

УДК 004.42

Віктор Кіфер, Оксана Цушко, Юрій Проців, Віталій Бревус

Тернопільський національний технічний університет імені Івана Пулюя, Україна

СКРАМ - ЕФЕКТИВНИЙ ПІДХІД ПРИ РОЗРОБЦІ ПЗ

Victor Kifer , Oksana Tsushko, Yuriy Protskiv, Vitaly Brevus

SCRUM – POWERFUL SOFTWARE DEVELOPMENT APPROACH

Ефективність діяльності підприємств значно залежить від їх організаційної структури. Сьогодні кожна ІТ-компанія шукає оптимальні методи організації робочого процесу, щоб за мінімальний час отримати максимальну ефективність роботи персоналу, а, отже, отримувати готове ПЗ швидше та якісніше. Це, у свою чергу, приносить більші доходи для бізнесу.

Скрам (Scrum) – підхід у гнучкій розробці ПЗ (agile software development), суть якого полягає у тому, що команда організовується і керується самостійно, контроль над нею зі сторони старших менеджерів обмежений. При такому процесі розробки використовується ітераційна модель. Ітерації, що називаються спринтами, проходять в чітко визначені і короткі часові терміни (до одного місяця). Згідно з правилами скраму, розробленими Nokia, після закінчення кожного спринту код має перевірятись відділом якості.

Scrum має 3 ролі: скрам мастер (scrum master), власник продукту (product owner) та команда розробників (team). Скрам мастер – найважливіша роль у скрам команді. Він є єдиним пунктом зв'язку між менеджером і командою. До його основних обов'язків відносять: створення атмосфери довіри та легкої комунікації у команді, виявлення та усунення будь-яких перешкод під час розробки. Він веде щоденні зустрічі команди і стежить за прогресом у виконанні проекту. Іноді керівник може допомагати власнику у формуванні журналу продукту. Власник продукту – людина, що відповідає за розробку продукту. Це завжди одна людина і саме вона приймає остаточні рішення для команди. До його основних обов'язків входить: посередництво між командою і замовником та формування і надання пріоритетів в журналі продукту. Команда є багатфункціональною, оскільки до неї входять розробники, аналітики, тестери, але члени команди не обмежені діяльністю лише в одній з цих областей. Вони беруть на себе обов'язки по розробці ПЗ і представленні його замовнику, оцінці елементів журналу спринту; прийнятті рішень щодо дизайну та імплементації.

Журнал продукту (Product backlog) – набір усіх бізнес та технічних вимог по функціональності ПЗ, що формується на основі вимог клієнта та зі сторони розробників. Записи у резерві проекту відсортовані згідно важливості. Важливість визначається певним натуральним числом. Переважно сусідні записи не мають пріоритети 5 і 6, бо передбачається, що може виникнути потреба вставити новий пункт з пріоритетом, вищим за перший пункт і нижчим за другий, а 5,5 задавати не зручно. Іноді записи тут розбиті за категоріями, на зразок: інтерфейс, функції, дефекти, бази даних... Пункт журналу продукту формується за загальним виглядом: «Я, як < користувач > хочу зробити <дію>, що призведе до <результату>», що дає змогу технічним вимогам бути зрозумілими як для власника, так і для розробників.

Журнал спринту (Sprint backlog) – частковий набір технічних вимог по функціональності ПЗ, що формується на основі вибірки з журналу продукту, і призначений для реалізації на певній стадії ітерації. Команда розробників на щоденних нарадах (Daily Scrum Meeting), що тривають не більше ніж 15 хв, стежить за діаграмою виконання журналу спринту (Burndown chart) і відповідно розподіляє аспекти роботи команди на весь робочий день. Спринт може бути зупинений в критичних ситуаціях командою, якщо вона не може виконати поставлені перед нею роботи, або власником, якщо зникає необхідність у розробці задач, перелічених у спринті.

Діаграма виконання журналу спринту (Burndown chart) – оновлюється щоденно і призначена для графічного представлення стану робіт над спринтом. Вона показує скільки задач, поставлених перед командою, вже виконано і скільки залишається.

Нарада з планування спринту (Sprint Planning Meeting) відбувається перед кожною новою ітерацією і триває 4-8 годин. Спочатку команда і власник продукту обирають задачі із журналу продукту, а згодом сама команда обговорює технічні деталі реалізації і заносить їх у журнал спринту.

Демонстрація (Demo Meeting) проходить в кінці ітерації і є презентацією усіх нових функцій у ПЗ. У якості глядачів постають усі зацікавлені особи, а кожен член команди показує, що зробив за спринт. Може тривати до 4 год.

Ретроспектива (Retrospective Meeting) – зустріч, на якій члени команди висловлюють свої думки щодо минулого спринту, зокрема, про його добрі аспекти, та про можливі покращення. Триває 1-3 год.

Серед переваг скраму варто відзначити, що він є доволі простим підходом. Він дозволяє легко вносити зміни, доповнення у план розробки проекту. Завдяки вільній від строгого контролю команді досягається можливість впровадження нових ідей розробки. А замовник може швидко отримувати потрібні йому результати і вносити власні поправки, якщо вважає це за потрібне. Іншою перевагою, і в той же час недоліком, скраму є те, що його потрібно пристосовувати до кожної окремої ситуації.

Однією з перших гнучкий підхід управління проектами застосувала ІВМ у 1980р. коли створювала ІВМ РС. Причиною цього були попередні невдачі компанії та випуск у 1976р. комп'ютера від Apple, які показали, що проблема створення такого ПК полягає не у складності проекту, а в неправильному керівництві, обмежені свободи творчості робочого колективу. Сформувавши команду із 12 чоловік та усунувши вплив менеджерів на них впродовж спринту, приблизно за рік ІВМ вдалось випустити популярний ІВМ РС. Це довело ефективність скраму. Ще більш красномовним є те, що після відмови від нього в ІВМ почались серйозні проблеми. Сьогодні, 30 років по тому, ІВМ знову повернулася до гнучкого управління проектами. Окрім ІВМ, скрам використовується і в інших великих ІТ-компаніях, таких як Google, Facebook, Nokia. Застосовують його і малі компанії, невеликі групи розробників ПЗ. Отже, скрам став основним підходом керівництва проектами в області розробки комп'ютерних програм на даний момент.