



Ю. І. Грицюк, О. А. Немова

Національний університет "Львівська політехніка", м. Львів, Україна

ОСОБЛИВОСТІ УПРАВЛІННЯ ПРОЦЕСОМ РОЗРОБЛЕННЯ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Наведено технологічні особливості управління процесами розроблення вимог до ПЗ та внесення змін до них, методи і засоби яких дають змогу проконтролювати стан їхнього виконання та встановити статус кожної з вимог протягом усіх етапів реалізації програмного проекту. Проаналізовано основні поняття про план виконання завдань програмного проекту та управління процесом його реалізації, що допомогло зрозуміти суть пропонованих методів і засобів здійснення контролю за станом виконання кожного із завдань та відстеження їхнього статусу на усіх етапах реалізації програмного проекту. Виявлено проблеми, які виникають під час управління процесом розроблення вимог до ПЗ, що дає змогу запобігти непередбачені втрати і неприємні інциденти, які можуть виникнути на деяких етапах реалізації програмного проекту. З'ясовано, що більшість виявлених проблем пов'язані насамперед з: плануванням послідовності виконання робіт; контролем за станом виконання робіт; управлінням змінами вимог до ПЗ; відстеженням статусу кожної вимоги. Характер проблем, з якими стикаються ІТ-компанії при управлінні вимогами до ПЗ, залежить також і від специфіки їхньої діяльності. Досліджено деякі особливості управління вимогами до ПЗ в процесі діяльності компанії його замовника, виконавця та виробника, що дає змогу своєчасно підготувати відповідні плани виконання завдань програмного проекту та ефективно управляти процесом їх реалізації. Встановлено, що на початкових етапах розроблення користувацьких вимог до ПЗ процедура внесення змін має мати незначний ступінь формалізації, яка повинна постійно збільшуватися в міру просування етапів реалізації програмного проекту. Зроблено відповідні висновки та надано рекомендації щодо практичного використання запропонованих методів і засобів управління процесом розроблення вимог до ПЗ.

Ключові слова: програмний проект; вимоги до програмного забезпечення; визначення вимог; розроблення вимог; управління вимогами; рецензування вимог; специфікація вимог; управління процесом розроблення вимог до ПЗ.

Вступ. Під вимогою (англ. *Requirement*) зазвичай розуміють будь-яку умову, якій має відповідати продукт, що потрібно виготовити чи розробити. Вимогою може бути деяка можливість, якою має володіти цей продукт, і деяке обмеження, якому він має задовольняти. Водночас, під вимогою до ПЗ (англ. *Software Requirements*) будемо розуміти функціонал, якому мають відповідати результати реалізації програмного проекту.

Управління вимогами (англ. *Requirements Management*) – це процес цілеспрямованого розроблення вимог до продукту, їхнього кваліфікованого аналізу, узгодження із замовником і затвердження керівництвом, встановлення пріоритету реалізації та відстеження стану виконання. Тут також приймають пропозиції щодо внесення змін у вимоги, проведення аналізу цих змін і узгодження з виконавцями, здійснення контролю за станом виконання та доведення цих змін до зацікавлених сторін. Загалом, це безперервний процес, який проходить протягом всіх етапів реалізації програмного проекту, зазвичай, згідно з лівостороннім логнормальним законом розподілу (Bronstein, & Semendiaev, 1986) кількості внесених змін.

Мета управління вимогами до ПЗ полягає в тому, щоб забезпечити процес його розроблення згідно з потребами і очікуваннями замовника, а також задовольнити умови співпраці з внутрішніми і зовнішніми зацікавленими сторонами (Stelman, & Greene, 2005). Процес управління вимогами до ПЗ починають з аналізу предметної області, виявлення цілей його застосування чи використання, а також врахування обмежень технологічних операцій на різних етапах реалізації програмного проекту. Управління ж змінами вимог додатково містить підтримку відповідної процедури планування цих змін, інтегрування користувацьких вимог з системними, вхідних вимог з похідними, а також взаємну організацію роботи усіх зацікавлених сторін з безпосередніми виконавцями проекту.

