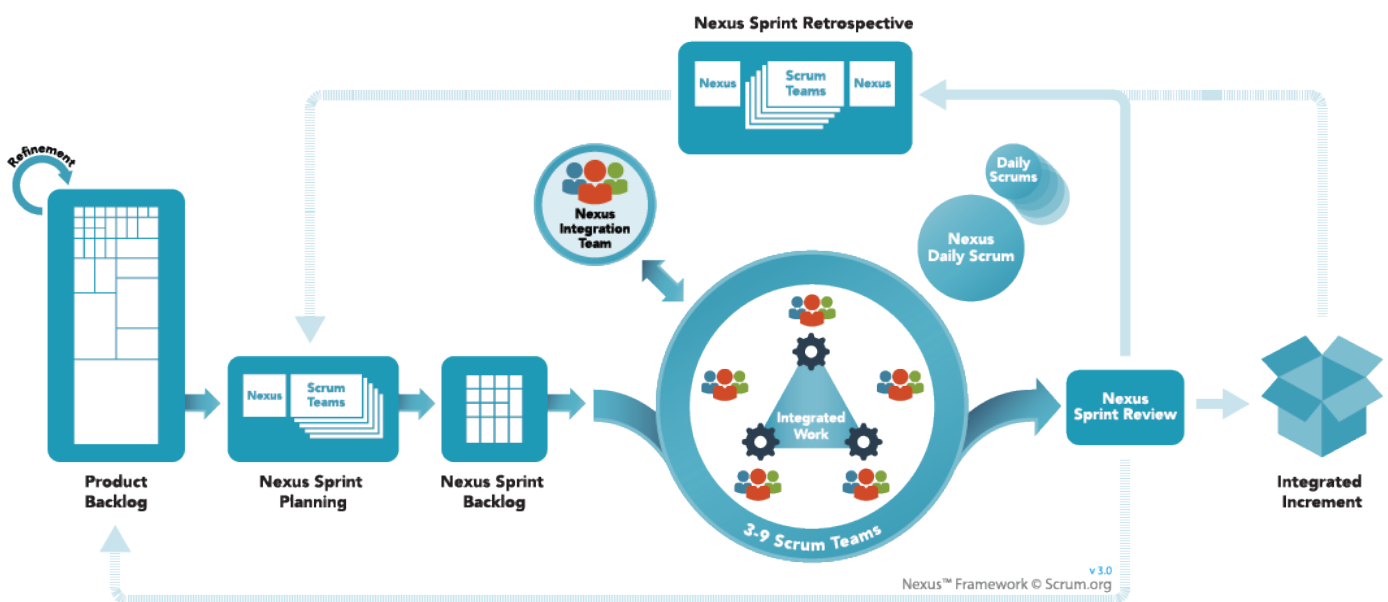


Посібник із Nexus™

Вичерпна інструкція до Nexus:
Основні принципи масштабованого розвитку
Scrum



Ken Schwaber

Розроблений та актуалізується Кеном Швабером і Scrum.org

Серпень 2015

Зміст

Знайомство із Nexus	2
Мета Посібника із Nexus	2
Визначення Nexus	2
Передумови Nexus	2
Структура Nexus	3
Процес роботи Nexus	4
Методи розробки програмного забезпечення	5
Nexus	5
Ролі Nexus	5
Інтеграційна Команда Nexus.....	5
Події Nexus	7
Планування Nexus Спринту	7
Щоденний Nexus Scrum	8
Огляд Nexus Спринту	8
Ретроспектива Nexus Спринту	8
Удосконалення	9
Артефакти Nexus	10
Беклог Продукту	10
Мета Nexus	10
Беклог Nexus Спринту.....	10
Інтегрований Інкремент	10
Прозорість Артефакту	11
Визначення критерію “Готово”	11
Заключне слово	11
Подяка	12
Переклад	12

Знайомство із Nexus

Мета Посібника із Nexus

Nexus є основою для розвитку і підтримки масштабованих продуктів, а також ініціатив розробки програмного забезпечення. Він базується на принципах методології Scrum. Цей посібник містить визначення Nexus. Воно складається з ролей, подій, артефактів та правил Nexus, які поєднують перелічені компоненти разом. Кен Швабер і команда Scrum.org, які розробили Nexus, є також авторами цього посібника.

