обещаното за текущата итерация.

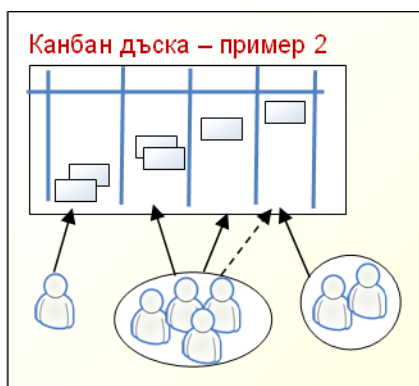


В Канбан мултифункционалните екипи не са задължителни и дъската не е нужно да се притежава само от един определен екип. Дъската се отнася към един работен поток, а не задължително към един екип.

Ето два примера:



Пример 1: Цялата дъска се обслужва от един мултифункционален екип. Точно като в Скръм.



Пример 2: Продукт Оунърът определя приоритетите в колона 1. Мултифункционалният екип разработва (колона 2) и тества (колона 3). Издаването на версия (колона 4) се прави от специализиран екип. Има известно припокриване на компетенциите, така че ако екипът по издаване се затрудни, някой от разработчиците да му помогне.

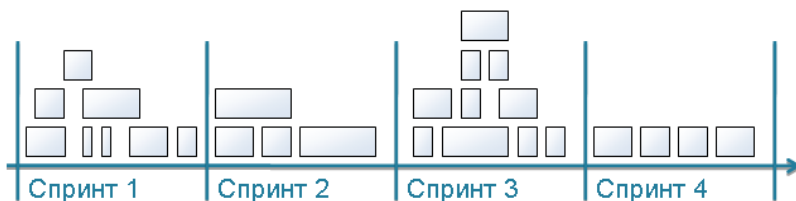
И така, в Канбан имате нужда да установите някакви правила за това кой използва дъската и как, после да експериментирате с правилата, за да оптимизирате начина на работа.

10

Списъкът със задачи в Скръм трябва да се побере в спринта

И Скръм, и Канбан са базирани на инкрементална разработка, т.е. разбиване на малки парчета. (Итерация = Спринт (Б.П.))

Един Скръм екип ще обещае да свърши точно толкова задачи, колкото мисли, че може да завърши в рамките на една итерация (според дефиницията за „Готово“). Ако някоя задача е прекалено голяма, за да влезе в спринт, екипът и Продукт Оунърът ще се опитат да я разбият на парчета, докато се вмести в 1 спринт. Ако в повечето случаи задачите са големи, итерациите са по-дълги (въпреки че обикновено не са повече от 4 седмици).



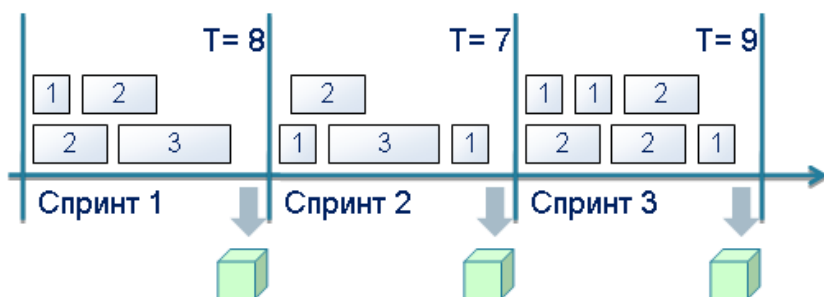
Канбан екипите се опитват да минимизират времето за изпълнение и нивото на потока и това индиректно се превръща в стимул задачите да се разбиват на сравнително малки парчета. Но няма изрично правило, според което задачите трябва да са достатъчно малки, за да се впишат в определено време. На една и съща дъска можем да имаме една задача, която отнема месец, и друга, която отнема 1 ден.



11

Скръм изисква оценяване и темпо

В Скръм се очаква екипите да оценяват в относителни размери (=количество работа) всяка задача, която обещават да изпълнят. Чрез събирането на размерите на всички изпълнени задачи в края на спринта, ние получаваме темпото. То е мярка за капацитета му – колко работа може да свърши за един спринт. Ето пример за екип, чието темпо е 8.



Добре е да се знае, че средното темпо е 8, защото тогава могат да се правят реалистични прогнози за това кои задачи могат да се изпълнят в предстоящия спринт.

В Канбан оценяването не е задължително. Ако се наложи да се дават обещания, трябва да се прецени как да се подsigури предвидимост.

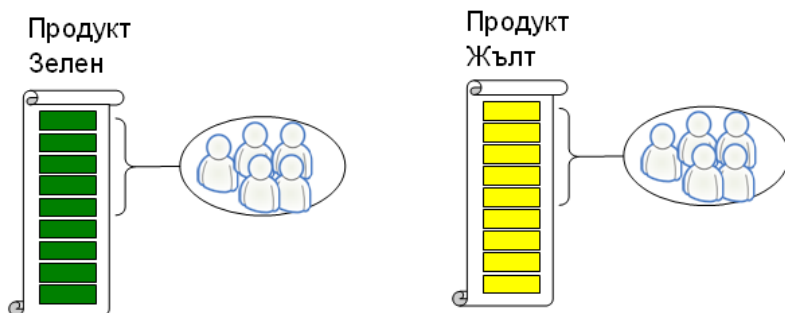
Някои екипи избират да оценяват и да измерват темпо, като в Скръм. Други екипи решават да пропуснат оценяването, но се опитват да разделят всяка задача на парчета с относително еднакъв размер – тогава те могат да измерват темпото си в термините на това колко задачи са изпълнени за единица време (например брой функционалности за седмица). Някои екипи групират задачите в минимални „продаваеми“ функционалности (MMF), измерват средното време за изпълнение на една такава и използват полученото, за да установят споразумения за нивото на услугите (SLA) – например, „Когато обещаем MMF, тя винаги ще е готова след 15 дни.“

Има различни видове интересни техники за продуктово планиране и обещания „ала Камбан“, но никоя конкретна техника не се препоръчва задължително. Така че – давайте, търсете и опитвайте нови различни техники, докато намерите онази, която ви пасва. Най-вероятно ще видим някои добри практики, които ще се наложат с времето.

12

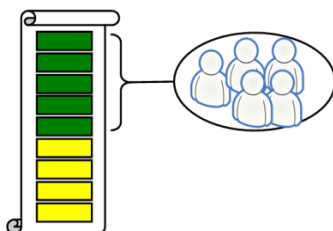
И двата позволяват работа по няколко продукта паралелно

В Скръм *продуктов списък* не е съвсем уместно име, тъй като предполага, че всички изисквания са за един продукт. Ето два продукта, зелен и жълт, всеки със собствен списък с изисквания и със собствен екип.

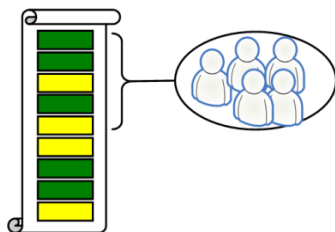


Какво ще правите, ако имате само един екип? Е, мислете за продуктовия списък по-скоро като за екипен списък. Той съдържа всички приоритети за предстоящите итерации за един конкретен екип (или набор от екипи). Ако екипът поддържа няколко продукта, смесете двата продукта в един списък. Това ни принуждава да приоритизираме между продуктите, което е полезно в дадени случаи.

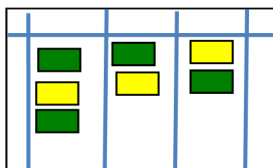
Има няколко начина да се направи това на практика. Едната стратегия би била екипът да се фокусира на един продукт в един спринт.



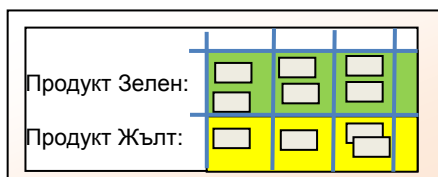
Друга стратегия би била екипът да работи по изисквания по двата продукта във всеки един спринт.



Същото е и в Канбан. Можем да имаме няколко продукта на една дъска. Можем да ги различаваме, използвайки различни по цвят карти.



... или чрез т.нар. „коридори“:



13

И двата са Лийн и Аджайл

Нямам намерение да минавам през Лийн мисленето и през Манифеста на гъвките методологии (Agile Manifesto), но най-общо казано и Скръм и Канбан се вписват в тези ценности и принципи. Например:

- И Скръм, и Канбан са системи за планиране с изтегляне (Pull system), които отговарят на Лийн принципа „точно навреме“ (Just in Time) за управление на наличностите. Това означава, че екипът избира кога и колко работа да обещае, той „изтегля“ задачи, когато е готов, а не ги получава отвън. Точно както един принтер издърпва следващия лист хартия, когато е готов да печата върху него (въпреки че има малък и ограничен куп хартия, от който може да си вземе).
- Скръм и Канбан са базирани на непрекъсната и емпирична оптимизация на процесите, която съответства на принципа за постоянно подобрене Кайзен (Kaizen) от Лийн.
- Скръм и Канбан наблягат на реакцията на промените пред стриктното следване на плана (въпреки че Канбан позволява по-бърза реакция от Скръм), това е един от четирите принципа на Манифеста на гъвките методологии.

... и така нататък.

От една страна Скръм може да се счита за по-малко лийн, защото задължава групиране на задачи в итерации с фиксирано време. Но това зависи от дължината на вашата итерация и това, с което я сравнявате. В сравнение с традиционния процес, където вероятно интегрираме и издаваме версия на продукта от 2 до 4 пъти в годината, Скръм екипът произвежда работещ код на всеки 2 седмици, което е доста лийн.

Но тогава, ако продължавате да правите итерациите по-кратки и по-кратки, вие се доближавате до Канбан. Когато започнете да

коментирате итерации, по-кратки от 1 седмица, можете да помислите за премахване на фиксираните итерации въобще.

Казах го по-рано и ще продължавам да го казвам: експериментирайте, докато намерите това което работи за вас! И после продължете да експериментирате :o)

14

Малки разлики

Ето някои разлики, които изглеждат по-маловажни в сравнение с нещата, споменати по-рано. Но все пак е добре да сте запознати с тях.

При Скръм е задължителен приоритизираният продуктов списък

В Скръм имаме винаги приоритизиран и сортиран продуктов списък, а промените по приоритетите могат да станат чак в следващия, а не в текущия спринт. В Канбан вие можете да изберете какъвто и да е начин за приоритизация (дори нищо) и промените се случват в момента, когато има свободен капацитет, а не на фиксирано време. Може да има или да няма продуктов списък и той може да е или да не е приоритизиран.

На практика, няма голямо значение. На Канбан дъската най-лявата колона обикновено върши същата работа като продуктовия списък в Скръм. Независимо дали списъкът е сортиран по приоритет, екипът има нужда от правила за това коя задача да изтегли най-напред. Примери за правила:

- Винаги взимайте най-горната задача.
- Винаги вземайте най-старата (всяка задача е белязана с време на постъпване).
- Вземете коя да е задача.
- Изкарвайте приблизително 20% в задачи по поддръжка и 80% за нови функционалности.
- Разделете капацитета на екипа грубо на две между продукт А и продукт Б.
- Винаги взимайте червени задачи, ако има такива.

В Скръм продуктивният списък може да се използва „ала Канбан“. Можем да ограничим размера му и да създадем правила за това как трябва да се приоритизира.

При Скръм са задължителни ежедневните срещи

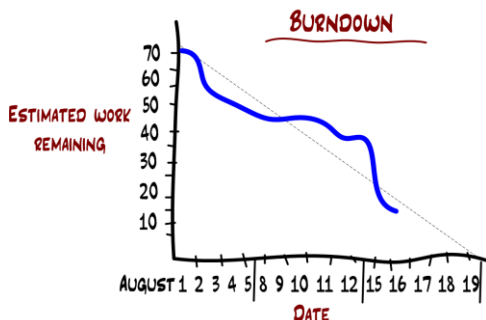
Един Скръм екип има кратка среща (максимум 15 минути) всеки ден, по едно и също време, на едно и също място. Целта на срещата е да се обмени информация за това какво се случва, да се планира работата за текущия ден и да се идентифицират сериозни проблеми. Понякога тази среща се нарича правостояща, тъй като обикновено участниците стоят прави (с цел да се запази кратка и да се поддържа високо нивото на енергия).