Управління процесом розроблення вимог до ПЗ (англ. *Process Management of Developing Software Requirements*) передбачає взаємодію членів команди програмного проекту з зацікавленими сторонами, а також їхню адаптацію до потреби внесення змін у вимоги та відстеження стану їхнього виконання протягом всіх етапів реалізації програмного проекту (Project, 2008).

Інформація про авторів:

Грицюк Юрій Іванович, д-р техн. наук, професор, кафедра програмного забезпечення. Email: yurii.i.hrytsiuk@lpnu.edu.ua; <https://orcid.org/0000-0001-8183-3466>; ResearcherID: V-3995-2017

Немова Олена Андріївна, студентка, кафедра програмного забезпечення. Email: olena.nemova@gmail.com

Цитування за ДСТУ: Грицюк Ю. І., Немова О. А. Особливості управління процесом розроблення вимог до програмного забезпечення. Науковий вісник НЛТУ України. 2018, т. 28, № 8. С. 161–169.

Citation APA: Hrytsiuk, Yu. I., & Nemova, E. A. (2018). Management Features Process of Developing Software Requirements. *Scientific Bulletin of UNFU*, 28(8), 161–169. <https://doi.org/10.15421/40280832>

Щоб запобігти перетину області дії одного набору вимог з іншим, між ними мають бути постійні зв'язки, інколи – критичні (Hull, Jackson, & Dick, 2005). Наприклад, під час розроблення ПЗ у зацікавлених представників бізнесу можуть бути настільки переконливі аргументи щодо врахування їхніх меркантильних потреб, що аналітики можуть інколи проігнорувати ключові потреби замовників від роботи майбутнього ПЗ, або вважатимуть, що створені ними сценарії використання покриють обидва набори вимог відповідними зв'язками.

Аналіз останніх досліджень та публікацій. Проблемі управління процесом розроблення вимог до ПЗ присвячено багато робіт українських й іноземних учених, а саме: С. Джонса (Jones, Bonsignour, 2012), Р. Фатрелла (Futrell, Shafer & Shafer, 2002), К. Лаврішевої (Lavrishcheva, 2013), Д. А. Маєвського (Maievskiy & Kozina, 2015), В. Яковини (Yakovyna, 2012), Г. Майерса (Myers, Badzhett & Sandler, 2012), С. МакКоннелла (McConnell, 2013), В. Міщенко і О. Поморової (Mishhenko, Pomorova, Novorushchenko, 2012), О. Медче (Maedche, Botzenhardt & Neer, 2012), І. Соммервілла (Sommerville, 2002), В. Харченка і Б. Конорева (Kharchenko et al., 2012; Konorev et al., 2009), О. Харченка (Kharchenko, Galay & Yatsyshyn, 2011).

Проте, багато науковців (Matciashak, 2002; Sommerville, 2002; Kornipaev, 2013; Laplante, 2009; Wieggers, 2003) вважають, що управління процесом розроблення вимог до ПЗ, особливо управління їхніми змінами, а також наявні методи і засоби забезпечення якості цих процесів залишаються незабезпеченими фундаментальною теорією та ефективною методологією. Практично усі дослідження щодо управління процесом розроблення вимог та їхніми змінами мають хаотичний, несистематизований характер (Berenbach et al., 2009; Kroll & Krachten, 2004). Водночас, як доведено у роботах (Cockburn, 2001; Leffingwell & Widrig; Mansilla et al., 2013; Sutcliffe, 1998), саме не ефективне управління процесами розроблення вимог та їхніми змінами може призвести до появи від 26 до 42 %% всіх недоліків майбутнього програмного продукту. Безумовно, є багато фундаментальних досліджень з інженерії ПЗ (Novorushchenko, 2018; Pomorova & Novorushchenko, 2013; Lavrishcheva, 2013, Kharchenko & Yatsyshyn, 2009 та ін.), але відсутня завершена, протестована та апробована теорія та методологія управління процесами розроблення вимог до ПЗ та їхніми змінами, а також методи і засоби побудови відповідних процедур, які можна було б застосовувати для контролю за станом виконання цих змін і відстеження статусу відповідних вимог. Тому проблема управління процесом розроблення вимог до ПЗ потребує проведення нагальних досліджень для запобігання непередбачених втрат і неприємних інцидентів, які можуть виникнути під час реалізації програмного проекту (Hrytsiuk, 2018).