Визначення Nexus

Термін Nexus (іменник): одиниця розробки (в Scaled Professional Scrum)

Nexus є структурою, що складається з ролей, подій, артефактів та методів, які пов'язують і поєднують разом роботу приблизно від трьох до дев'яти Scrum-команд, що працюють над одним Беклогом Продукту, аби створити Інтегрований Інкремент, який відповідає меті.

Передумови Nexus

Розробка програмного забезпечення є комплексним процесом, а інтеграція її результатів у функціонуюче програмне забезпечення складається з багатьох компонентів і подій, які потрібно узгодити між собою, аби наприкінці отримати продукт із статусом «Готово». Робота повинна бути організованою, послідовною, питання взаємозалежності мають бути вирішені, результати досягаються поетапно. Доки програмне забезпечення ще не існує у фізичному стані, воно демонструє додаткові складнощі.

Багато хто із розробників користувався структурою Scrum для колективної роботи в команді, щоб розробити Інкремент функціонуючого програмного забезпечення. Проте якщо у Беклозі Продукту і в одній і тій самій кодовій базі працює більше однієї Scrum-команди, проблем не уникнути. Яким чином спілкуватимуться розробники програмного забезпечення із непов'язаних між собою команд під час виконання завдань, що мають безпосередній вплив одне на одне? Як вони, працюючи у різних командах, поєднуюватимуть результати виконаної роботи та тестуватимуть Інтегрований Інкремент? Такі питання вже постають, коли об'єднуються дві команди, а коли йдеться про три команди чи навіть більше, ці питання перетворюються на проблеми.

Існує багато залежностей, які виникають у роботі кількох команд, що співпрацюють для створення завершеного Інкументу із статусом «Готово» принаймні кожного Спринта. До них належать:

1. Вимоги: Обсяг вимог може дублюватися, а способи, в який вони реалізуються, також можуть впливати один на одне. Цей факт необхідно враховувати під час упорядкування Беклога Продукту та вибору вимог.

2. Знання сфери діяльності: учасники команд знаються на різних професійних галузях та комп'ютерних системах. Ці знання потрібно розповсюдити між Scrum-командами для забезпечення компетентності їх учасників, а також для того, аби мінімізувати взаємне відволікання між командами протягом Спринта.
3. Програмне забезпечення та тестові артефакти: вимоги відображаються або відобразяться в майбутньому в коді програмного забезпечення та тест-комплектах.

Залежність між командами може бути знижена в залежності від міри розподілення вимог, професійної обізнаності членів команди та кодових/тестових артефактів, віднесених до одних і тих самих Scrum-команд.

Під час масштабування розробки програмного забезпечення за методологією Scrum команди повинні організовуватись з огляду на згадані взаємозв'язки вимог, знань, а також програмного забезпечення/тестових артефактів. В залежності від цього буде оптимізовано й продуктивність роботи.