Ежедневните срещи не са задължителни в Канбан, но въпреки това, повечето Канбан екипи ги правят. Това е чудесна техника, независимо кой процес използвате.

В Скръм форматът на срещата е насочен към хората – всички споделят един по един. Много Канбан екипи използват формат, насочен към дъската, фокусирайки се върху затрудненията и други очевидни проблеми. Този подход позволява мащабиране. Ако имате 4 екипа, които споделят една дъска и имат обща ежедневна среща, може да не е необходимо всеки да се изказва, тъй като фокусът е върху затрудненията на дъската.

При Скръм е задължителна графиката на оставащата работа

Графиката на оставащата работа за спринта ежедневно показва колко работа остава до края на текущата итерация.



Мерната единица на вертикалната ос е същата като мерната единица, която се използва за задачите от спринта. Обикновено часове или дни (ако екипът развива елемент от продуктивния списък за спринта

на задачи) или сюжетни-точки (ако екипът не го прави). Има много вариации по темата.

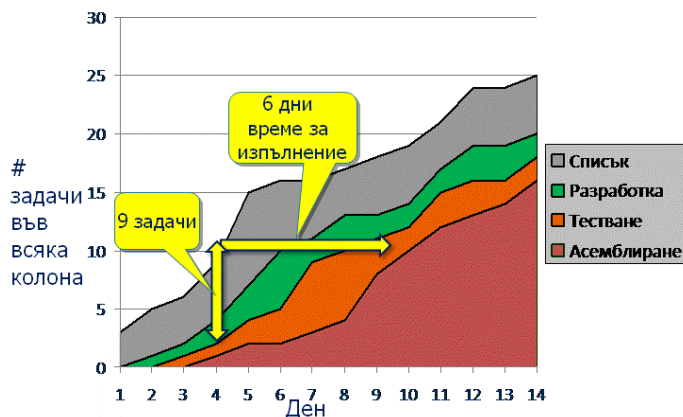
В Скръм, графиката на оставащата работа за спринта е един от основните инструменти за следене на прогреса в итерацията.

Някои екипи използват графика на оставащата работа за версията на продукти, която използва същия формат, но се отнася за версията – обикновено показва колко сюжетни-точки са останали в продуктивния списък след всеки спринт.

Основната цел на графиката на оставащата работа е лесно да се установи възможно най-рано дали сме изостанали от графика, за да можем да се адаптираме.

В Канбан графиката на оставащата работа не е задължителна. На практика никой тип графика не е задължителен. Но, разбира се, вие можете да ползвате какъвто и да е тип графика по ваше желание. (включително и графиката на оставащата работа).

Ето пример за диаграма на кумулативните потоци. Този тип графика показва колко гладко върви потокът на работата ви и доколко броя недовършените задачи влияе на времето за изпълнение.



Ето как работи. Всеки ден събирайте броя на задачите във всяка колона от Канбан дъската и го слагайте по вертикалната ос. На четвъртия ден ще имаме 9 задачи на дъската. Започвайки от най-дясната колона, има 1 задача в Асемблиране, 1 задача в Тестване, 2 задачи в Разработка, 5 задачи в Продуктов списък. Ако визуализираме точките ежедневно, ще получим чудесна диаграма

като тази по-горе. Вертикалните и хоризонтални стрелки илюстрират релацията между броя недовършени задачи и време за изпълнение.

Хоризонталната стрелка показва, че има задачи, поставени в продуктивния списък на четвъртия ден при средни 6 дена до достигане на фазата асемблиране. Около половината от времето е било за тестване. Това, което виждаме е, че ако бяхме намалили ограничението на броя недовършени задачи за фазите тест и продуктивния списък, съществено щеше да се намали общото време за изпълнение.

Наклонът на тъмночервената област показва темпото (т.е. броя на елементите, инсталирани за ден). С времето се забелязва, че високото темпо намалява средната продължителност на цикъла, а високото ограничение на броя недовършени задачи го увеличава.

Повечето организации искат да получат нещата завършени възможно най-бързо (= да се намали времето за изпълнение). За съжаление, доста от тях попадат в капан, тъй като предполагат, че това ще се случи, ако въвлекат повече хора или чрез работа извън работно време. Обикновено най-ефективният начин да се свърши работата максимално бързо е да се изглади потока на работа и да се ограничи работата според капацитета, а не да се добавят повече хора или да се работи по-усилено. Такъв тип диаграма показва защо, и това увеличава възможността екипът и мениджмънтът да си сътрудничат ефективно.

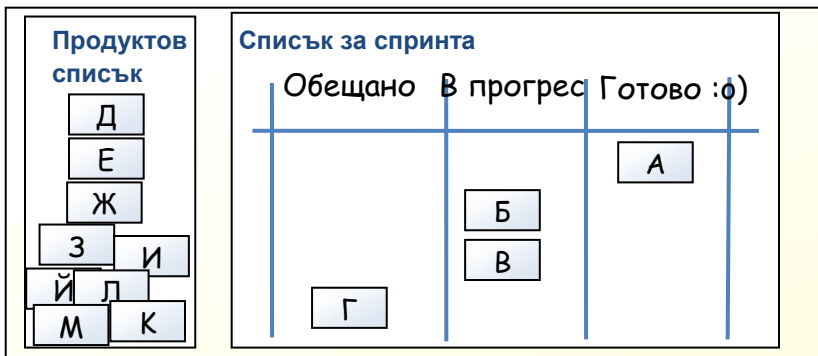
Още по-ясно става, ако разграничим състоянията на изчакване в опашките (например „Чака за тест“) и активните състояния (такива като „Тестване“). Искаме да минимизираме броя задачи, които изчакват обработка в опашките, а графиката на кумулативните потоци предоставя необходимата мотивация за това.

15

Скръм дъска срещу Канбан дъска – нетривиален пример

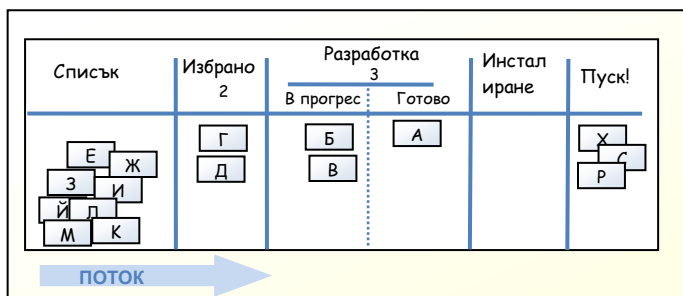
В Скръм списъкът с изисквания за спринта е само една част от картинката – онази част, по която екипът работи по време на текущия спринт. Другата част е продуктивният списък – списъкът с неща, които Продукт Оунърът иска да има в следващи спринтове.

Продукт Оунърът може да вижда, но не може да променя списъка с изисквания за спринта. Той може да променя продуктивния списък когато пожелае, но промените не се вземат предвид до следващия спринт.



Когато спринтът е приключил, екипът „доставя продукт, потенциално готов за ползване” на Продукт Оунъра. И така екипът приключва спринта, прави преглед на постигнатото в спринта и с гордост демонстрира на Продукт Оунъра новите функционалности А, Б, В и Г. Продукт Оунърът тогава решава дали продукта може да се предостави на клиенти или не. Последната фаза – самото предоставяне на продукта, не е част от спринта и затова не се отбелязва в списъка с изисквания за спринта.

За този сценарий Канбан дъската би могла да изглежда така:



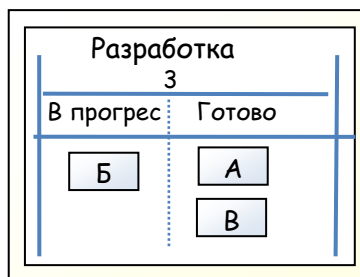
Сега целият работен поток е на една и съща дъска – не просто да се наблюдава какво прави един Скръм екип в една итерация.

В примера по-горе колоната „Списък” е просто списък, без специална подредба. Колоната „Избрано” с ограничение 2 съдържа най-приоритетните задачи. В такъв случай може да има само 2 високоприоритетни задачи в даден момент. Когато екипът е готов да поеме нова задача, те вземат най-горната от „Избрано”. Продукт Оунърът може да прави промени в „Списък” и „Избрано” по всяко време, когато поиска, но не и в останалите колони.

Колоната „Разработка” с ограничение 3 (разделена на две под-колони) показва какво се разработва в момента. В терминологията на мрежите ограничението на броя недовършени задачи съответства на „ширина на лентата”, а времето за изпълнение съответства на „пинг” (или време за отговор).

Защо сме разделили колоната „Разработка” на две по-малки „В прогрес” и „Готово”? Целта е да е ясно на екипът за асемблиране кои задачи могат да се изтеглят във фазата за асемблиране.

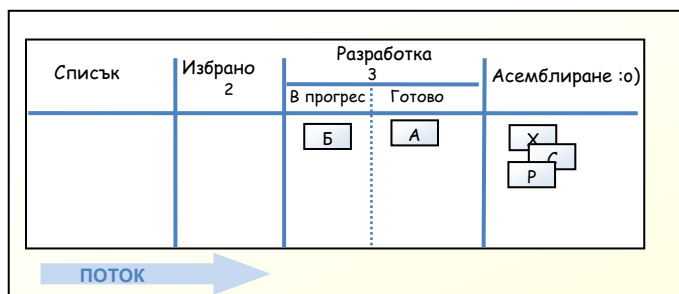
Ограничението 3 за колоната „Разработка” е общо за двете под-колони. Защо? Нека да предположим, че 2 задачи са „готови”:



Това означава, че само една задача може да е в колона „В прогрес”. А това означава, че ако човек вземе нова задача, ще се превиши поставеното ограничение. Това стимулира екипа да се фокусира и да помогне свършената работа да стигне до асемблиране, да изчисти колоната „Готово” и да оптимизира потока. Ефектът е чудесен – повече задачи в „Готово”, по-малко задачи, позволени в колоната „В прогрес”, което помага на екипа да се фокусира върху правилните неща.

Едноелементен поток

Едноелементният поток (one-piece flow) е един вид „перфектен сценарий”, при който една задача преминава през дъската, без никъде да изчакава в опашка. Това означава във всеки един момент да има някой, който да работи по тази задача. Ето как може да изглежда дъската в този случай:

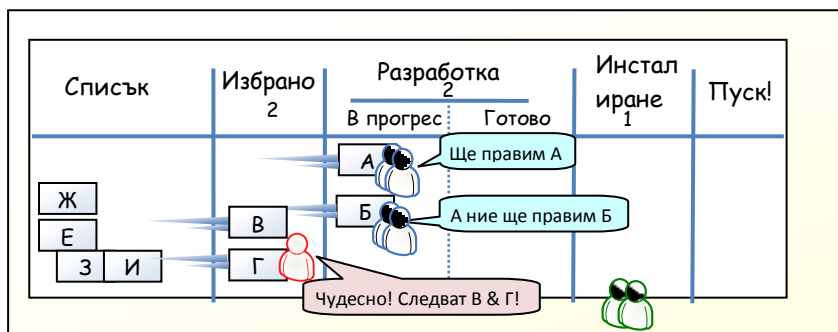
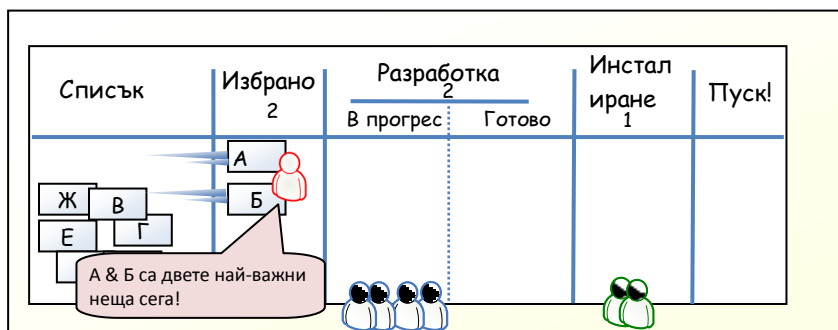
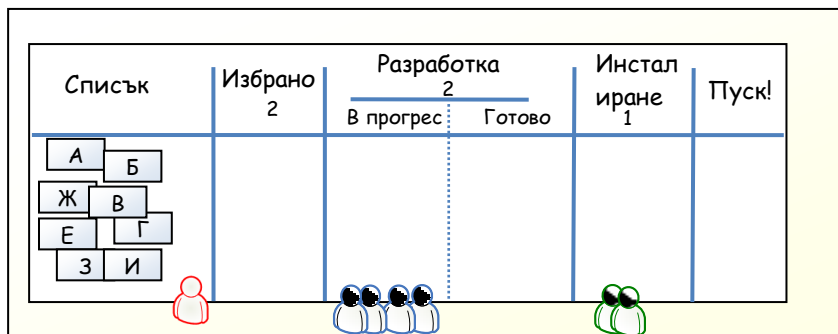