Не претендуючи на кардинальні зрушення в сучасних технологіях розроблення вимог до ПЗ, спробуємо внести й свою лепту в деякі підходи до процесу управління змінами вимог. Тому, як на сьогодні, видається нам актуальним дослідження, яке стосується розроблення методів і засобів управління процесом розроблення вимог до ПЗ, особливо тих процедур, які стосуються внесення змін до них, контролю стану їх виконання та відстеження статусу цих вимог.

Об'єкт дослідження – управління процесом розроблення вимог до ПЗ.

Предмет дослідження – методи та засоби управління процесом розроблення вимог до ПЗ, які дадуть змогу здійснити контроль за станом їх виконання та встановити статус кожної з них на всіх етапах реалізації програмного проекту.

Мета дослідження полягає у встановленні технологічних особливостей управління процесами розроблення вимог до ПЗ та внесення змін до них, методи і засоби яких дадуть змогу проконтролювати стан їх виконання та встановити статус кожної з вимог протягом усіх етапів реалізації програмного проекту.

Для реалізації зазначеної мети потрібно виконати такі основні завдання:

- 1) з'ясувати основні поняття про план виконання завдань програмного проекту та управління процесом його реалізації, що допоможе зрозуміти суть пропонованих методів і засобів здійснення контролю за станом їх виконання та відстеження статусу;
- 2) виявити проблеми, які виникають під час управління процесом розроблення вимог до ПЗ, що дасть змогу запобігти непередбачені втрати і неприємні інциденти, які можуть виникнути під час деяких етапів реалізації програмного проекту;
- 3) проаналізувати особливості управління вимогами до ПЗ в процесі діяльності компанії його замовника, виконавця та виробника, щоб згодом підготувати відповідні плани виконання завдань програмного проекту та ефективно управляти процесом його реалізації;
- 4) зробити відповідні висновки та надати рекомендації щодо практичного використання запропонованих методів і засобів управління процесом розроблення вимог до ПЗ.

1. План виконання завдань програмного проекту та управління процесом його реалізації

Управління процесом розроблення вимог до ПЗ практично не відрізняється від управління будь-якими іншими виробничими чи технологічними процесами. Перед тим, як розпочати будь-який програмний проект, його керівнику необхідно зрозуміти, що саме потрібно отримати замовнику на виході внаслідок його реалізації. Насамперед йому потрібно з'ясувати характер робіт, які необхідно виконати команді розробників ПЗ, щоб вчасно отримати очікуваний замовником результат. Також керівнику проекту потрібно зрозуміти, чи існуватимуть будь-які взаємозв'язки між окремими завданнями, які стосуються реалізації програмного проекту. Наприклад, чи можна наступне завдання почати виконувати тільки після завершення попереднього або виконувати їх паралельно чи, можливо, з деяким зміщенням у часі. Окрім цього, йому також потрібно з'ясувати, якими професійними навиками мають володіти члени команди розробників ПЗ, що виконуватимуть ці завдання.

Вважається, що керівнику проекту під час складання плану виконання завдань доцільно звернути увагу членам команди саме на тому очікуваному замовником результаті, який потрібно отримати після виконання кожного завдання. Оскільки кожен поточний результат має бути наочним і вимірним, то це забезпечить реальне підтвердження його виконання та, як наслідок, успішну реалізацію всього програмного проекту.

Грунтуючись на всій цій інформації, керівник проекту має розробити такий план послідовності дій, який

міститиме усі завдання програмного проекту, їхніх безпосередніх виконавців й термінів їхнього завершення. На початку реалізації розробленого плану дій керівник проекту, роздавши завдання членам команди розробників, має контролювати як стан їхнього виконання, так і відстежувати статус кожного з них.

