

Кратък обзор на Scrum метода

CSCB778 Самостоятелна работа по
софтуерен инженеринг

Николай Атанасов Ванков F48897

1/24/2013

Съдържание

Какво е Scrum	2
Съставни части на Scrum	3
Scrum екип.....	3
Scrum Master	3
Product Owner	4
Екип на разработчици	4
Scrum артефакти	5
Product Backlog.....	5
Sprint Backlog.....	6
Инкремент.....	6
Scrum събития	6
Scrum спринт	6
Планиране на Scrum спринт	8
Дневен Scrum	8
Ревю на спринта.....	9
Ретроспекция на спринта.....	10
Backlog Refinement Meeting (Backlog Grooming Meeting).....	10
Заклучение	10
Използвани ресурси	11

Какво е Scrum

През 1990г. Кен Швабер (Ken Schwaber) и Джеф Сутерланд (Jeff Sutherland) въвеждат във фирмите си един по – различен подход на разработка, който по – късно става известен като Scrum. Заедно те представят получената информация и опит от използването на този метод през 1995 на OOPSLA конференцията, състояла се в Тексас.

Scrum е agile (гъвкав) метод, който представлява алтернатива на предишните подходи спрямо проектиране и разработване на продукт (било то софтуер или друг). Макар и труден за усъвършенстване той е лесен за разбиране и прилагане. В основата му стои идеята, че знанието идва от опита и че много повече информация ще е налична по време на разработката отколкото в началото. Предполага се, че колкото повече се работи по даден проект толкова повече се научава за него.

Scrum методологията ни позволява да концентрираме усилията си в завършването на най – важните задачи първо. На кратък период от време (спринт) се получава готов, тестван и работещ софтуер, който може да се имплементира. Това от своя страна води до по – добра възможност да се прецени дали може да се използва софтуера, дали има нужда да се усъвършенства и да се правят промени. Scrum метода не предполага, че всичко ще се знае в началото – клиентът може да промени решението си, може някаква функционалност да се окаже излишна или липсваща. Основна цел е в кратки срокове да има готов софтуер, който да може да се представи на клиента и да се получи обратна връзка вместо да се работи по предварително установен договор, който може и да не задоволява напълно нуждите му.

Scrum попада в категорията на гъвкавите методи. Те се фокусират върху това да се доставя качествен софтуер бързо, на малки итерации, вместо да се работи с един огромен и труден за подържане продукт.

Гъвкавите методи:

- Ценят отделните индивиди и техните способности пред процеса и наличните средства
- Работещ софтуер пред подробна документация
- Работа и постоянно взаимодействие с клиента пред преговори по договори
- Адаптивност към промени пред изпълнение по план

Съставни части на Scrum

Scrum екип

Състои се от:

- **Scrum Master** – следи Scrum методологията да се спазва с цел оптимизиране на процеса на работа, отстранява външни дразнители, които могат да попречат на екипа на разработчиците да свършат работата си
- **Product Owner** – отговаря за Product Backlog и е връзката на екипа със заинтересованите лица
- **Екип на разработчиците (Development Team)** – работят по проекта, създават и тестват парчета работещ софтуер на през определено време (спринт)

Целта на организацията на Scrum екипа е да насърчи гъвкавост, креативност и продуктивност като оставя на екипа да реши сам как трябва да изпълни поставените му задачи. Екипът представя готови продукти итеративно като по този начин дава голяма възможност да получи обратна връзка от заинтересованите лица.

Scrum Master

Отговорникът за това Scrum методологията да бъде спазвана и следвана, да се следват всички теории, практики и правила. Негова е грижата всички аспекти от гъвкавия метод да бъдат разбрани. Той е този, който помага на другите, извън екипа на Scrum, да разберат методологията и как да си взаимодействат с екипа така, че да има най – голям резултат.