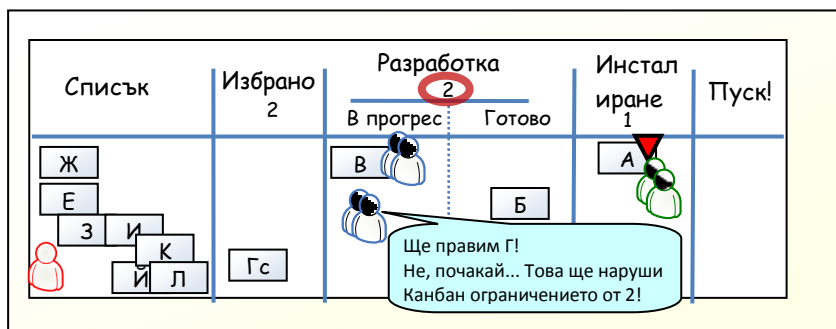
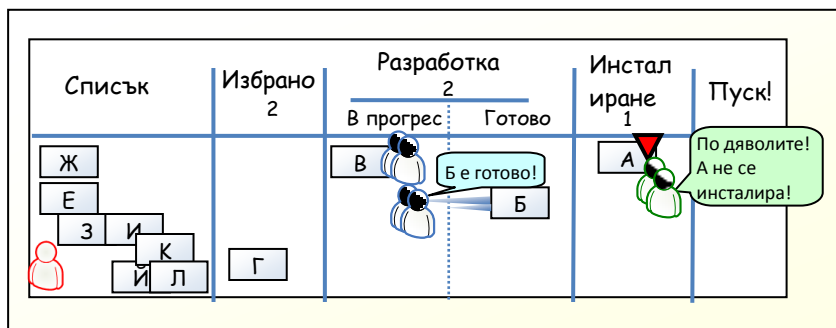
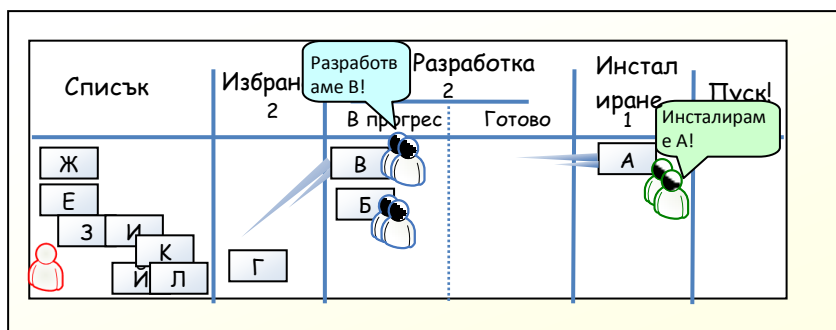
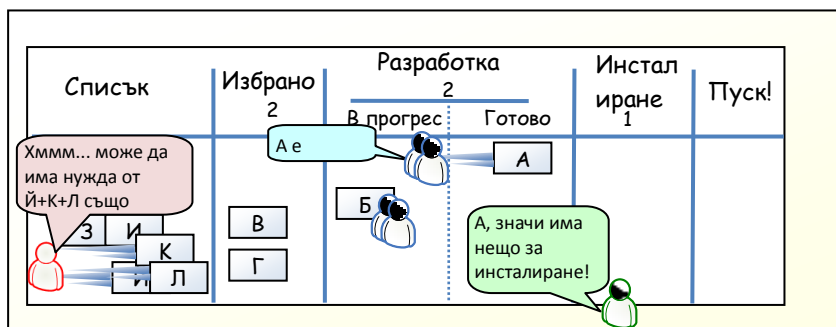
Б се разработва в този момент, а **А** се слага за асемблиране. В момента, в който екипът е готов за следваща задача, те питат Продукт Оунъра кое е най-важно и получават мигновен отговор. Ако този идеален сценарий се запази, можем да се лишим от двата списъка – „Списък” и „Избрано”, и можем да получим наистина кратко време за изпълнение.

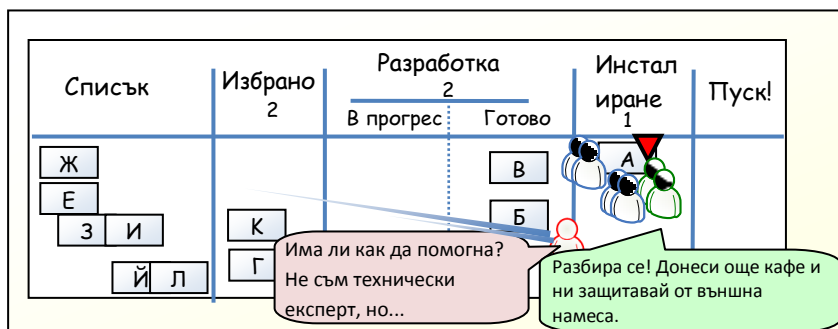
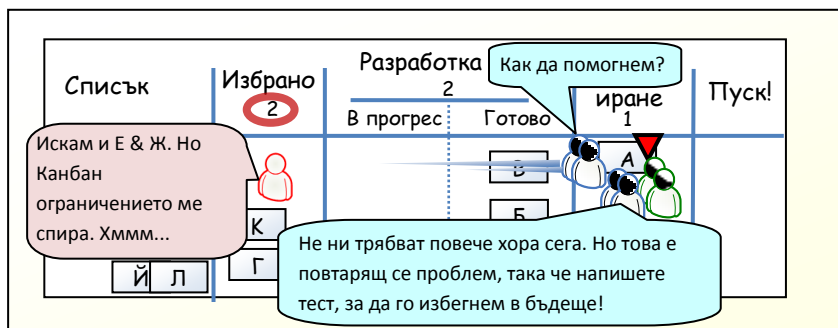
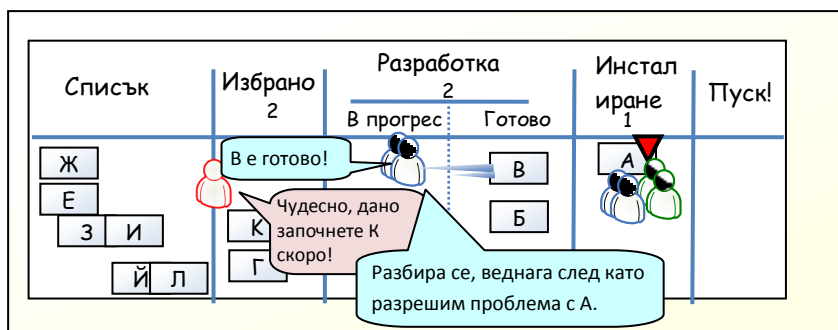
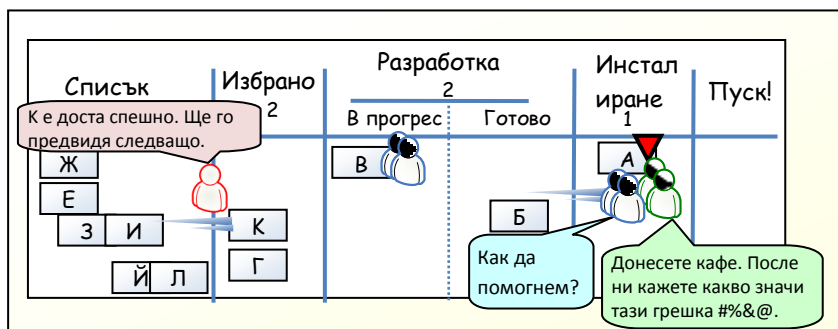
Кори Ладас го описва много добре: „Идеалният процес на работа трябва винаги да предоставя на екипа за разработка най-добрата следваща задача, върху която да продължи да работи, нито повече, нито по-малко.”

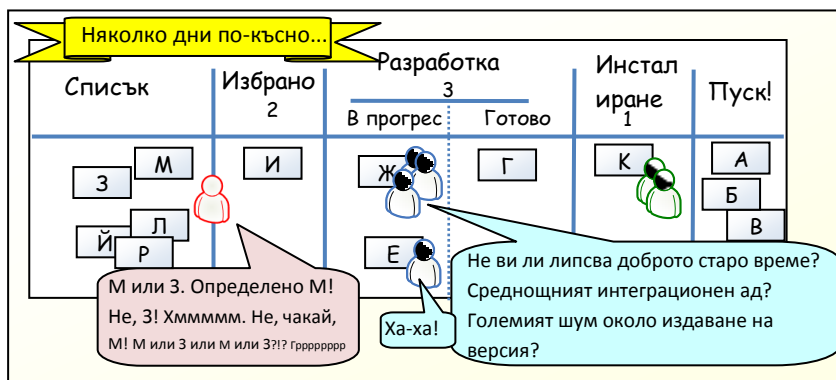
Ограниченията за броя недовършени задачи са само, за да не излязат нещата извън контрол, така че ако нещата вървят гладко, няма нужда от тези ограничения.

Един ден в страната на Канбан









Трябва ли Канбан дъската да изглежда така?

Не, дъската по-горе е само пример.

Единственото нещо, което Канбан задължава, е потокът на работа да е визуален и броят недовършени задачи да е ограничена. Целта е да се създаде гладък процес и да се минимизира времето за изпълнение. За целта е необходимо редовно да си задавате въпроси като:

Какви колони трябва да имам?

Всяка колона визуализира една фаза от потока на работа или буфер между две фази. Започнете с прости колони, добавяйте колони, ако е необходимо.

Какви трябва да са ограниченията?

Когато ограничението на „вашата“ колона бъде достигнато и вие нямате какво да правите, огледайте се за затруднения по трасето (т.е. задачите, намиращи се вдясно на дъската) и се опитайте да помогнете. Ако няма затруднения, това може да означава, че ограничението за тази колона е твърде ниско (причината за ограниченията е да се намали риска от затруднения по трасето).

Ако забележите задачи, по които дълго време не се работи, това означава, че ограниченията може би са твърде високи.

- Прекалено ниско ограничение => хора, стоящи без работа=> лоша продуктивност
- Прекалено високо ограничение=> задачи, стоящи, без да се работят => лошо време за изпълнение

Колко стриктни са ограниченията?

Някои екипи ги приемат за стриктни правила (т.е. екипът не може да превиши ограничението), някои екипи ги приемат за препоръка или като провокация за дискусия (т.е. нарушаването на ограниченията е позволено, но трябва да е премерено и поради конкретна причина).
Още веднъж искам да повтора – зависи изцяло от вас. Казах ви, че Канбан не е много задължаващ, нали?

16

Обобщение на сравнението на Скръм с Канбан

Прилики

- И двата са Лийн и Аджайл.
- И двата използват система на изтегляне за организация на работата.
- И двата ограничават броя на започнати и недовършени задачи.
- И двата използват прозрачност, за да има подобрения в процеса.
- И двата се фокусират върху доставянето на работещ софтуер рано и често.
- И двата се основават на самоорганизиращи се екипи.
- И при двата има разбиване на работата на парчета.
- И при двата планът за версията на продукта се оптимизира постоянно на базата на емпирични данни (темпо/време за изпълнение).