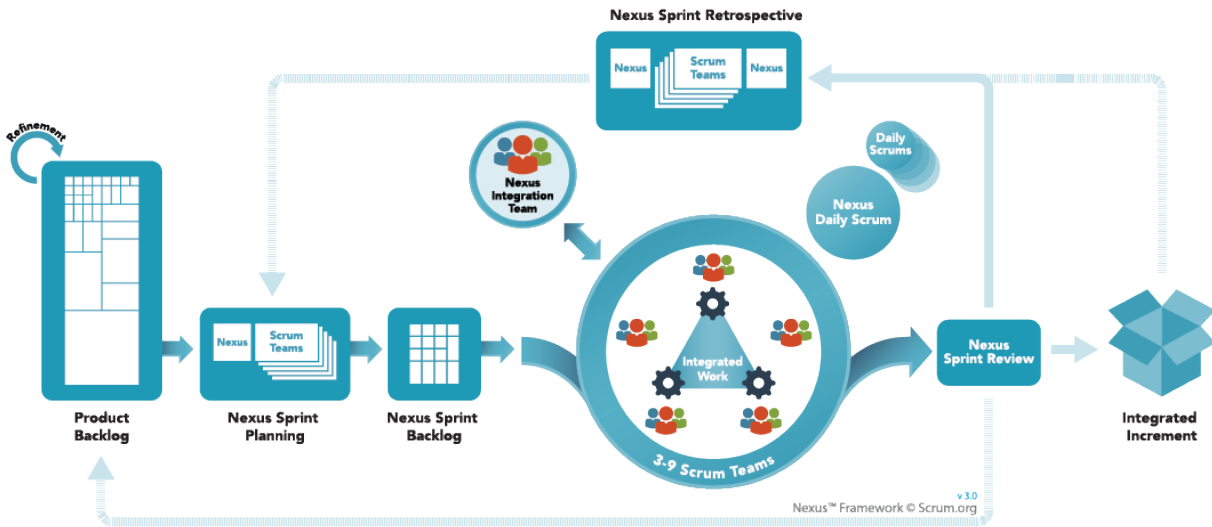
Структура Nexus

Nexus - це зовнішній каркас, який охоплює одразу декілька Scrum-команд, об'єднаних задля створення Інтегрованого Інкременту. Nexus сумісний із Scrum, і його фрагменти вже знайомі тим, хто працював над проектами за принципами Scrum. Різниця між Nexus та Scrum полягає в тому, що в першому випадку більше уваги приділяється залежностям і взаємодії між Scrum командами, що забезпечують один "Готовий" Інтегрований Інкремент принаймні кожного Спринта.

Як показано на наступному зображенні, Nexus складається з таких елементів:

- Ролей: нова роль, Інтеграційна Команда Nexus, створюється для керування, тренування (коучингу) та контролю над застосуванням Nexus і роботою Scrum таким чином, щоб здобути найкращі результати. Інтеграційна Команда Nexus складається з Власника Продукту, Scrum-майстра й Учасників Інтеграційної Команди Nexus.
- Артефактів: усі Scrum-команди використовують єдиний загальний Беклог Продукту. Водночас із тим як елементи Беклога Продукту удосконалюються та підготовлюються, стає видимою інформація стосовно того, яка саме команда працюватиме у Спринті. Новий артефакт, Беклог Nexus Спринту, допомагає забезпечити прозорість протягом Спринту. Усі Scrum-команди опікуються своїми індивідуальними Беклогами Спринту.
- Подій: події додано до регулярних подій Scrum, розташовано довкола них, або ж (у випадку Огляду Спринта) вони замінюють регулярні події Scrum, щоб їх посилити. Коли події модифікуються, то допомагають як спільним зусиллям усіх Scrum-команд в Nexus, так і для кожної окремої команди.

Структура Nexus™, екзоскелет масштабованого Scrum



Процес роботи Nexus

Будь-яку роботу в Nexus можуть виконати всі члени Scrum-команд, оскільки вони є крос-функціональними членами Nexus. З огляду на залежності, для виконання певних завдань команди можуть обрати найбільш придатних своїх учасників.

- Покращити Беклог Продукта: Беклог Продукта потрібно розкласти на компоненти таким чином, щоб залежності можна було встановити і видалити або мінімізувати. Елементи Беклогу Продукта можна розкласти на дрібні частки функціональності, і команду, яка напевно виконуватиме роботу, потрібно призначити якомога раніше.
- Планування Nexus Спринта: відповідні представники від кожної Scrum-команди зустрічаються, щоб обговорити і проаналізувати покращений Беклог Продукта. Вони обирають його елементи для кожної команди. Потім кожна Scrum-команда планує свій власний Спринт, взаємодіючи належним чином з іншими командами. Результатом є ряд цілей Спринта, які долучаються до загальної мети Nexus, Беклогом Спринта кожної Scrum-команди і єдиного Беклогу Nexus Спринта. Беклог Nexus Спринта робить прозорими обрані елементи Беклога Спринта Scrum-команди і будь-які залежності.
- Процес розробки: всі команди розроблюють програмне забезпечення, повсякчас інтегруючи свою роботу в загальному середовищі, яке можна протестувати, аби переконатися в тому, що інтеграція відбулася успішно.
- Щоденний Nexus Scrum: відповідні представники від кожної Scrum-команди з розробки зустрічаються щодня, аби визначити, чи існують які-небудь проблеми інтеграції. Якщо існують, то ця інформація вертається назад до Щоденного Scrum кожної Scrum-команди. Scrum-команди потім використовують свій Щоденний Scrum

для створення плану на день аби вирішити проблеми інтеграції, встановлені протягом Щоденного Nexus Scrum.