Негова е грижата всички от екипа да са запознати със Scrum метода. Той трябва винаги да следи за новости при прилагане на определени техники, да обсъжда с други Scrum Master-и във фирмата и да намира нови подходи, които могат да подобрят процеса. Трябва да може да разбира дългосрочното проектиране в една емпирична среда, да оказва гъвкавост и да може да съдейства в Scrum събитията когато е нужно. Той е този, който ще обясни на всички в организациите за Scrum методологията и ефективността на прилагане на емпиричния подход. Макар и недокументирано в официалното ръководство в много случаи Scrum Master следи да няма напрежение между другите участници и се грижи да осигури спокойна среда, в която всеки да се чувства сигурен да споделя своето мнение.

Той следи списъкът със задачите (Product Backlog) да бъде напълно разбран от всички, Product Owner да поставя разбираеми и семпли елементи в Product Backlog и асистира при предлагане на нови такива от екипа на разработчиците. Scrum Master помага при комуникацията между Product Owner и екипът на разработчиците. Той се грижи да обучи всички в екипа на самоорганизираност и многофункционалност, насочва екипа към създаване на качествен софтуерен продукт. Едно от по – важните задължения на Scrum Master е да не се пречи на екипа на разработчиците по време на спринт. Той следи за всички външни фактори, които могат да попречат. В много случаи клиентът може да иска допълнителна функционалност от екипа на разработчиците - тогава Scrum Master трябва да се намеси и да не допусне задачите от спринтът да се изменят от външни лица и да насочи клиентът да доуточни с Product Owner новите изисквания.

Product Owner

Product Owner е човекът, който се грижи за **Product Backlog** – списък с всичките изисквания и функционалности, които трябва да се добавят в проекта. Винаги ролята се изпълнява от един човек, а не от група хора.

Той се грижи всички задачи от списъка да бъдат подредени по приоритет на изпълнение. Той трябва да направи всички елементи (или поне тези, които ще бъдат включени в следващият спринт) от Product Backlog разбираеми за екипа. Екипът на разработчици може да помогне на Product Owner в поддръждането и поддържането на Product Backlog, може да помага елементите му (**PBI – Product Backlog Item**) да се разделят на по – малки, изпълними в един спринт, елементи, но винаги крайното решение за включените PBI и поддържае им принадлежи на Product Owner.

Той играе ролята на буфер между заинтересованите лица (**product stakeholders**) и екипът на разработчици. Всеки може да предложи да бъдат добавяни нови PBI, но Product Owner решава дали да ги включи ли не. Следи Product Backlog да бъде разбираем за разработчиците и за заинтересованите лица.

За да може Product Owner да постигне резултат всички трябва да уважават неговите решения, а самият той да бъде добре запознат както с възможностите на екипа си, така и с исканията на клиентите. Той е начело на списъкът със задачите, той го приоритизира и поддържа, негова е отговорността екипът да разработва използваем и качествен софтуер и да не се разсейва от приоритетните задачи.

Екип на разработчици

Екипът на разработчиците е самоорганизирана и многофункционална относително малка група от хора (от 3 до 5 души). Те сами решават как ще свършат работата си по време на спринта и кой с какви задачи ще се занимава. От целият Scrum екип само членове на екипа на разработчиците могат да създават инкременти.

Екипът трябва да работи като едно цяло и да има всички необходими специалисти за да приключи дадена задача, не се допускат и признават хора извън екипа. Желателно е да не се разделят хората по ранг (старши/младши) – при нужда от помощ може да се намеси и Scrum Master и да неутрализира нещата, за да е сигурен, че екипът няма да се раздели. Въпреки, че всеки има различни квалификации, екипът никога не се разделя на по – малки екипи. Отговорността да се свърши дадена задача се пада върху всички и всеки един член на екипа има право на глас.

Големина на екипа:

- не твърде голям - губи се връзка и се изразходва време и ресурси за да се осъществи комуникацията между отделните членове на екипа
- не твърде малък - итерациите ще имат по - малко завършени задачи, има шанс някои даже да отпаднат, може по време на итерация да се окаже, че липсва специфично умение, което може да забави работата

Добра практика е при екип, чиито членове са разделени да се намери начин за по – честа комуникация или да се разбие на по – малки екипи.