Якщо уявити ідеальний механізм, у якому все має проходити згідно з узгодженим планом дій, то програмний проект, наприклад, міський трамвай має безпомилково слідувати встановленому маршруту, щоб до призначеної кінцевої зупинки довести навіть одного пасажира у призначений час, дотримуючись як графіку маршруту, так і задовольняючи його бажання. Стосовно ж ПЗ, то тут потрібно отримати готовий програмний продукт, який би задовольняв, насамперед, потреби його замовника, так і усі зацікавлені сторони, причетні до його виконання, не виходячи за межі бюджету та терміну завершення проекту. Однак, якими б сукупними зусиллями усі учасники проекту та зацікавлені сторони не намагалися цього досягти, у реальному житті все істотно відрізняється від ідеалу, зазвичай, з таких причин:

- надзвичайно важко точно оцінити тривалість реалізації завдань проекту та потенційні трудовитрати, потрібні для їх виконання. Вкрай рідко трапляються ситуації, коли керівник проекту, маючи великий досвід управління, може достеменно визначити тривалість реалізації завдань проекту наявними у його розпорядженні людськими ресурсами;
- у ході виконання потрібних робіт над програмним проектом можуть виявитися такі труднощі чи форс-мажорні обставини, які не можливо було завчасно передбачити. Наприклад, частину плану дій було складено з розрахунку, що ключовий працівник виконуватиме деякі важливі завдання у певний період часу, але виявилось, що з різних причин він на той момент недоступний – залучений до виконання дещо важливішого проекту.

Подібні ситуації часто призводять до відхилення реальних подій від початкового плану дій, які, вочевидь, на кожному етапі реалізації програмного проекту потребуватимуть його перегляду. Якщо ж план послідовності дій переглянуто, узгоджено і затверджено, то все може повторитися знову знову ж таки з різних причин. Постійні коригування плану дій призводять до того, що початкова вартість проекту поступово збільшується, терміни реалізації його завдань переносяться на пізніше і, як наслідок, все поступово й неминуче йде якщо не до провалу, то до конкретного незадоволення, насамперед, замовника ПЗ, усіх зацікавлених сторін і, особливо, інвесторів. Виходом з цієї ситуації могло б стати залучення в команду виконавців професійних працівників або заміна керівника проекту фаховим кризис-менеджером (Bevz, Papinov, & Skadan, 2010).

Однією з альтернатив може стати домовленість виконавця ПЗ і з його замовником, згідно з якою терміни завершення етапів реалізації програмного проекту і його вартість залишаються колишніми, однак зменшується кількість поставлених завдань чи обсяг виконуваних робіт. Такий підхід у деяких ситуаціях дає позитивний результат, насамперед це стосується компаній-виконавців проекту. Наприклад, якщо для компанії-виконавця принципово важливо вийти на ринок з новим програмним продуктом до певної дати і при цьому не вийти за межі встановленого бюджету (позаяк замовник не може дозволити собі витратити більше), то запланований функціонал, який мав би мати програмний продукт, можна дещо скоротити. Часто виконавцю проекту

потрібно реалізувати й продемонструвати замовнику тільки перелік можливостей майбутнього ПЗ для того, щоб воно хоча б чимось відрізнялося від попереднього, яким був вельми незадоволений замовник. Це може слугувати прикладом того, як економічні важелі та ділові стосунки є головною рушійною силою не тільки будь-якого проекту, в т.ч. і програмного, а й наукового чи технологічного прогресу загалом.

Також керівнику проекту важливо усвідомлювати ще й те, що будь-який програмний проект під час його реалізації характеризується такими чинниками:

- функціональними можливостями майбутнього програмного продукту;
- додатковою кількістю та вартістю виконуваних робіт внаслідок зміни початкового чи поточного плану дій;
- термінами виконання завдань програмного проекту та датою його завершення та передачі замовнику готового продукту проекту.

Ці чинники дуже тісно пов'язані між собою (див. рис. 1), позаяк будь-яка зміна хоча б одного з них призводить до зміни як мінімум ще одного чинника. Рисунок також демонструє той важливий момент, коли керівнику проекту для успішного просування програмного продукту потрібно постійно приймати як поточні, так перспективні рішення, кожне з яких заново позиціонує продукт проекту відносно цих трьох фундаментальних чинників.