- Огляд Nexus Спринта: всі команди зустрічаються із Власником Продукту для огляду Інтегрованого Інкременту. До Беклога Продукту можна вносити зміни.
- Ретроспектива Nexus Спринта: відповідні представники від кожної Scrum-команди зустрічаються для виявлення загальних проблем. Потім кожна Scrum-команда проводить індивідуальну зустріч, Спринт Ретроспективу. Після чого певні представники від кожної команди зустрічаються знову, щоб обговорити всі необхідні дії на основі загальних проблем, щоб поширити відомості у висхідному, тобто знизу вгору, порядку.

Методи розробки програмного забезпечення

Для поєднання роботи Scrum-команд, які співпрацюють, щоб створити Інтегрований Інкремент, необхідно багато різних методів розробки програмного забезпечення. Більшість цих методів вимагає автоматизації, яка допомагає керувати обсягом і складністю роботи і артефактів, перш за все в масштабованих середовищах.

Nexus

Ролі, події та артефакти Nexus наслідують мету Scrum та зосереджуються на атрибутах, відповідних ролям, подіях і артефактах Scrum згідно зі Scrum Guide.

Ролі Nexus

Nexus складається з Інтеграційної Команди Nexus та близько трьох-дев'яти Scrum-Команд.

Інтеграційна Команда Nexus

Інтеграційна Команда Nexus відповідає за те, щоб Інтегрований Інкремент (спільна робота, завершена Nexus) розроблявся принаймні кожного Спринта. За принципами Scrum, Scrum-команди відповідають за розробку Інкrementів програмного забезпечення, яке потенційно готове до випуску. Усі ролі для членів Scrum-команд зазначені в Scrum Guide.

Інтеграційна Команда Nexus - це Scrum- команда, яка складається з:

- Власника Продукту
- Scrum-майстра
- Одного або кількох членів Інтеграційної Команди Nexus

В залежності від обставин і потреб Члени Інтеграційної Команди Nexus можуть також працювати у Scrum-командах в тому ж Nexus. Якщо така необхідність є, то пріоритетною є робота для Інтеграційної Команди Nexus. Участь в Інтеграційній Команді Nexus вагомша,

ніж участь в окремій Scrum-команді. Завдяки цьому пріоритетною лишається робота з вирішення проблем, які мають наслідки для багатьох різних команд.

Склад Інтеграційної Команди Nexus може з часом змінюватися, щоб, таким чином, відобразити нагальні потреби Nexus. До спільних активностей, якими може займатись Інтеграційна Команда Nexus, належать коучинг, консультування та підвищення рівню усвідомлення важливості проблеми залежностей, а також крос-командних ускладнень. Також може виконуватися робота з Беклога Продукту.

Інтеграційна Команда Nexus бере на себе будь-які проблеми інтеграції. Вона відповідає за успішну інтеграцію виконаної усіма Scrum-командами роботи в Nexus. Сюди належить вирішення будь-яких технічних та нетехнічних крос-командних проблем, які можуть перешкоджати Nexus безперервно розробляти Інтегрований Інкремент. Аби досягти згоди в цьому процесі, необхідно дотримуватися висхідного принципу поширення інформації.

Власник Продукту в Інтегрованій Команді Nexus

Nexus опрацьовує єдиний Беклог Продукту, і, як зазначено в структурі Scrum, Беклог Продукту має одного Власника продукту, який приймає остаточне рішення щодо його вмісту. Власник продукту відповідає за максимальне збільшення цінності продукту і роботу, виконану й впроваджену Scrum-командами. Власник продукту знаходиться в Інтеграційній Команді Nexus.

Власник продукту відповідає за упорядкування та удосконалення (уточнення) Беклогу Продукту щоб, таким чином, можна було здобути максимально можливу цінність Інтегрованого Інкременту, створеного Nexus. Прийоми, які дозволяють досягнути необхідного результату, можуть сильно відрізнятись в залежності від Організацій, Nexus та Scrum команд, а також окремих осіб.

Scrum-майстер в Інтеграційній Команді Nexus

Scrum-майстер в Інтеграційній Команді Nexus несе повну відповідальність за те, щоб структура Nexus була зрозумілою для інших та активована. Такий Scrum-майстер може також бути Scrum-майстром в одній або декількох інших Scrum командах в цьому Nexus.