Scrum артефакти

Product Backlog

Нареден по важност списък за цялата желана функционалност, която е нужна да присъства в продукта. Също така е единствен източник на всякакви искания за промени по продукта. Product Backlog съдържа всички функционалности, изисквания, подобрения и корекции, които трябва да бъдат направени в следващите версии на продукта за да може да остане конкурентно способен.

Всеки елемент (**PBI – Product Backlog Item**) има записана кратка история, която описва за кого е нужна функционалността и как ще се използва – т.нар. user stories. В user story се обяснява в едно до две изречения кой ще е вършителят, какво иска да направи и защо (*„Като учител искам да мога да добавям оценките на студентите в системата, защото много пъти си губят студентските книжки“*) за да може екипът по – адекватно да разбере проблема и да отговори на исканията.

Всички проиграни потребителски истории (user stories) трябва да спазват определени изисквания, следвайки **INVEST**¹ рамката:

- **I – Independent** (независима). Една история е много по – лесно за проследяване ако не зависи от други такива. Когато историите не се застъпват по какъвто и да е начин екипът има възможност да избере в какъв ред да изпълни задачите.
- **N – Negotiable** (договоряема). Историята не трябва да задължава определени функции да присъстват – тя трябва да улавя есенцията на проблема, не функционалността. Всички детайли трябва да бъдат уточнени от програмистите и клиентите
- **V – Valuable** (ценна). Една история трябва да бъде ценна за клиента. Дори и екипът да има нещо предвид, добавяйки някоя история, главните цели трябва да бъдат отговаряне на потребителските искания. Често пъти най – лесният начин да се покаже функционалността на заинтересованите лица е чрез user stories.
- **E – Estimable** (пресмятане на приблизително време) – за всяка история трябва да може да се даде приблизително време за изпълнение. Освен че служи за ориентир на екипа, Product Owner и клиента, също така показва, че задачата е напълно ясна и може да се определи колко време ще отнеме реализирането ѝ.
- **S – Small** (малка). Всяка история трябва да бъде достатъчно малка за да бъде реализирана в един спринт. Ако не е то тази история може да се раздели на по – малки такива.
- **T – Testable** (може да се тества). След реализирането на задачата, поставена от историята, трябва да има точно дефинирани спецификации за да може да се определи дали реализацията отговаря на изискванията. Ако не може да се даде критерий за тестване това може да означава, че историята не е разбрана добре.

¹ INVEST in Good Stories, and SMART Tasks - <http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>

Product Owner отговаря за Product Backlog, подредбата и съдържанието му. Той подрежда елементите по приоритет като най – важните са на върха. Никога не може да има два или повече елемента на едно място. Колкото по - високо е даден елемент толкова по - добре и подробно е описан и толкова по - наложителен е. Най – важните елементи на Product Backlog винаги имат име, user story и оценка като оценката се определя от екипа на разработчиците.

Когато един продукт се разпространи информацията, която се получава от потребителите също оказва влияние върху Product Backlog. Той не спира да се изменя докато самият продукт също продължава да се използва. Product Backlog никога не се завършва, само се променя за да може по - адекватно да отговаря на изканията. Първоначалната му версия съдържа само най - добре разбрани и най - базови неща, които трябва да съдържа продукта, и постепенно се *изменя и еволюира*, както и се изменя и еволюира самият продукт.

Sprint Backlog

Всички елементи от Product Backlog, които са избрани да се осъществят в текущият спринт. Приоритетни задачи са най – важните от списъкът с елементи като екипът на разработчиците определя колко неща може да завърши в рамките на текущият спринт и до колко може да се натовари. Екипът на разработчиците описва в Backlog на спринта каква функционалност смята да реализира в следващият инкремент. В Sprint Backlog трябва да има достатъчно информация, така че да може да се разбират задачите в дневният скръм и как да се напредва в проекта.