Разлики

Скръм	Канбан
Ограничените по време итерации са задължителни.	Ограничените по време итерации са опционални. Може да има отделни тактове за планиране, изкарване на версия и подобрене на процеса. Може да е провокиран от събитията, вместо от ограничения във времето.
Екипът обещава да свърши определена работа в дадена итерация.	Обещаването на определена работа за определено време не е задължително.
Използва се темпо като метрика за планиране и за подобрения в процеса.	Използва се време за изпълнение като метрика за планиране и за подобрения в процеса.
Задължително е да има мултифункционални екипи.	Мултифункционалните екипи са опционални. Позволен са специализирани екипи.
Работата трябва да се разбие на парчета , така че да бъдат завършени в един спринт.	Няма препоръка за определен размер на задачите.
Задължително е използването на графика на оставащата работа.	Няма диаграма или графика, която да е необходимо да се ползва.
Ограничаване на броя недовършени задачи косвено (за спринт)	Ограничаване на броя на недовършени задачи директно (за фаза в потока на работа)
Оценяването е задължително.	Оценяването не е задължително.
Не може да се добавят задачи по време на итерация.	Може да се добавят задачи винаги, когато има свободен капацитет.

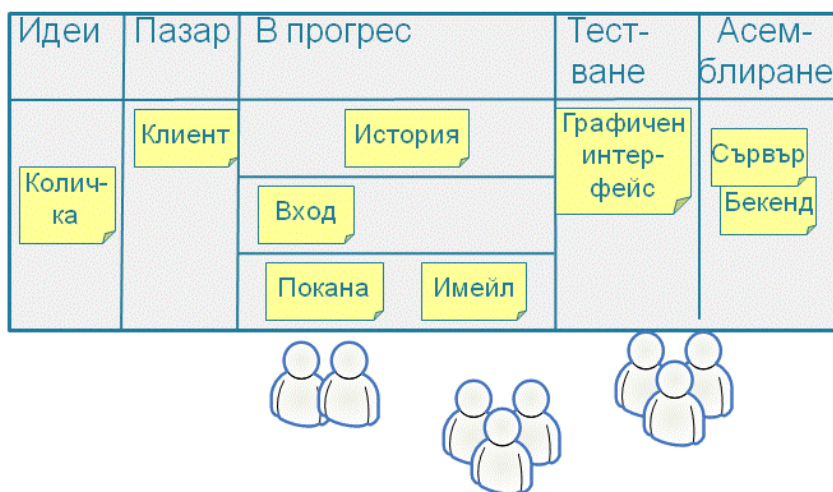
Продуктивният списък за спринта е собственост на един екип.	Дъската на Канбан може да се споделя от различни екипи или хора.
Има 3 задължителни роли (Продукт Оунър, Скръм Мастър и Екип)	Няма никакви задължителни роли.
Скръм дъската се изчиства между спринтовете.	Канбан дъската е постоянна.
Задължителен е приоритизираният продуктов списък.	Приоритизацията е опционална.

Ето. Това е. Сега знаете кои са разликите.

Но това не е всичко, предстои най-добрата част! Обуйте си обувките, време е да скочим „в окопите” с Матиас и да видим как всичко това се прилага на практика!

Част 2 – Пример от практиката

Канбан в реалния живот



Тази история е за това как се научихме да се усъвършенстваме, използвайки Канбан. Когато започнахме, нямаше много информация и д-р Гугъл за първи път не ни помогна. Днес Канбан се развива успешно и знанията по темата се трупат. Например силно ви препоръчвам работата на Дейвид Андерсън по въпроса за „класовете услуги“. Тук трябва да направя първата (и последната) уговорка (обеждавам!). Каквото и решение да приложите, уверете се, че то адресира вашия специфичен проблем. Това беше, край. Хайде да се впускате в темата. И така, ето я нашата история.

/ Матиас Скарин /

17

Природата на администраторската работа

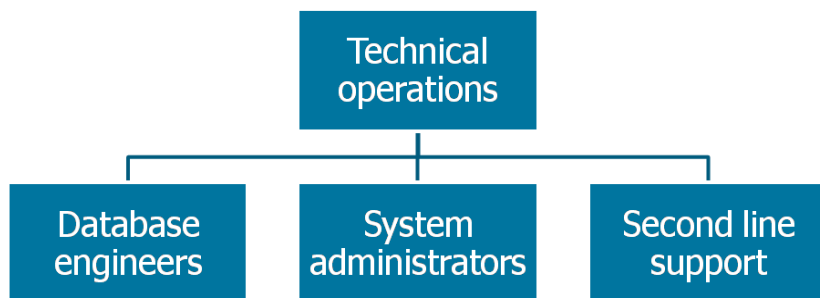
Ако някога сте били дежурни на повикване 24/7, имате добра представа колко отговорно е да управляваш производствена среда. От вас се очаква да разрешавате ситуации дори през нощта, независимо дали вие сте били причината за проблема. Никой не знае, затова се обаждат на вас. Това е голямо предизвикателство, защото вие не сте създали хардуера, драйверите, операционната система или специалния софтуер. Често опциите ви се ограничават до това да стесните рамките на проблема, да намалите влиянието му, да запазите следите, необходими да се пресъздаде той и да изчакате човека, който е отговорен за възникването му, за да го възпроизведе и разреши.

В администраторската работа както бързината, така и прецизността, и на двете - реакция и решение са ключови за успеха.

18

Защо, за Бога, да променяме нещата?

През 2008 г. един от нашите клиенти – скандинавска компания за производство на игри, предприе цяла серия от подобрения в процесите си. Едно от тези подобрения включваше въвеждането на Скръм в цялата организация за разработка на софтуер и постепенното премахване на пречките, които възпрепятстваха екипите да доставят продукта. Когато процесът на разработка на софтуер стана плавен и производителността се повиши, напрежението надолу към администраторските екипи се увеличи значително. Преди това те наблюдаваха процеса предимно отстрани, а сега все повече се включваха като активна страна в него.



Фигура 1. Администраторската организация включваше три екипа: администратори на бази данни, системни администратори и второ ниво консултанти

Като резултат, вече не беше достатъчно да помагаме само на екипите за разработка на софтуер. Ако продължавахме да се фокусираме върху тях, това щеше да доведе до забавяния в критични инфраструктурни подобрения, които се правеха от администраторските екипи. Бяха необходими подобрения и в двете области.

Освен това, прогресът на софтуерните екипи означаваше, че от мениджърите все повече се изискваше да помагат с анализ и мнения за даваните идеи. А това означаваше, че те имаха все по-малко време

за приоритизиране на задачи и решаване на проблеми в реално време. Мениджърският екип осъзна, че трябва да действат преди ситуацията да стане неуправляема.

19

Откъде да започнем?

Добра начална точка беше да попитаме софтуерните екипи, които са явяват наши клиенти.

Гледната точка на програмистите за администраторите

Попитах: „Кои са трите неща, които ви идват първи наум, когато мислите за „администратори“?“ Най-честите отговори бяха:

“Различно ниво на знания”

“Тяхната система за управление на процесите не струва”

“Много компетентни, що се отнася до инфраструктура”

“Какво правят тези хора?”

“Искат да помогнат, но всъщност е трудно да получиш помощ”

“Трябва да напишеш много писма, за да се свърши нещо просто”

“Проектите отнемат прекалено много време”

“Трудно се контактува с тях”

Накратко, това беше гледната точка на програмистите за администраторите. Сега да я сравним с погледа на администраторите върху работата на програмистите.

Гледната точка на администраторите за програмистите

"Ние"

(администраторите)



"Те"

(програμισите)



"Защо не използвате предимствата на съществуващата платформа?"

"Да направим публикуването на версия не толкова тежка работа!"

"Обидени сме от лошо качество на това което правите!"

"Те трябва да се променят", беше общ мотив в аргументацията и от двете страни. Очевидно трябваше да се промени тази нагласа, ако искахме да постигнем сработване при решаването на общи проблеми. Откъм положителната страна, „много компетентни, що се отнася до инфраструктура" (показващо доверие по отношение на основните компетенции) ме накара да вярвам, че мисленето „ние срещу тях" можеше да бъде поправено, ако създадем правилните условия за работа. Една възможна опция беше да елиминираме извънредната работа и да се фокусираме върху качеството.

20

Да накараме нещата да заработят

И така, трябваше да накараме нещата да заработят, но откъде да започнем? Единственото нещо, което знаехме със сигурност, беше, че там, откъдето започвахме, нямаше да е там, където ще завършим.

Аз съм бил програмист и със сигурност знаех малко за същността на администраторската работа. Нямаше да „нахлуя и да започна да правя промени“. Нуждаех се от подход, който да изисква по-малко конфронтация, но въпреки това да ни научи кои са важните неща, да изключи маловажните и да е лесен за изучаване.

Кандидатите бяха:

1. Скръм – той работеше добре със софтуерните екипи.
2. Канбан – нов и неизтестван, но подходящ заради Лийн принципите, които липсваха.

В дискусии с мениджърите Канбан и Лийн принципите изглежда отговаряха на проблемите, които се опитвахме да адресираме. От тяхна гледна точка, спринтовете нямаше да бъдат много подходящи, тъй като те правеха приоритизация ежедневно. Така че Канбан беше логичната отправна точка, макар да беше нов за всички нас.

21

Да стартираме с екипите

Как да стартираме с екипите? Нямаше наръчник за това как да накараме нещата да заработят. Ако направехме грешка, щяхме да рискуваме много. Не само, че щяхме да се простим с подобренията, но и ставаше дума за производствена платформа, обслужвана от високо специализирани и квалифицирани хора, които трудно се заменяха. Да ги отчуждим щеше де е мноооого лоша идея.

- Дали просто да започнем и да се справяме с последствията, когато се появят?
- Или първо да направим работен семинар?

Беше очевидно – първо трябва да направим семинар. Но как? Беше голямо предизвикателство да съберем целия администраторски отдел на семинар (кой щеше да отговаря, ако някой позвъни?). Накрая решихме да проведем полудневен семинар и да го направим прост и базиран на упражнения.

Семинарът

Една от положителните страни на семинара беше, че щеше да изведе проблемите на повърхността рано. Освен това, той осигуряваше и среда на високо доверие, където последствията можеха да бъдат дискутирани директно с членовете на екипите. Нека бъдем честни – не всички бяха крайно ентузиазирани да променят настоящия начин на работа. Но повечето членове на екипите бяха готови да опитат. И ние проведохме семинара, като демонстрирахме най-важните принципи и направихме опростена Канбан симулация.

Научете някои основни принципи	Канбан демо
<ul style="list-style-type: none"> • Ограничете работата до наличния капацитет. • Брой паралелни задачи срещу циклично време • Брой недовършени задачи срещу дебит • Теория на ограниченията 	<ul style="list-style-type: none"> • 3 "типа работа"; да се отговаря на въпроси, да се построи кола Лего, да се проектира и построи къща. • 3 итерации Измерете темпото за тип работа. Експериментирайте, настройте ограниченията за брой недовършени задачи. • Анализирайте.

В края на семинара гласувахме чрез метода „fist of five”, за да проверим дали екипите наистина искаха да опитат. Нямаше възражения на този етап, така че имяхме одобрение да продължим. (Б.П. - Методът „fist of five” е техника за постигане на консенсус, при която всеки участник показва съгласието си с пръстите на едната ръка. Юмрук означава блокирам решението, 1-2 пръста означава силно несъгласие, 3 означава резервирано съгласие, 4-5 пръста съгласие. Консенсусът е постигнат, когато всеки в групата покаже поне 3 пръста).

22

Да се обърнем към заинтересованите лица

Вероятно и заинтересованите лица биха били засегнати от въвеждането на Канбан. Но промените биха били към по-добро – те биха означававали, че екипът ще започне да казва „не” на работа, която не може да завърши, зада поддържа високо качество и да премахне задачите с нисък приоритет от списъка. Въпреки това, да се дискутира предварително винаги е добра идея.