Учасники Інтеграційної Команди Nexus

Масштабована робота із розробки вимагає застосування інструментів і методів, які зрідка використовуються окремими Scrum командами. Інтеграційна Команда Nexus складається з професіоналів, які є кваліфікованими спеціалістами у використанні цих методів та інструментів в галузі систем розробки взагалі. Учасники Інтеграційної Команди Nexus відповідають за те, аби методи і інструменти дійсно застосовувались, були зрозумілими і використовувались для виявлення залежностей, а також часто поєднують усі артефакти відповідно до визначенню критерію "Готово".

Члени Інтеграційної Команди Nexus відповідають за коучинг і управління Scrum-командами в Nexus, щоб набути, впровадити і вивчити ці методи і інструменти. Крім того, вони навчають Scrum-команди необхідним стандартам розробки, інфраструктури або архітектури софту, яких дотримується організація задля розробки якісних Інтегрованих Інкрементів.

Якщо основні обов'язки Членів Інтеграційної Команди Nexus ними виконуються, то вони можуть також працювати в якості членів Команди Розробки в одній або декількох Scrum-командах.

Події Nexus

Тривалість подій Nexus встановлюється з огляду на тривалість відповідних подій, поданих в Scrum Guide. Ці події обмежені за часом (є таймбоксами) на додачу до відповідних подій Scrum.

Планування Nexus Спринту

Метою Планування Nexus Спринту є узгодження діяльності всіх Scrum-команд в Nexus для єдиного Спринту. Власник Продукту поширює знання професійної галузі та контролює вибір і пріоритетність рішень.

Аби розпочати Планування Nexus Спринту, відповідні представники кожної Scrum-команди перевіряють і корегують перебіг роботи, що виконується під час подій з Удосконалення (уточнення). Всі члени Scrum-команди повинні брати участь в Плануванні, аби мінімізувати виникнення проблем комунікації.

Мету Nexus Спринту формулюють протягом Планування Nexus Спринту. Вона описує ціль, якої досягнуть Scrum-команди під час Спринту. Коли усі учасники повністю розуміють роботу, яку потрібно виконати для Nexus, кожна Scrum-команда береться до планування власного окремого Планування Спринту. Якщо ця зустріч відбувається у спільному для всіх середовищі, команди можуть продовжувати ділитися свіже з'ясованими залежностями. Планування Nexus Спринту вважається завершеним, коли кожна Scrum-команда закінчила свої особисті події з Планування Спринту.

Протягом Планування Nexus Спринту можуть виникати нові залежності. Їх необхідно візуалізувати та звести до мінімуму. Також можна скоригувати послідовність виконання роботи командами. Беклог Продукту, покращений належним чином, дозволить мінімізувати появу нових залежностей під час Планування Nexus Спринту. Усі елементи Беклога Продукту, які було обрано для Спринту та їх залежності потрібно візуалізувати у Беклозі Nexus Спринту.

Беклог Продукту мусить бути належним чином удосконалений шляхом встановлення та усунення/зменшення залежностей ще до того, як розпочато Плануванням Nexus Спринту.

Щоденний Nexus Scrum

Щоденний Nexus Scrum є такою подією, яка дозволяє відповідним представникам окремих Scrum-команд (а саме Розробникам) перевірити поточний стан Інтегрованого Інкременту та виявити проблеми інтеграції чи нововиявлені крос-командні залежності.

Протягом Щоденного Nexus Scrum учасники повинні зосередитися на впливі кожної команди на Інтегрований Інкремент і обговорити наступні питання:

- Чи було успішно інтегровано роботу, виконану попереднього дня? Якщо ні, то чому?
- Які нові залежності було виявлено?
- Яку інформацію слід поширити між командами в Nexus?

Протягом Щоденного Nexus Scrum потрібно використати Беклог Nexus Спринту задля візуалізації поточних залежностей, а також їх впорядкування.

Робота, яка визначається під час Щоденного Nexus Scrum, потім повертається до окремих Scrum-команд для планування в межах своїх подій Щоденного Scrum.

Огляд Nexus Спринту

Огляд Nexus Спринту проводиться наприкінці Спринту, для того, щоб надати зворотній зв'язок щодо Інтегрованого Інкременту, створеного Nexus протягом Спринту.