Само екипът на разработчиците има достъп до Sprint Backlog по време на спринт и само той може да редактира. Освен в частни случаи никой, дори и Product Owner, не може да добавя елементи. Ако елемент от Backlog на спринта стане ненужен може да се премахне. Sprint Backlog представлява реална картина за това какво екипът смята да направи по време на спринта.

Инкремент

Инкрементът представлява съвкупността на всички завършени задачи от Product Backlog по време на спринт. Инкрементът трябва да може да работи и да се тества, да е в напълно готово състояние. Product Owner решава дали инкрементът да се пусне в експлоатация.

Scrum събития

Scrum спринт

Единица време (месец или по - малко) за което се създава продуктов инкремент, изцяло завършен и готов за ползване компонент. По време на спринта се изпълняват само задачи, дефинирани в Sprint Backlog и елементите от него не се променят. Много пъти екипът създава табло с графи като „да се свърши“, „изпълнява се“ и „приключено“ за да може по – лесно да проследяват и контролират задачите си.

В частни случаи по време на спринт елементите от Sprint Backlog могат да бъдат преразгледани от екипа и Product Owner ако първите установят, че не могат да се справят с всички поставени задачи или че имат възможност да свършат допълнително работа.

Всеки спринт може да бъде разглеждан като отделен проект с време за изпълнение - един месец. Всеки спринт, подобно на всеки проект трябва в края на срока да постигне някаква завършеност. След края на спринтът изготвеното трябва да може да се тества и да работи.

Спринтът е ограничен във времето до един месец. Ако е по - дълъг съществува рискът да се променят изискванията докато протича спринта и се усложнява планирането, работата и управлението. Рискът също може да се повиши, докато спазвайки времевият период рискът се намалява до загуба на месец време в най - лошият случай.

Един Scrum спринт се състои от:

- среща за планиране на спринта
- ежедневните Scrum срещи
- работата, която се извършва по време на спринта
- ревю на спринта
- анализиране и ретроспекция на спринта
- среща за редактиране и оформяне на Product Backlog

По време на спринта:

- не се правят промени, които могат да променят целта на спринта (но спринтът може да бъде спрял от Product Owner)
- съставът на екипа не бива променян
- целите не се променят
- в крайни случаи времето на спринтът може да бъде доизясняван и преразгледан от Product Owner и екипът на разработчиците ако е получена нова информация, която може да афектира по някакъв начин разработката

Отменяне на спринт

В случаи, когато Product Owner забележи, че инкрементът няма да реализира нищо, което да е от важност за проекта, той може да спре текущият спринт. Причина за това може да бъде налагане на нова технология, която отхвърля старата, рязка промяна в политиката на фирмата или исканията на клиента, пропадане на проекта и други.

Отменянето на спринта е крайна мярка и рядко се стига до това, тъй като дори и да настъпи голяма промяна почти винаги инкрементът носи някаква стойност и Scrum спринтовете са твърде кратки за да може да възникне такава критична ситуация, който да наложи моментално спиране.

Право да спре спринтът има право само Product Owner, въпреки, че той може да го направи под влияние на другите участници в Scrum екипа или под влияние на заинтересованите лица.

Когато спринтът се спре се виждат кои цели са завършени и всички останали се преразглеждат и се преценява дали да бъдат върнати в Product Backlog. Спирането на спринтове е рядко срещано, защото много време и ресурси се отделят в преразглеждането и регрупирането и в повечето случаи времето на спринта е толкова кратко, че не може да възникне нещо толкова важно или което да си заслужава прекратяването на спринта.

Планиране на Scrum спринт

Срещата определя какво ще бъде направено по време на спринта. Планът за изпълнение се разработва от целия екип на Scrum. Времето, което се предоставя за срещата е осем часа планиране за един месец работа, а ако времето за разработка е по - кратко то времето за планиране също намалява пропорционално (за две седмици ще се отделят четири часа, за една - два и т.н.).