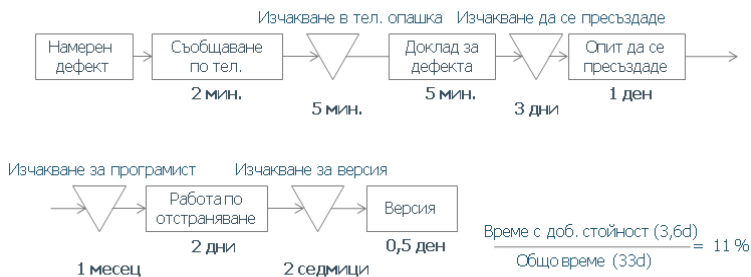
Най-близките заинтересовани лица, в нашият случай бяха първото ниво консултанти и мениджърите на отдела. Тъй като те участваха в семинара, вече бяха позитивно настроени към по-нататъшните промени. Същото се отнасяше и за софтуерните екипи (които така или иначе очакваха подобрения в по-малка или по-голяма степен). Но за един от екипите – консултантския, нещата бяха различни. Техният най-голям проблем беше, че бяха затрупани с работа. Също така, те се занимаваха с проблеми на клиентите, а компанията имаше ангажимент да отговаря на всички проблеми. Беше твърде вероятно това да се промени, ако въведем Канбан и започнем да налагаме ограничения за броят недовършени задачи.

И така, ние се срещнахме с ключовите заинтересовани лица и представихме намеренията си, очакваните ползи и възможните последици. За мое облекчение, идеите ни като цяло бяха добре посрещнати, понякога със забележки като „страхотно, ако най-после можем да приключим с тези проблеми”.

23

Да създадем първата дъска

Добър начин да започнете с оформянето на Канбан дъска е да направите карта на стойностния поток. В общи линии това е визуализация на стойностната верига и предоставя детайли за фазите на работата, потока и времето, за което задачите са били в системата (средно време на цикъл – cycle time).



Примерна карта на стойностния поток

Но ние започнахме с нещо дори по-просто – примерна Канбан дъска, която нарисувахме върху хартия заедно с мениджъра. Прегледахме я няколко пъти и започнахме. На този етап се появиха следните въпроси:

- Какви типове работа имаме?
- Кой се занимава с тях?
- Трябва ли отговорността да е споделена за различни типове работа?
- Какво да правим със споделената отговорност, имайки предвид специализираните умения?

Тъй като различните типове работа имаха различни споразумения за нивото на предоставената услуга (service level agreement), беше

естествено да оставим всеки екип да контролира дизайна на собствената си дъска. Те сами измислиха колоните и редовете.

Следващото голямо решение беше дали да има споделена отговорност върху различните типове работа. „Трябва ли да оставим една определена част от екипа да се занимава с директните въпроси (реактивна работа), а останалата част да се фокусира върху проектите (проактивна работа)?” Решихме първо да опитаме със споделена отговорност. Ключова причина беше, че идентифицирахме самоорганизацията и непрекъснатото обучение и трансфер на знания от страна на членовете на екипа като жизненоважен фактор за устойчив растеж. Недостатъкът на това решение бяха потенциалните прекъсвания за всички, но това беше най-доброто решение, което измислихме за начало. Малка странична забележка: когато проведохме семинара, екипите всъщност се самоорганизираха около този проблем. Те оставиха един човек да се занимава със спешните изисквания, а останалите – с по-големите задачи.

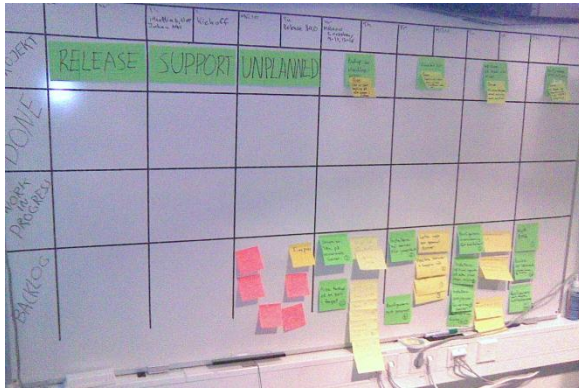
Първият Канбан модел

Отдолу е основният модел, който използвахме за Канбан. Забележете, че екипите решиха задачите да се движат нагоре (като мехурчета във вода), вместо както е в по-типичния модел отляво надясно.



Фигура 2. Това е първият модел на Канбан дъска. Приоритетите са подредени отляво надясно, а работният поток върви нагоре. Ограничението на броя на недовършени задачи (WIP) се смята като сбор от всички задачи на реда „Работа в прогрес” (ограден в

червено). Моделът е повлиян от примерите, описани от Линда Кук.



Фигура 3. Първата Канбан дъска на екипа системни администратори.

Използвани редове

Фази в процеса (ред)	Как го дефинирахме
Списък с истории	Историите, за които мениджърът е решил, че са нужни.
Готови да влязат в процеса	Истории, които са оценени и са разделени на задачи с максимална големина от 8 часа.
Започнати и недовършенит задачи	Редът с ограничение за недовършената работа. Започнахме с ограничение от 2 пъти по размера на екипа минус 1 (-1 за съвместна дейност). Така екип от 4 човека имаше лимит 7.
Готови	Може да се използва от потребителя.

Използвани колони

Тип работа	Как го дефинирахме
Издаване на версия	Помощ за екипа програмисти да издаде версия на софтуера за клиенти
Консултация	По-малки запитвания от други екипи.
Непланирани	Неочаквана работа, която трябва да бъде свършена, но няма ясен отговорник за нея – например - малки инфраструктурни подобрения.
Проект А	По-голям проект, свързан с администраторската работа, например подмяна на хардуера на тестова среда.
Проект Б	Друг по-голям проект.

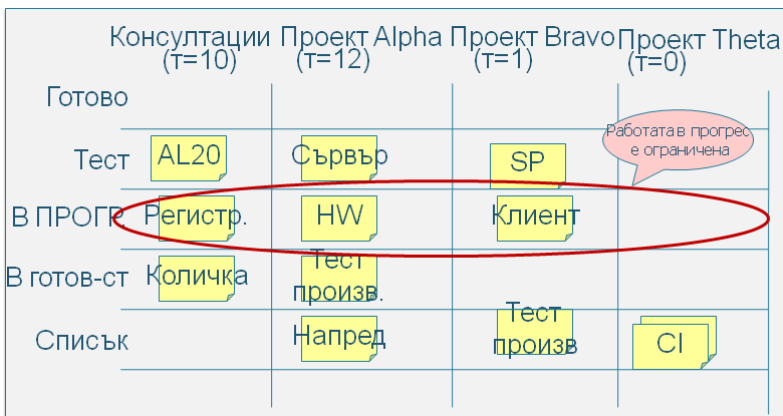
Не всички Канбан дъски изглеждаха по един и същи начин. Всички започнаха от проста схема и се развиваха в процеса на работа.

24

Да поставим първите ограничения за недовършени задачи (WIP)

Първото ни ограничение за недовършенит задачи (WIP) беше доста щедро. Аргументите ни бяха, че чрез визуализирането на потока ще видим и усетим какво се случва и беше малко вероятно да успеем да отгатнем най-доброто ограничение от самото начало. С течение на времето адаптирахме ограниченията всеки път, когато можехме да намерим добра причина за това (всичко, което трябваше да направим, беше да посочим към дъската).

Първото WIP ограничение, което използвахме, беше $2n-1$ (n =броят на членовете на екипа, -1 , за да насърчим съвместната работа). Защо? Просто е – няхаме по-добра идея ☺. Освен това изглеждаше обяснимо за начало. Формулата предоставяше просто и логично обяснение на всеки, който се опитваше да възложи работа на екипа. „...имайки предвид, че всеки член от екипа може да работи най-много по две неща едновременно, едно активно и едно в опашката, защо очаквате от тях да вземат *още*?“ Ако погледна назад, всеки щедр лимит щеше да свърши работа в началото. Като наблюдаваш Канбан дъската, много лесно можеш да разбереш какви са правилните ограничения в процеса на работа.



Фигура 4. Как приложихме ограничението за броят недовършенит задачи при екипите системни администратори и администратори на бази данни – едно ограничение, общо за всички типове работа.

Едно от наблюденията ни беше, че е безсмислено да се дефинира WIP ограничение в сюжетни точки. Беше прекалено трудно да се следи. Единственото достатъчно лесно за проследяване ограничение беше общият брой паралелни задачи.

За екипа консултанти дефинирахме ограничения по колони. Това беше нужно, тъй като ни трябваше по-бърза реакция, когато лимитът е достигнат.

25

Да съблюдаваме ограниченията за броя недовършени задачи (WIP)

Докато да спазваш ограниченията звучи лесно на теория, на практика е доста трудно. Това означава да кажеш „не” в някой момент. Пробвахме различни подходи да се справим с този проблем.

Дискусия пред дъската

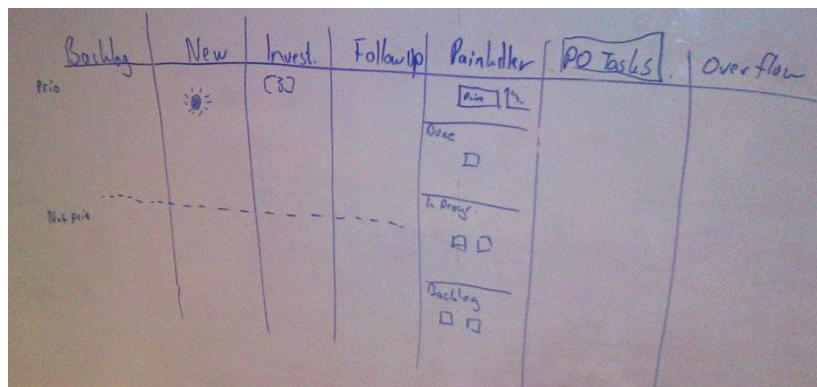
Ако откриехме нарушение в ограниченията, довеждахме заинтересованите лица при дъската и ги питахме какво искат да постигнат. В началото най-честата причина за нарушаването на ограничението беше липса на опит. В някои случаи се касаеше за различни гледни точки върху приоритизацията – обикновено ако експерт от екипа работи по специфична област. Това бяха единствените случаи, когато имахме търкания. През повечето време проблемите се разрешаваха на място в дискусия пред дъската.

Създаване на секция „резерв”

Ако да кажеш „не” предполагаше прекалено голям сблъсък и махането на елементи от списъка беше трудно, ние премествахме нещата с нисък приоритет в секция „резерв” на дъската, когато ограниченията бяха надхвърлени. Имаше две правила за задачите в резервната секция:

1. Те не са забравени; когато имаме време, ще се занимаем с тях.
2. Ако ги махнем от списъка, ще бъдете информирани.

Само след 2 седмици беше ясно, че никога няма да стигнем до резервните задачи, така че с помощта на мениджъра на екипа най-после те бяха махнати.

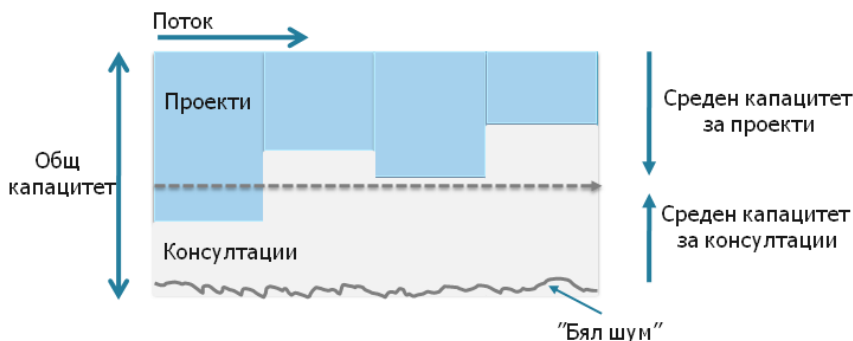


Фигура 5. Скица на Канбан дъската за екипа консултанти. Секцията резерв (Overflow) е накрая вдясно.

26

Кои задачи стигат до дъската?

Още в началото решихме да не слагаме на дъската *всичката* работа, която екипът върши. Да наблюдаваме неща като телефонно обаждане или пък кафе пауза щеше да превърне Канбан дъската в административно чудовище. Ние бяхме там, за да *решаваме* проблеми, а не да ги създаваме ☺. Така че решихме да слагаме само задачи с големина >1 час на дъската, а всичко по-малко се смяташе за „бял шум”. Ограничението от 1 час всъщност работеше сравнително добре и беше едно от малкото неща, които останаха непроменени. (Трябваше да преоценим предположението си за влиянието на страничния шум, но за това ще говорим по-нататък).