Рис. 1. Взаємозв'язок функціональних можливостей програмного продукту, вартості його розроблення та термінів реалізації

Зазвичай, будь-який керівник проекту у своїх снах і мріях бачить, як кожне його рішення значно покращує можливості майбутнього програмного продукту, одночасно знижуючи вартість та терміни на його розроблення. Незважаючи на часту "провальність" програмних проектів, така практика діяльності багатьох їхніх керівників дуже поширена серед значної кількості компаній-виконавців ПЗ.

2. Проблеми управління процесом розроблення вимог до програмного забезпечення

Спробуємо проаналізувати ті специфічні проблеми, які роблять управління процесом розроблення вимог до ПЗ значно складнішим порівняно з управлінням будь-яким іншим виробничим чи технологічним процесами (рис. 2).

Перша проблема полягає в тому, що відомо не так вже й багато фахівців (аналітиків, архітекторів чи програмістів), які володіють достатнім досвідом управління вимогами до ПЗ. Пов'язано це з тим, що дуже мало компаній-виконавців мають хоча б якість організованого процес розроблення вимог до ПЗ, а ще менше знають,

як саме управляти цим процесом. Тому працівники, які в ході реалізації програмного проекту стикаються з потребою управляти вимогами до ПЗ, виявляються просто не готовими до виконання такої роботи. У такій ситуації керівнику проекту (наприклад, початківцю) неймовірно важко оцінити потрібний час і можливі трудовитрати, оскільки одним з основних моментів отримання такої достовірної оцінки якраз і є відповідний досвід його роботи або фахова робота його аналітиків.

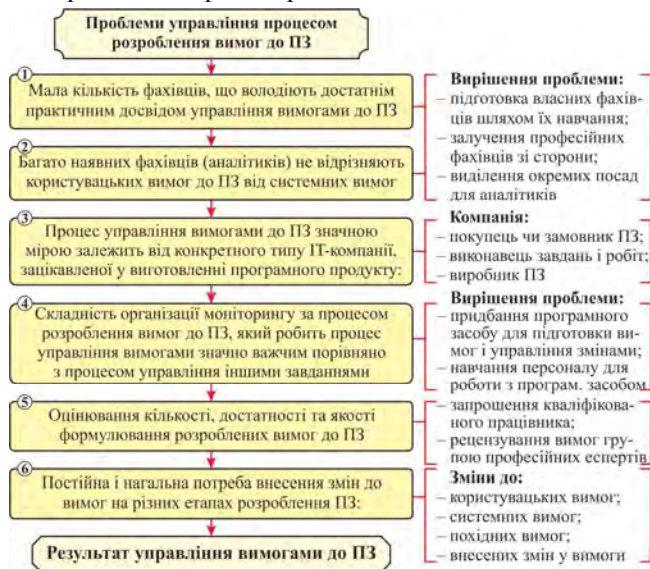


Рис. 2. Специфічні проблеми управління процесом розроблення вимог до ПЗ

Отже, за відсутності досвіду роботи керівника проекту, так і професійних навиків роботи багатьох членів його команди розробників уже з самого початку все виглядає надзвичайно примарно й безперспективно щодо виконання завдань проекту. Тут доречно згадати вислів Леонардо Да Вінчі про наявність проблеми: "Існує три різновиди людей: ті, хто бачать проблему загалом; ті, хто бачать проблему, коли їм її показують; ті, хто взагалі не бачать жодних проблем".

З наведеного вище можна сформулювати дещо повчальний висновок: якщо керівник проекту немає досвіду управління процесом розроблення вимог до ПЗ, то він може навіть і не здогадуватися про те, які завдання йому потрібно запланувати для того, щоб розробити хоча б якісь вимоги. Як свого часу висловився Марк Туллій Цицерон: "Непередбачене вражає набагато сильніше, ніж заплановане". Тому в роботі (Hrytsiuk, 2018) достатньо детально описано перелік тих завдань, які керівнику проекту потрібно виконати в процесі розроблення вимог до ПЗ різних типів.

Друга проблема полягає в тому, що багато фахівців, які займають так звані посади аналітиків, не завжди відрізняють користувацьких вимог до ПЗ від системних вимог. Вони часто не розуміють різниці між системними вимогами й системним моделюванням роботи об'єкта, перебігу процесу чи явища. Це призводить до того, що фахівці-програмісти відразу починають виробляти детальне технічне рішення щодо генерування програмного коду, замість того, щоб спочатку розробити вимоги до ПЗ, які часто навіть не залежать від деталей процесу кодування.