Времето се разделя на две равни части:

- какво ще се изпълни по време на спринта
- как ще се изпълни

Какво да се изпълни по време на спринта

Scrum екипът се събира за да обсъди какво върху какво да бъдат съсредоточени усилията. Product Owner представя елементите от Product Backlog и екипът разглежда какво да се свърши по време на спринта.

Информацията, с която се разполага е резултатът от предишния спринт, капацитетът на задачи, които екипът на разработчици могат да свършат за даден период и предишната му работа. Само екипът на разработчиците може да определи броя на задачи за следващият спринт.

След като задачите са уточнени се изгражда Sprint Goal - цел на спринта, която ще се осъществи ако се изпълнят всички задачи от Product Backlog-а на предстоящият спринт.

Как да се изпълнят задачите

Начертава се план, който да бъде следван и по стъпките му да се свърши дадената задача. Екипът се самоорганизира да изготви плана по време на планирането и по време на спринта (ако е необходима промяна). Product Owner може да присъства за да осигури правилното разбиране на задачите. Ако при преразглеждането се установи, че задачите са малко или много могат да се добавят или премахнат елементи.

Цел на спринта

Определя се задачата, която трябва да се извърши. Екипът на разработчиците вършат работата си като имат в предвид целта на спринта, който може да се разглежда и като *milestone* на проекта. Ако задачата се окаже твърде трудна екипът може да обсъди с Product Owner Backlog-а на спринта.

Дневен Scrum

Всеки ден се прави по 15 минутна среща, в която се събират всички от екипа на разработчиците и обсъждат свършена и оставаща работа. Дневният scrum се състои на едно и също място, по едно и също време за да се избегнат недоразумения. Събирането започва в точният час, не се изчакват закъснели.

Всеки от екипа споделя:

- Какво е свършил вчера
- Какво възнамерява да свърши днес
- Какви трудности среща

Срещата е ограничена в 15 минути (**time boxed**) като Scrum Master трябва да следи всички да се придържат към основните проблеми. Ако възникне странична тема за дискутиране той може да я запише някъде и след срещата да останат тези, които имат желание да дискутират и доуточняват. Scrum Master се грижи всеки ден да се провежда събиране, но е задача на екипа на разработчици да проведат самият дневен Scrum. Това събиране има за главна цел да се определи как задачи от Product Backlog ще се трансформират в инкремент, а не за да се определи статуса на свършената работа.

Дневният Scrum се използва за да се установи до колко се напредва към целта на спринта и колко задачи от Product Backlog могат да се изпълнят. Екипът на разработчици трябва всеки ден да бъде в състояние да обясни на Product Owner и Scrum Master как възнамерява да постигне дневните задачи като самоуправяван екип и да създаде очаквания инкремент.

Дневният Scrum подобрява комуникациите, премахва нуждата от други срещи, открива и премахва пречки, които биха затруднили работата и подобрява знанията на екипа за целия проект. Той няма за цел да дава отчет на външни лица, а да запознае участниците с проекта и да им помогне в организацията помежду им.

Ревю на спринта

Провежда се на края на всеки спринт. Екипът на разработчиците и заинтересованите лица заедно обсъждат какво е направено по време на спринта. Базирайки се на информацията за свършената работа (и ако са възникнали някакви промени по Product Backlog) се определят целите на следващият спринт. Срещата е неформална, като времето, което се отделя за нея е 4 часа ревю за един месец работа (ако спринтът е по - кратък то ревюто също е пропорционално по - кратко).

Ревюто може да бъде разделено на четири основни части:

1. Демонстрация на работещите продукти, направени по време на спринта
2. Product Owner определя кое е завършено и кое трябва да се върне в Product Backlog
3. Измерва се свършената работа (незадължително)
4. Получава се обратна връзка от заинтересованите лица

По време на ревюто се разглежда броят и трудността на изпълнените задачи. След време Product Owner може да придобие представа относно каква работа и в какъв срок може да се изпълни и да дава относителни прогнози на клиентите.

Екипът на разработчиците споделят кои неща са минали добре, какви проблеми са имали по време на спринта и как са ги разрешили.