Фигура 6. Започнахме с предположение, че целият капацитет ще бъде консумиран от 2 типа работа – по-големи задачи (проекти) и по-малки задачи (консултации). Като следяхме темпото по проектите, можехме да получим индикация за датата на приключване, ако това беше необходимо. „Белият шум” (малки консултации < 1 час, срещи, кафе паузи, помощ за колеги) винаги беше очакван.

27

Как да правим оценки?

Това е постоянно случващо се събитие и със сигурност има повече от един отговор:

- Оценявайте редовно.
- Оценявайте, когато има нужда.
- Използвайте идеални дни/сюжетни точки за оценките.
- Оценките са несигурни, използвайте метода с размери тениски (малко, средно, голямо).
- Не правете оценки или оценявайте само когато има цена за забавянето, която оправдава това упражнение.

Леко повлияни от Скръм (тъй като идваме оттам, в крайна сметка), решихме да започнем със сюжетни точки. Но на практика, екипите третираха сюжетните точки като еквивалент на човекочасове (така им беше по-естествено). В началото, всички истории бяха оценявани. С течение на времето мениджърите научиха, че ако не правят много проекти паралелно, не карат заинтересованите лица да чакат. Те научиха също, че в случай на внезапна промяна, можеха да променят приоритетите и да адресират проблема.

Нуждата от планиране на крайна дата не беше вече толкова голяма. Това накара мениджърите да престанат да искат предварителни оценки. Правеха го само ако се страхуваха, че ще карат хора да чакат.

В един случай по-рано, един от мениджърите, стресиран от телефонно обаждане, обеща проектът да бъде завършен „до края на тази седмица”. Тъй като това беше проект на Канбан дъската, беше лесно да се оцени прогреса (като се преброят завършените истории) и да се заключи, че след една седмица около 25% цяха да бъдат завършени. Така цяха да са нужни още 3 седмици. Поставен пред този факт, мениджърът промени приоритета на проекта,

спря паралелната работа и направи обещания срок възможен. Винаги проверявайте на дъската ☺.

Какво означава оцененият размер? Времето за завършване на задачата или времето за работа по нея?

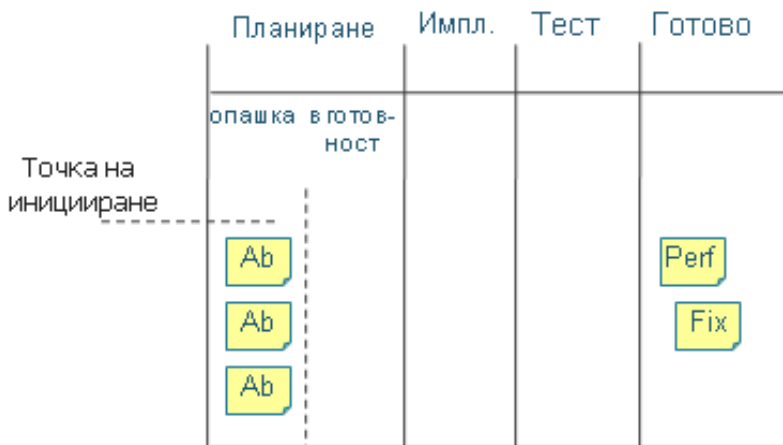
Нашите сюжетни точки означаваха време за работа, тоест колко часа непрекъснат труд очаквахме да отнеме тази история, а не времето за завършване на задачата (календарно време или колко часа ще чака клиентът). Чрез измерването на броя сюжетни точки, които достигаха графата „завършено” всяка седмица (темпо), можехме да изведем времето за завършване.

Всяка нова история оценявахме само по веднъж, не ревизирахме оценките по време на изпълнението. Така минимизирахме времето, което екипът отделя за планиране.

28

И така, как работехме в действителност?

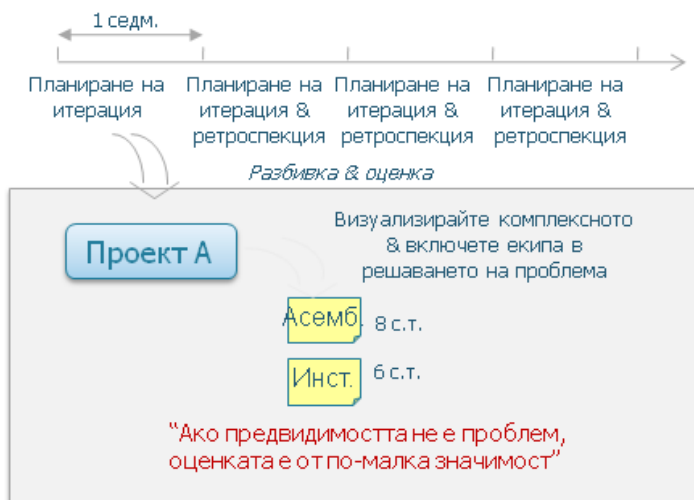
Канбан наистина има много малко ограничения, можеш да работиш по всякакви начини. Можеш да оставиш екипа да работи по планирани за определено време задачи или пък да избереш да вършиш тези дейности, когато се набере достатъчно инерция те да бъдат оправдани.



Фигура 7: Когато в списъка са постъпили 3 задачи, това води до сесия за планиране/оценка.

Избрахме да насрочим 2 регулярни срещи:

- Ежедневна правостояща среща – с екипа пред дъската, за да извадим проблемите наяве и да създадем обща представа за задачите на останалите членове на екипа
- Среща за планиране на седмична итерация – с цел планиране и непрекъснато усъвършенстване.



Това работеше добре за нас.

Ежедневна среща

Ежедневната правостояща среща беше сходна с тази в Скръм. Провеждаше се след общата (“Scrum of Scrums”) среща с участници от всички екипи (за разработка, тестване и администрация). Общата среща даваше важна информация на Канбан екипите, като проблемите, с които трябва да се занимаят на първо място или кой екип за разработка има най-голяма нужда от помощ в момента. В началото мениджърите често участваха в ежедневните правостоящи срещи, като предлагаха отговори на проблемите и приоритизираха решенията. С течение на времето, екипите започнаха да се самоорганизират по-добре и мениджърите посещаваха срещите по-рядко (но бяха на разположение в случай на нужда).

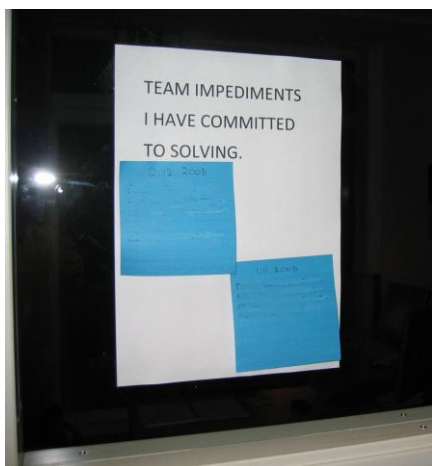
Планиране на итерация

Веднъж седмично провеждахме среща за планиране на итерация. Фиксирахме я в определено време всяка седмица, защото установихме, че ако не я планираме, времето се консумираше от други приоритети ☺. Имахме нужда и от повече комуникация в екипа. Обикновено точките за обсъждане бяха:

- Да актуализираме графиките и дъската (завършените проекти премествахме на „стената на завършеното”).

- Да си направим равностетка за изминалата седмица. Какво се случи? Защо? Какво може да се направи, за да се подобри положението?
- Адаптации на ограниченията за броя недовършени задачи (ако е необходимо)
- Разбиване на задачите на по-малки и оценка на нов проект (ако е необходимо)

В общи линии, планирането на итерация беше комбинация между оценка и проверка дали подобряваме процеса. Малки и средно големи проблеми се разрешаваха на място с помощта на преките мениджъри. Но по-комплексните проблеми, касаещи инфраструктурата, бяха по-трудно разрешими. За да се справим с това, дадохме на екипите възможност да дават до 2 „пречки за екипа” на своите мениджъри.



Правилата бяха:

1. Всеки мениджър може да работи по 2 задачи по всяко време.
2. Ако и двете позиции са заети, може да бъде добавена нова задача, ако се махне по-маловажната от другите 2.
3. Екипът решава кога проблемът е преодолян.

Това беше положителна промяна. Изведнъж екипите можеха да видят как мениджърите работят, за да им помогнат да решат дори трудните проблеми. Можеха да посочат дадена пречка и да попитат „как върви?”. Нямаше да бъдат забравени или пренебрегнати в полза на нова стратегия с висок приоритет.

Пример за сериозна пречка беше, че екипът администратори не получаваше необходимата помощ от програмистите, когато администрато чрите предполагаха дефект. Те се нуждаеха от програмист, който да им помогне да разберат коя част от системата причинява проблема, но тъй като програмистите бяха заети да пишат нови неща в спринтовете, проблемите продължаваха да се натрупват.

Не е изненада, че администраторите имаха чувството, че програмистите не се грижат достатъчно за качеството.

Когато тази пречка излезе наяве, тя беше ескалирана първо до прекия мениджър, след това по-нагоре до мениджъра на отдела. Той насрочи среща с мениджъра на отдела за разработка. В дискусиата, която последва, мениджърите се разбраха да поставят качеството на първо място. Разработиха ротационен принцип за консултации – всеки спринт един от екипите програмисти ще е на повикване и постоянно в готовност да помогне на администраторите. След като си осигури подкрепата от неговите мениджъри, шефът на отдела за разработка даде списък с имена за контакт. Те веднага решиха да проверят това решение, сигурни, че няма да проработи. Но този път беше истина; необходимата подготовка беше направена и пречката беше решена. Това донесе огромно облекчение на администраторите.

29

Да намерим концепция за планиране, която работи

Една история

Спомням си повратната точка за един от екипите. Седях с тях по време на втората им сесия за оценка. Екипът буксуваше с един проект, който не знаеше как да оцени. Имаше толкова много неизвестни и цялата сесия за оценка беше блокирана. Вместо да се намеся и да поема нещата в свои ръце, аз ги помолих да преработят процеса, за да намерят по-добро решение. Начело с мениджъра си, те поеха предизвикателството и започнаха да мислят за собствено решение. Това събитие беше важна повратна точка, печелившия ситуация, в която те се превърнаха в екип със самочувствие. След това започнаха да напредват толкова бързо, че ние трябваше да се оттеглим от пътя им.

Два месеца по-късно, техният мениджър дойде при мен след една ретроспектива. „Имам проблем“, каза той, посочвайки Канбан дъската на екипа си. „Нямаме истински проблеми, какво да правим?“

Да преоткрием планирането

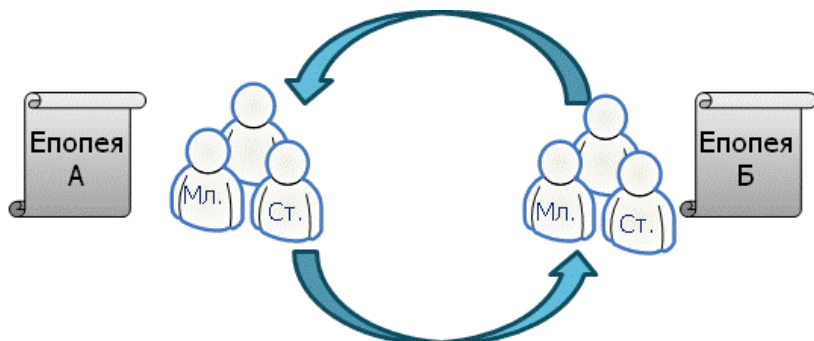
Сесиите за оценка с покер планиране, в които да се включват всички членове на екипите не работеха добре за никой от администраторските екипи. Някои от причините бяха:

1. Знанието беше разпределено прекалено неравномерно в екипа.
2. Най-често само един човек говореше.

3. Членовете на екипа искаха да се върнат към спешните задачи, които бяха оставили на работното си място.

Но с експерименти, екипите самостоятелно стигнаха до два различни процеса за оценка. Всеки от тях работеше добре за съответния екип.