Третя основна проблема полягає в тому, що управління процесом розроблення вимог до ПЗ значною мірою залежить від конкретного типу ІТ-компанії, заці-

кавленої в успішному виготовленні програмного продукту. Однак, сама процедура формування набору вимог до ПЗ та управління процесом їх розроблення, а також їхніми взаємозв'язками вирізняються між собою залежно від типу ІТ-компанії, що безпосередньо впливає на технологічні операції кожного з етапів реалізації програмного проекту.

Існують такі основні типи ІТ-компаній:

- **Компанія-замовник** – зацікавлена в придбанні ПЗ для вирішення власних потреб. Її головним завданням є підготовка власних користувацьких вимог до ПЗ, які згодом будуть використовувати для приймання релізів ПЗ чи остаточної його версії. Така компанія також може вносити зміни до вимог і узгоджувати їх з компанією-виконавцем.
- **Компанія-виконавець** – відповідає на запити компанії-замовника або компанії-виконавця вищого рівня (генпідрядника). Така компанія, зазвичай, отримує вхідні вимоги до ПЗ та на їх підставі розробляє свої похідні вимоги, а також підтримують процес управління змінами до них. Важасться, що компанія-виконавець може також виступати в ролі компанії-замовника, придбавши деякі підсистеми або компоненти ПЗ нижчих рівнів. Однак, це вже трохи інша форма діяльності компанії, позаяк у цьому випадку компанія-виконавець, зазвичай, зацікавлена в придбанні якісної специфікації вимог до ПЗ, а не його самого.
- **Компанія-виробник** – компанія, яка власноруч розробляє та продає масовий програмний продукт для виконання, наприклад, векторної чи растрової графіки. Така компанія формує набори користувацьких вимог для своїх продуктів, насамперед, під впливом кон'юнктури ринку, ніж спілкуючись з конкретними його користувачами або компаніями-замовниками. Збиранням вимог у такій компанії, зазвичай, займається відділ маркетингу, у штаті якого знаходяться кваліфіковані бізнес-аналітики та маркетологи. Компанія-виробник розробляє програмний продукт відповідно до ринкових (кон'юнктурних) вимог і потім продає його багатьом зацікавленим замовникам.

За своєю сутністю, компанія-виробник може одночасно виступати як у ролі замовника, так і в ролі виконавця, проте стосунки між різними відділами, які виконують ці ролі всередині компанії, відрізняються від стандартних взаємин між звичайною компанією-виконавцем чи компанією-замовником.

Четверта проблема полягає в труднощах реалізації моніторингу за процесом розроблення вимог до ПЗ, що робить процес управління вимогами значно важчим, ніж процес управління іншими завданнями. Одне з важливих і, водночас, найважчих завдань тут полягає в тому, що керівнику проекту потрібно з'ясувати, чи є наявний у нього перелік вимог до ПЗ *повним (достатнім)*, чи входять у сформовані набори вимог ключові вимоги, без яких майбутнє ПЗ не варте будь-якої уваги. Йдеться насамперед про те, щоб керівник проекту зміг визначити той момент, коли йому варто зупинити процес розроблення вимог до ПЗ. Для цього йому потрібно правильно визначити частку вже виконаних робіт з формування наборів вимог до ПЗ та встановити ті моменти, коли діяльність аналітика ще доволі далека від завершення, чи, навпаки, близька до повного виконання. Також йому потрібно залучити кваліфікованих експертів для рецензування зібраних вимог і визначення серед них наявності ключових вимог.

П'ята проблема полягає в тому, що потрібно оцінювати якість розроблених вимог до ПЗ. Можливо, до певного моменту аналітиком(ами) розроблено достатньо багато користувацьких вимог, але як керівник

проекту може переконатися в тому, що кожна з цих вимог добре і правильно сформульована? Чи може він бути впевненим у тому, що кожна з наявних вимог до ПЗ є унікальною, а всі вони разом є достатніми для успішної реалізації програмного проекту? Одним із можливих виходів з цієї ситуації є введення в команду виконавців кваліфікованого працівника (експерта) з досвідом виконання аналогічних програмних проектів. Також можна запровадити рецензування вимог кваліфікованими експертами на предмет їх достатності, цілісності та якості формулювання.