По време на ревюто:

- Екипът демонстрира свършената работа и отговаря на въпроси, свързани с инкремента
- Product Owner обсъжда Product Backlog на проекта и базирайки се на информацията за свършената до момента работа дава прогнози за дати на приключване
- Цялата група обсъжда какво да се свърши от тук нататък. Информацията се използва като входни данни за следващото планиране на спринт.

Ретроспекция на спринта

Ретроспекцията дава възможност на целият скръм екип да определи свършената работа и възникналите проблеми и базирайки се върху информацията до сега да предложи план за подобрене на работата, който да се следва по време на следващият спринт. Ретроспекцията е тричасова за едномесечин спринт, като пропорционално се намалява с намаляването на времето на спринта.

Ретроспекцията се провежда след ревюто на спринта и преди следващата среща за планиране на спринт. Целта и е да се определи как е минал последният спринт по отношение на хората, участващи в него, взаимоотношенията между екипа, използваните инструменти, да се определи как може да се подпомогне и усъвършенства работата на екипа на разработчиците и други. Тук важна роля има Scrum Master за да следи дали има проблеми, които се появяват постоянно и да предприеме някакви мерки за да ги предотврати в бъдеще.

Backlog Refinement Meeting (Backlog Grooming Meeting)

Макар и в официалното ръководство на Scrum да не е отделена като отделно събитие срещата за редактиране и оформяне на Product Backlog се провежда всеки спринт и помага на Product Owner да подобри Product Backlog, да добави нови PBI към него и информация към вече съществуващи елементи или да раздели един епичен елемент на няколко обикновени такива.

По време на Backlog Refinement Meeting всички елементи се преразглеждат. Времето, отделено за това събитие не трябва да заема повече от 10% от времето за спринта.

Епичен елемент (Epic)

Епичният елемент представлява сложен PBI, който може да бъде разделен на два или повече обикновени елемента. Epic елемент не се приема от екипът на разработчици, те помагат на Product Owner да се определи всички подзадачи.

Заклучение

В много случаи Scrum подхода може да добринесе в развитието на една организация. Той дава алтернативен подход като позволява постоянна връзка с клиента и постепенно развитие на продукт без да се знаят всички първоначални изисквания.

Използван и в много големи фирми като Microsoft, Google, HP и други, той е лесен за усвояване, дава добри резултати и позволява бързо да се предложи готов продукт на пазара.

Използвани ресурси

- Michael James, **SCRUM REFERENCE CARD**. Адрес: <http://scrumreferencecard.com/ScrumReferenceCard.pdf> (последно достъпен на 26.01.2013 в 19:20 ч.)
- Bill Wake, **INVEST IN GOOD STORIES, AND SMART TASKS**. Адрес: <http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/> (последно достъпен на 26.01.2013 в 19:20 ч.)
- Michael James, **SELECTED ARTICLES ON SCRUM**. Адрес: <http://scrumreferencecard.com/> (последно достъпен на 26.01.2013 в 19:20 ч.)
- **ОФИЦИАЛНО РЪКОВОДСТВО НА SCRUM НА АНГЛИЙСКИ**. Адрес: http://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum_Guide.pdf (последно достъпен на 26.01.2013 в 19:20 ч.)
- **SCRUM ALLIANCE**. Адрес: <http://www.scrumalliance.org/> (последно проверен на 26.01.2013 в 19:20 ч.)
- **РЕСУРСИ НА AGILE ALLIANCE НА ТЕМА SCRUM**. Адрес: http://cf.agilealliance.org/articles/article_list.cfm?CategoryID=17 (последно проверен на 26.01.2013 в 19:38 ч.)
- **ECLIPSE PROCESS FRAMEWORK (EPF) ПРОЕКТ - КРАТКО ОБЯСНЕНИЕ НА SCRUM ПРОЦЕСА**. Адрес: <http://epf.eclipse.org/wikis/scrum/> (последно проверен на 26.01.2013 в 19:38 ч.)