Подход 1 – Размяна и преглед



- За всеки проект/история се определя двойка от по-опитен и по-малко опитен член на екипа, за да го оценят (например един човек, който е много добре запознат с тази история, и един, който не е). Това помага за обмяна на знания.
- Останалите членове на екипа си избират за коя история искат да помогнат в оценката (но при ограничение от 4 човека на история, за да е ефективна дискусиата).
- Всеки оценяващ екип разбива своята история на задачи и – ако е необходимо, я оценява.
- След това екипите си разменят задачите и преглеждат работата на другите (по един човек от екип остава, за да обясни на преглеждащите работата на своя група).
- Готово!

Обикновено цялата сесия за планиране на итерацията отнемаше около 45 минути и хората оставаха енергични по време на цялата среща. В повечето случаи се правеха по 1-2 промени при размяната на истории и преглеждането им от допълнително хора.

Подход 2 – общ преглед от експерт, след това оценка

Двама опитни членове на екипа правеха общ преглед на историята/проекта преди планирането. Те анализираха архитектурни решения и избираха едно от тях за проблема. След тази стъпка, екипът се включваше и правеше разбивка по задачи, като използваше предложеното решение за начална точка.



Фигура 8. Разбиване на задачи с преглед от друг екип по време на срещата за планиране на итерация. (Подход 1)

30

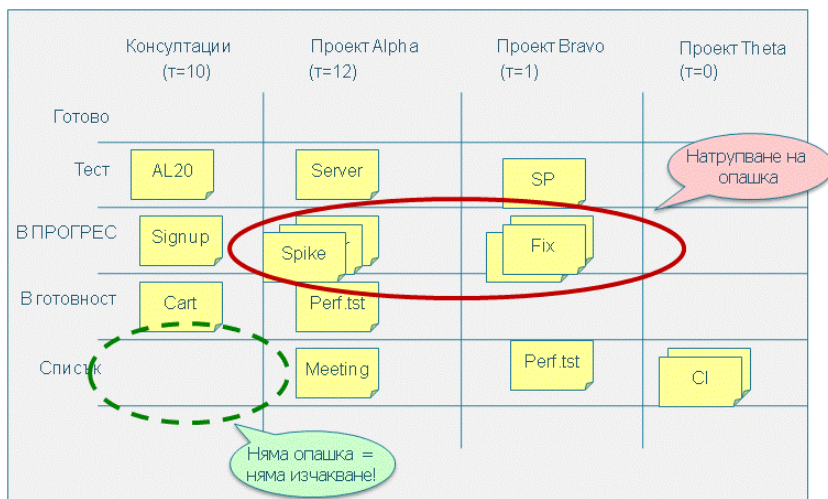
Какво да мерим?

Има много неща, които биха могли да се измерват – средна продължителност на цикъл (cyclic time - времето от момента, в който нуждата е открита, до момента, в който тя е удовлетворена), темпо, опашки, графики на оставаща работа... Най-важният въпрос е кои метрики могат да се *използват*, за да се подобри процеса. Моят съвет е да експериментирате и да видите какво работи за вас. Ние научихме, че графиките на оставащата работа са прекалено голямо усилие за всеки проект, по-кратък от 4 седмици. Цялостният прогрес можеше да бъде установен просто като се погледне Канбан дъската (колко истории са в списъка и колко са завършени).

Кандидат-метрика	Плюсове	Минуси
Средна продължителност на цикъл	Лесно се мери. Не е необходима оценка. Започва и завършва с клиента.	Не отчита размера.
Общо темпо (агрегирано за всички типове работа)	Груб, но лесен индикатор за посоката на подобрене и за вариациите.	Не помага за прогнозирането на крайни срокове за специфични типове работа.
Темпо за тип работа	По-точен индикатор от общото темпо.	За да е полезен, трябва да започне от нуждата на клиента до крайния резултат. Отнема по-дълго да се проследи в сравнение с общото темпо.
Дължина на опашките	Бърз индикатор за промени в търсенето.	Не показва дали причината е

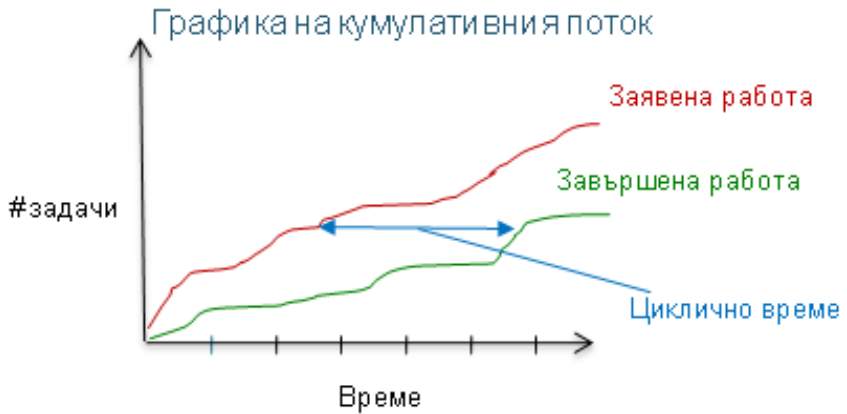
	Лесен за визуализиране.	<p>неравномерно търсене или неравномерен капацитет.</p> <p>Нулева опашка може всъщност да е показател за свръх-капацитет.</p>
--	-------------------------	---

Започнахме със замерването на „темпо за тип работа” и „дължина на опашките”. Темпото за тип работа се мери лесно и върши работа. Дължината на опашките са добри водещи индикатори, тъй като могат да се забележат веднага (стига да знаеш къде да гледаш за тях).



Фигура 9. Затруднения и възможности. С червено е отбелязано как опашките са се разраснали, което показва затруднения при тестването. *Липсата* на опашка в колоната за консултации показва, че няма изчакване за нови задачи там. Това е добър знак за високото ниво на обслужване на клиентите.

Не използвахме графика на кумулативните потоци, но би било интересно.



Не използвахме такива диаграми, тъй като Канбан дъската и графиката за темпото ни даваха достатъчно информация, поне за ранните ни цели да достигнем зрялост в процеса. Лесно можехме да идентифицираме затруднения, неравномерности и свръхработа и разрешаването на тези проблеми ни запълваше времето през първите 6 месеца.

31

Как започнаха да се променят нещата

Три месеца след като въведохме Канбан, мениджмънтът обяви екипа системни администратори за „най-продуктивен екип” в ИТ отдела. По същото време на ретроспекцията на компанията се проведе гласуване за „положително развитие” и този екип беше в топ 3. Ретроспекцията е събитие, което се провежда на 6 седмици и обхваща цялата компания, и това беше първият случай, в който *екип* се озоваваше в челната тройка! А само 3 месеца по-рано тези екипи създаваха затруднения и повечето хора се оплакваха от тях.

Качеството на услугите със сигурност се беше повишило. Как стана това?

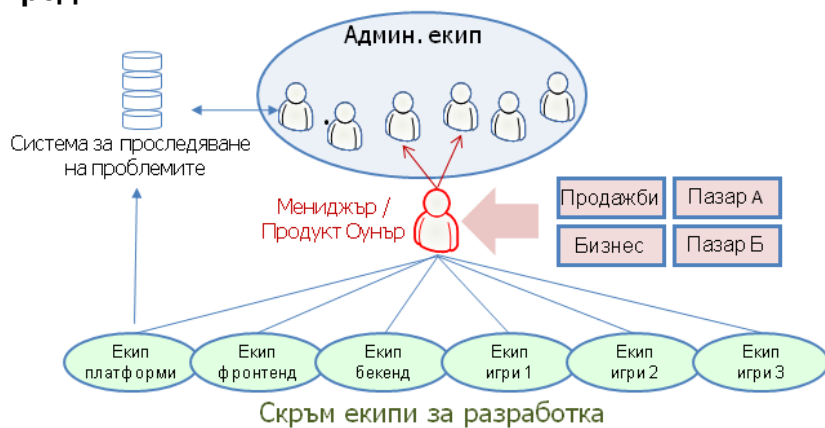
Важният момент беше, че всички започнаха да работят в една насока. Мениджърите предоставяха ясен фокус и предпазваха екипите от работа, която не е за тях, а екипите поемаха отговорността за качеството и крайните срокове. Отне приблизително 3-4 месеца, докато се появи този ефект, но след това работата вървеше гладко. Не че бяха изчезнали всички проблеми (това щеше да остави всички нас без работа, нали? ☺), но имаме нови предизвикателства, като например „как да запазим екипа мотивиран да се усъвършенства (когато основният проблем беше премахнат)?”

Едно важно парченце от пъзела на самоорганизацията беше въвеждането на идеята за „един контакт от администраторите за всеки екип”. Това означаваше да се даде на всеки екип програмисти техен личен консултант от екипа администратори. Канбан го направи възможно, като позволи на членовете на администраторските екипи да се самоорганизируют около работата, предотвратяваше свръхнатовареност и позволяваше непрекъснатото усъвършенстване. Преди, произволен човек от екипа щеше да вземе задача от опашката, да я свърши, доколкото може, и да започне със следващата. Всяка погрешна комуникация означаваше да се започне наново с ново искане за консултация. Когато беше въведена идеята за един контакт за екип, консултантите изведнъж получиха

възможност да отговарят бързо, когато системата беше застрашена заради лошо качество.

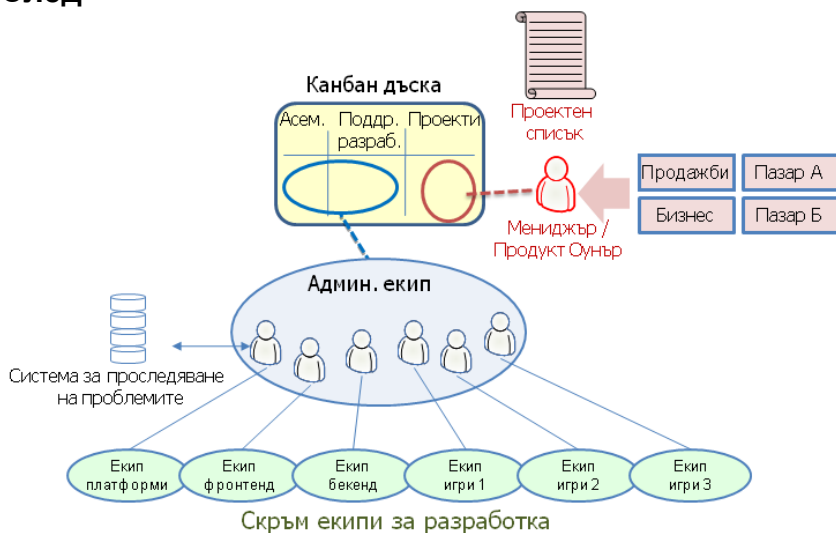
Бързо се появиха и специфични протоколи за комуникация; администраторите започнаха да използват програми за съобщения в реално време, за да говорят с програмистите, които познаваха добре, писма за онези, които пишеха по-добре, отколкото говореха, и телефон, ако това беше най-бързият начин да се реши проблема ☺.

Преди



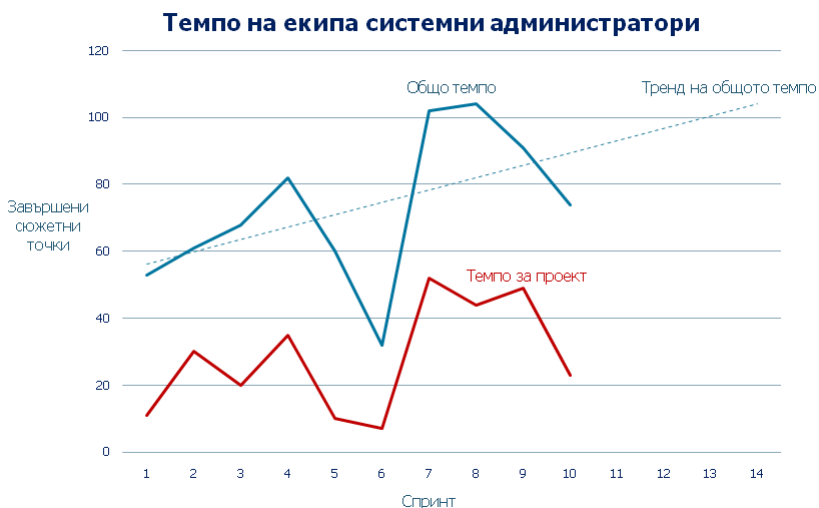
Фигура 10. Преди: прекият мениджър е основната точка за контакт с екипа. Всичко важно, което трябва да се свърши, минава през него. По-малките проблеми, обикновено от някой програмист, идват през система за проследяване на дефекти. Малко директни взаимодействия между конкретни хора.