Огляд Nexus Спринту замінює окремі Огляди Спринту, проведені Scrum-командами, тому що в першу чергу необхідно зібрати відгуки зацікавлених осіб саме стосовно усього цілого Інтегрованого Інкременту. Неможливо буде продемонструвати завершену роботу в подробицях. Можуть знадобитися певні прийоми для максимального отримання відгуків з боку зацікавлених осіб.

Ретроспектива Nexus Спринту

Ретроспектива Nexus Спринту - це офіційна можливість для Nexus зосередитися на перевірці та адаптації. Вона складається з трьох частин:

1. Перша частина Ретроспективи дає відповідним представникам Nexus зустрітися та виявити проблеми, які стосувались більш ніж однієї команди. Мета зустрічі полягає в тому, щоб зробити загальні проблеми, прозорими для всіх Scrum-команд.
2. Друга частина складається з того, що кожна Scrum-команда проводить власну Ретроспективу Спринту, в спосіб, описаний в Scrum Guide. У своїх групових обговореннях команди можуть звертатись до проблем, що були оголошені в першій частині Ретроспективи Nexus. Окремі Scrum-команди повинні спланувати заходи щодо вирішення цих проблем під час своїх власних командних Ретроспектив Спринту.

3. Заключна, третя частина надає можливість відповідним представникам Scrum-команд знову зустрітися і домовитися про те, яким чином візуалізувати і відстежувати встановлені дії. Це дозволяє цілісно застосовувати Nexus.

Оскільки вони є звичайними дисфункціями масштабування, кожна Ретроспектива має розглянути наступні питання:

- Чи лишилась якась частина роботи невиконаною? Чи Nexus згенерував технічний борг?
- Чи були всі компоненти, зокрема, код, часто (щодня) успішно інтегровані?
- Чи було програмне забезпечення успішно розроблене, протестоване і розгорнуте досить часто, аби запобігти надмірному накопиченню невирішених залежностей?

До вищезазначених питань за необхідності додати ще питання:

- Чому це сталося?
- Як можна запобігти створенню технічного боргу?
- Як запобігти повторенню?

Удосконалення

У масштабі Nexus існує багато рівнів удосконалення. Тільки тоді, коли елементи Беклога Продукту є адекватно незалежними, їх можна обрати та опрацювати без зайвого конфлікту між Scrum-командами в Nexus.

В основі таких показників як кількість, частота, тривалість і відвідуваність зустрічей з Удосконалення лежать залежності, притаманні Беклогу Продукту. Чим вище складність і більше залежностей, тим більше елементів Беклога Продукту потрібно вдосконалити (уточнити), аби позбутися залежностей. Елементи Беклогу Продукту проходять через різні рівні декомпозиції, від дуже великих і нечітких запитів - до здійсненої роботи, яку змогла виконати одна Scrum-команда у Спринті.

Удосконалення Беклога Продукту в масштабі переслідує подвійну мету. Воно прогнозує, яка з команд буде виконувати конкретні елементи Беклога Продукту, а також визначає залежності між цими командами. Візуалізація дозволяє командам контролювати залежності і звести їх до мінімуму.

У першій частині крос-командного Удосконалення потрібно розкласти елементи Беклога Продукту якомога детальніше, для того, аби збагнути, які команди могли б їх виконати, і в якій послідовності майбутніх Спринтів.

Протягом другої частини Удосконалення потрібно сфокусуватися на залежностях. Їх необхідно встановити і візуалізувати для команд та Спринтів. Ця інформація знадобиться командам для того, щоб повторно впорядкувати послідовність та розподіл роботи, аби мінімізувати кількість крос-командних залежностей.

Якщо під час зустрічі з Планування Спринту елементи Беклога Продукту готові і їх можна вибрати з мінімальною кількістю залежностей, це означає, що протягом Спринту відбулась достатня кількість зустрічей з Удосконалення.

Артефакти Nexus

Артефакти відображають роботу або цінність для забезпечення прозорості та можливостей для перевірки і адаптації, як це описано в Scrum Guide.

Беклог Продукту

Існує лише один Беклог Продукту для цілого Nexus та всіх Scrum-команд. Власник Продукту відповідає за Беклог Продукту разом із його вмістом, доступністю та впорядкуванням.