След



Фигура 11. След: дефиниран е „един контакт от администраторите за екип“. Екипите за разработка говорят директно с определения им контакт. Много персонални комуникации. Администраторите се самоорганизират в работата си чрез Канбан дъската. Мениджърът насочва фокуса си към приоритизиране на по-големи проекти и предоставяне на помощ при по-трудни проблеми.

А какво е влиянието върху продуктивността на екипа?



Фигура 12. Общо темпо и темпо по проект, измерени със сюжетки точките, завършени през седмицата. Общото темпо е сумата от всички колони, а темпото по проект представлява частта, отделена за „проекти” (по-големи задачи, като например подобряване на хардуерната платформа). Двата спада се отнасят към 1) седмица, в която почти всички членове на екипа пътуваха, и 2) издаване на основна версия на софтуерния продукт.

И така, екипът демонстрираше положителна тенденция като цяло. В същото време той инвестира много в обмяна на знания чрез програмиране по двойки.

Докато сме на тази тема, нека погледнем продуктивността на екипа администратори на бази данни.



Фигура 13. Общо темпо и малки консултации. Спадът в средата е по Коледа.

Общото темпо се увеличава, макар че има значителни вариации. Размерът на вариациите вдъхнови екипа да наблюдава броят малки консултации (задачи, които обикновено са прекалено малки, за да бъдат отразени на Канбан дъската). Както можете да видите, графиката показва ясна обратна пропорционалност между броят малки задачи и общото темпо.

Екипът от консултанти започна с Канбан по-късно от другите два екипа, затова за него нямаме много надеждни данни все още.

Съзряване

Когато започнахме, беше лесно да се намират проблемните области. Но да се открие най-голямата възможност за усъвършенстване беше трудно. Канбан дъската ни даде ново ниво на прозрачност. Не само че беше по-лесно да се отбележат проблемите, но бяха повдигнати и важни въпроси за потока, вариациите и опашките. Започнахме да използваме опашките като инструмент за откриване на проблеми. Четири месеца след като започнахме да правим Канбан, мениджърите преследваха източниците на вариации, които тормозеха техните екипи.

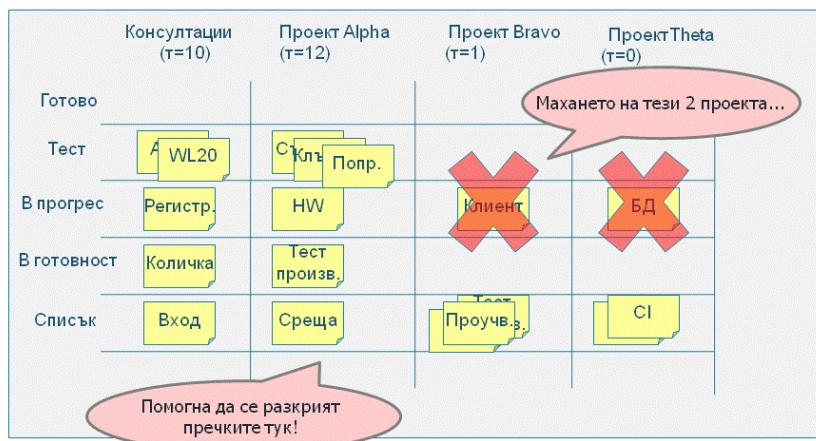
С превръщането на екипите от група индивиди в самоорганизиращи се единици, мениджърите осъзнаха, че пред тях стоят нови ръководни предизвикателства. Трябваше да се занимават повече с проблемите на хората – да отговарят на оплаквания, да дефинират общи цели, да решават конфликти и да уговарят споразумения. Това не беше безболезнен преход – те открито отбелязаха, че им е отнело енергия и умения да се научат да се справят. Но поеха предизвикателството и като красив резултат станаха по-добри лидери.

С намаляването на броя недовършени задачи, се появяват ограничения

Всички екипи започнаха с доста щедри ограничения за броя недовършени задачи. По това време се отделяше повече енергия за създаване на потока и за подsigуряване на подкрепата, от която се нуждаеше организацията.

В началото мениджърите искаха да имат много паралелни проекти, но след няколко седмици стана очевидно, че няма достатъчно капацитет за нископриоритетните проекти. Беше необходим един бърз поглед към дъската, за да се види, че работата с нисък приоритет никога не биваше започната. Това принуди мениджърите да намалят броя проекти на екип.

С течение на времето, потокът ставаше все по-стабилен за високоприоритетните задачи и започнахме да затягаме ограниченията за броя недовършени задачи. Направихме го, като намалихме броя проекти (колони) от три на два, а после на един. С това се показаха ограниченията извън екипа. Хората започнаха да споделят, че не получават помощ от другите навреме, и мениджърите насочиха вниманието си към този проблем.



Някои от останалите неща, които се появиха, включваха влиянието на некачествена работа, идваща от другите екипи, върху продуктивността на администраторите. Беше трудно да се поддържа равномерен и бърз поток, когато входящите елементи непрекъснато се нуждаеха от корекция.

Тези проблеми не бяха невидими и преди да започнем. По-скоро въпросът беше „с кои проблеми да се занимаем на първо място” и постигането на общо съгласие по този въпрос. С Канбан дъската всички можеха да видят как даден проблем оказваше влияние върху потока, което улесни набирането на инерция за справянето с този проблем на фирмено ниво.

Дъската ще се промени с течение на времето, бъдете гъвкави

Всички Канбан дъски се промениха с течение на времето. Обикновено отнемаше 2-3 промени в дизайна, докато екипът открие какво му върши работа. Така че вероятно няма смисъл да се инвестира много време в първия дизайн на дъската. Подсигурете си, че можете да я промените лесно. Ние използвахме черно тиксо, за да я оформим. Така лесно можехме да променяме и можехме да използваме и стените, както и бяла дъска. Друг начин, който съм виждал, е да се нарисуват рамките на дъската с дебел маркер (но проверете, че можете да го изтриете! ☺).

По-долу ще видите типичен пример на оптимизация на дизайна. В началото приоритетите са се променяли често, така че за да избегнат преместването на цяла колона от листчета напред и назад, екипът е сложил номера на приоритета върху всяка колона.



Фигура 14. Ранна Канбан дъска със стикери за настоящите приоритети

Не се страхувайте от експерименти, дори и да са неуспешни

Урокът, който аз научих от това приключение, беше, че наистина няма крайна точка. Ще се провалим в момента, в който решим, че има. Съществуват само безкрайни експерименти и учене. Да не грешиш означава да не се учиш. Ние се провалихме няколко пъти (лош дизайн на дъската, оценки, объркани графики на оставаща работа и т.н.), но след всеки провал научавахме нещо ново и важно. Ако бяхме спрели да опитваме, как щяхме да се научим?

Успехът на Канбан сега е вдъхновил мениджмънта и Скръм екипите също да експериментират с Канбан дъски. Може би тази книга ще им помогне!

Финални точки, които да запомните

Започнете с ретроспекции!

Много възможности и неща, които да обмислите, а? Надявам се, че тази книга е разпръснала част от мъглата. На нас поне ни помогна :o)

Ако искате да промените и подобрите процеса си, позволете ни да вземем едно решение вместо вас веднага. Ако не правите ретроспекции редовно, започнете с това! И те трябва да водят до реални промени. Извикайте външен модератор, ако е необходимо.

След като имате ефективни ретроспекции, пътуването към изграждане на точния процес за вашия контекст е започнало – независимо дали той е базиран на Скръм, XP, Канбан, комбинация от тях или каквото и да е друго.

Никога не спирайте да експериментирате!

Целта не е Канбан или Скръм, а да учим непрекъснато. Едно от страхотните неща в софтуерния бизнес е краткият цикъл за обратна връзка, който е ключът към научаването. Използвайте го! Поставайте под въпрос всичко, експериментирайте, проваляйте се, учете и експериментирайте наново. Не се страхувайте от грешки в началото, защото такива ще има! Просто започнете от някъде и се развивайте нататък.

Единственият истински провал е да не учиш от грешките си.

Но, хей, и от това можете да научите нещо.

Успех и приятно пътешествие!

/Хенрик & Матиас, Стокхолм, 2009-06-24

Х: Това ли е всичко?

М: Да, струва ми се. Нека спрем тук.

Х: Може би трябва да кажем кои сме ние?

М: Добра забележка. Ако се представим добре, може да си намерим клиенти за консултации.

Х: Хайде да го направим тогава! И след това приключваме.

М: Да, имаме и други неща за вършене, читателите също.

Х: Всъщност, отпускът ми тъкмо започва :о)

М: Хей, не ми го натяквай.

За авторите

Хенрик Книберг и Матиас Скарин са консултанти в Crisp, Стокхолм. Те обичат да помагат за успеха на фирмите, както в техническите, така и в човешките аспекти на разработването на софтуер, и са помогнали на десетки компании да приложат Лийн и Аджайл принципите на практика.

Хенрик Книберг

През последното десетилетие Хенрик е бил технически директор (СТО) на 3 шведски ИТ компании и е помогнал на много други да подобрят процесите си. Той е сертифициран като Certified Scrum Trainer и работи често с пионери в Лийн и Аджайл практиките, като Джеф Съдърланд, Мери Попендийк и Дейвид Андерсън.



Предишната книга на Хенрик – „Скръм и XP от фронтната линия“, има над 150,000 читатели и е сред най-популярните по темата. Той няколко пъти е печелил награди за най-добър лектор за представянето си в международни конференции.

Хенрик е израснал в Токио и сега живее в Стокхолм с жена си София и три деца. В свободното си време е активен музикант, композира музика и свири на бас китара и синтезатор с местни групи.

henrik.kniberg@crisp.se

<http://blog.crisp.se/henrikkniberg>

<http://www.crisp.se/henrik.kniberg>

Матиас Скарин

Матиас работи като Лийн консултант и помага на компаниите да постигнат ползите от Лийн и Аджайл. Той е ментор на всички нива – от програмисти до мениджъри. Помогнал е на компания за игри да намали времето за създаване на нова игра от 24 на 4 месеца, възстановил е доверието в цял един отдел за разработка и е един от ранните пионери на Канбан.



Като предприемач, е създател и управител на 2 компании.

Матиас има магистърска степен по управление на качеството и е работил като програмист 10 години по критични (mission-critical) системи.

Матиас живее в Стокхолм и харесва рокендрол, да танцува, да се състезава с кола и да кара ски.

mattias.skarin@crisp.se

<http://blog.crisp.se/mattiasskarin>

<http://www.crisp.se/mattias.skarin>

Речник

Голяма част от използваните в книгата термини са преведени на български език. Някои много специфични понятия, обаче, са транслитерирани, за да се запази по-широкият контекст, който те носят в себе си в допълнение към конкретното им значение на български. Тези понятия са обяснени при първото им срещане в книгата.

За улеснение на читателите, тук прилагаме речник на най-важните преведени понятия с оригиналните английски термини.

Оригинален термин	Превод на български език
block/impediment	пречка
bug	дефект
cycle time/lead time	средна продължителност на цикъл
daily standup meeting	ежедневната правостояща среща
developers	програμισи
done	готово
epic	епопея
flow	Поток
in process	в прогрес
limit	ограничение
OK	добре

pull system	система за изтегляне
release	версия на продукта
software development	разработка на софтуер
stakeholders	заинтересовани лица
story	История
story points	сюжетни точки
throughput	Дебит
to do	да се направи
value map stream	карта на стойностния поток
velocity	Темпо
visual control	визуален контрол
WIP	брой недовършени задачи /започнати и недовършени задачи, в някои случаи - текуща работа
work states	фази на работа
technical operations/system administration	администратори/администраторска работа