У масштабі, Беклог Продукту слід розуміти на такому рівні, де залежності можна встановити та мінімізувати. Задля підтримки вирішуваності елементи Беклога Продукту часто подрібнюють на частинки, що називаються "тонко нарізана" функціональність. Елементи Беклога Продукту вважаються "готовими" до розгляду на Плануванні Nexus Спринту в тому випадку, коли Scrum-команди можуть їх розробляти за відсутності або ж з мінімальними залежностями від інших Scrum-команд.

Мета Nexus

Під час Планування Nexus Спринту формулюється мета для всього Спринту. Це називається Метою Nexus. Вона є сумою всіх робіт і Цілей Спринту окремих Scrum-команд у Nexus. Nexus повинен продемонструвати функціональність, яку розробили для досягнення Мети Nexus в Огляді Nexus Спринту.

Беклог Nexus Спринту

Беклог Nexus Спринту - це комбінація всіх елементів Беклога Продукту із Беклогів Спринту окремих Scrum-команд. Він використовується, щоб підкреслити залежності і виділити процес роботи протягом Спринту. Він оновлюється як мінімум щодня, часто в межах Щоденного Nexus Scrum.

Інтегрований Інкремент

Інтегрований Інкремент є сумою всіх інтегрованих робіт, виконаних за допомогою Nexus. Інтегрований Інкремент має бути здатним до використання і потенційного випуску. Це означає, що він повинен відповідати визначенню критерію "Готово". Інтегрований Інкремент перевіряється в Огляді Nexus Спринту.

Прозорість Артефакту

Аналогічно до Scrum, що є його основним структурним елементом, Nexus ґрунтується на прозорості. Інтеграційна Команда Nexus працює із Scrum-Командами в межах Nexus та Організації для того, щоб забезпечити прозорість, яка є наочною в усіх артефактах, а також для того, аби інтегрований стан Інкременту був для всіх зрозумілим.

Рішення, які приймалися з огляду на стан артефактів Nexus є ефективними пропорційно до рівня прозорості цих артефактів. Неповна або часткова інформація призведе до прийняття хибних або частково невірних рішень. Наслідки таких рішень можна збільшувати у масштабі Nexus. Відсутність повної прозорості унеможливить впровадження ефективного Nexus, метою якого є зменшення ризиків та збільшення цінності.

Програмне забезпечення потрібно розробляти таким чином, аби можна було встановити залежності і врегулювати їх, перш ніж технічний борг перетвориться на неприйнятний. Тестування неприйнятного технічного боргу відбувається тоді, коли відбувається інтеграція, але відповіді на питання чи було врегульовано залежності, немає. У таких випадках неврегульовані залежності лишаються прихованими у кодї та тестувальній базї, що призводить до зниження загальної цінності продукту.

Визначення критерію «Готово»

Інтеграційна Команда Nexus відповідає за визначенню критерію "Готово", який застосовується до Інтегрованого Інкременту, розробленого протягом кожного Спринту. Усі Scrum-Команди в межах Nexus дотримуються цього визначення критерію «Готово». Інкремент набуває статусу «Готово» лише коли він готовий для використання та є потенційно готовим до релізу Власником Продукту.

Елемент Беклога Продукту можна вважати «Готовим» тоді, коли його функціональність була успішно додана до продукту та інтегрована в Інкремент. Усі Scrum-Команди несуть відповідальність за розробку та інтеграцію своєї роботи в Інкремент, що відповідає цим вимогам.

Окремі Scrum-Команди всередині себе можуть дотримуватись більш обмеженого та суворого визначення критерію «Готово», проте він не може бути більш вільним у трактуванні, ніж було узгоджено для Інкременту.

Заключне слово

Nexus є безкоштовним продуктом, його принципи пояснюються у цьому посібнику. Так само як і в структурі Scrum, ролі, артефакти, події і правила Nexus є незмінними. І хоча можна застосовувати лише деякі фрагменти Nexus, отриманий результат не буде власне Nexus.

Подяка

Nexus та Масштабований Професійний Scrum були розроблені Кеном Швабером, Девідом Деймом, Річардом Хандхаузенем, Патрицією Конг, Робом Махером, Стівом Портером, Крістіною Швабер і Гюнтером Верхейеном.

Переклад

Цей посібник було перекладено з англійської мови оригіналу, створеного особами, зазначеними вище. Над перекладом працювали Bogdan Misyura і команда BrainRain.com.ua