Завершальною *шостою проблемою* є постійна й нагальна потреба внесення змін у вимоги на різних ієрархічних рівнях: змін до користувацьких чи системних вимог; змін до похідних вимог; змін до внесених змін і т.д. Управління вимогами – це, передусім, управління змінами та контроль стану їхнього виконання. Зазвичай, кожна запропонована зміна залучає до перегляду в найкращому випадку одну або декілька вимог нижчого рівня, а в гіршому випадку – з десяток вимог на різних ієрархічних рівнях. Однак, вплив будь-якої зміни на ту чи іншу вимогу часто надзвичайно важко оцінити, а без цього неможливо передбачити, як саме ця зміна вплине на функціональні можливості майбутнього програмного продукту, на його остаточну вартість і, що найважливіше, на терміни його виконання.

Отже, основні проблеми, що виникають при управлінні процесом розроблення вимог до ПЗ, пов'язані з: плануванням послідовності виконання робіт; контролем за станом виконання робіт; управлінням змінами вимог до ПЗ; відстеженням статусу кожної вимоги. Характер проблем, з якими стикаються ІТ-компанії при управлінні вимогами до ПЗ, залежить також і від специфіки їхньої діяльності. Тому, нижче спробуємо детально описати різні особливості процесу розроблення вимог до ПЗ у тих компаніях, які було згадано вище.

3. Управління вимогами до ПЗ в процесі діяльності компанії-замовника

Для аналізу особливостей процесу управління вимогами до ПЗ у компанії-замовника потрібно розглянути такі основні питання: з'ясувати особливості планування послідовності виконання робіт; виявити основні проблеми, пов'язані з контролем за ходом виконання робіт; встановити складові процесу управління змінами вимог до ПЗ.

1. Планування послідовності виконання робіт.

Початковим моментом реалізації програмного проекту в компанії-замовника є визначення основної *концепції програмного продукту*. Зазвичай, концепція продукту – це щось дещо детальніше описане і доволі добре сформульоване, щоб споживачі хотіли його придбати чи інвестори забажали фінансувати процес його виготовлення. Щонайменше – це може бути просто якась програмна ідея чи схематична модель майбутнього ПЗ. У будь-якому випадку потрібна їхня формалізація з дуже простотою причини: програмний проект, перш ніж стартувати, має бути зі всіма зацікавленими сторонами погоджений і затверджений керівництвом. А для його затвердження потрібен, вочевидь, деякий документ, який доводить, що компанія-виконавець не дарма витратить свій час і ресурси на його реалізацію. Такий документ, зазвичай, містить короткий перелік основних можливостей майбутнього ПЗ, тобто перелік потреб користувачів і

опис того, які переваги отримає компанія-замовник внаслідок придбання програмного продукту.

Інформація, яка міститься в документі, детально описує концепцію процесу реалізації ПЗ, тобто дає змогу його керівнику почати планувати послідовність виконання робіт, прогнозувати перелік подальших дій: як своїх, так і членів команди виконавців. Оскільки в концепції проекту вже міститься опис того, що саме "користувачі повинні мати можливість робити", то компанія-замовник із самого початку отримує попередній перелік зацікавлених сторін (потенційних користувачів) в успішній реалізації проекту, плюс один або декілька сценаріїв використання майбутнього ПЗ (тобто, можливостей робити щось таке, чого не роблять інші програми, відомі на сьогодні).

Першим кроком керівника проекту при розробленні плану дій є складання детального переліку типів зацікавлених сторін, а також детального набору сценаріїв використання майбутнього ПЗ, що покривають увесь перелік можливостей, очікуваних від нього, в т.ч. і режими його аварійного (розрахованого на дурня) функціонування. Після цього можна переходити до детального планування тих дій, які вказуватимуть, як саме потрібно формувати набори вимог до ПЗ (рис. 3).

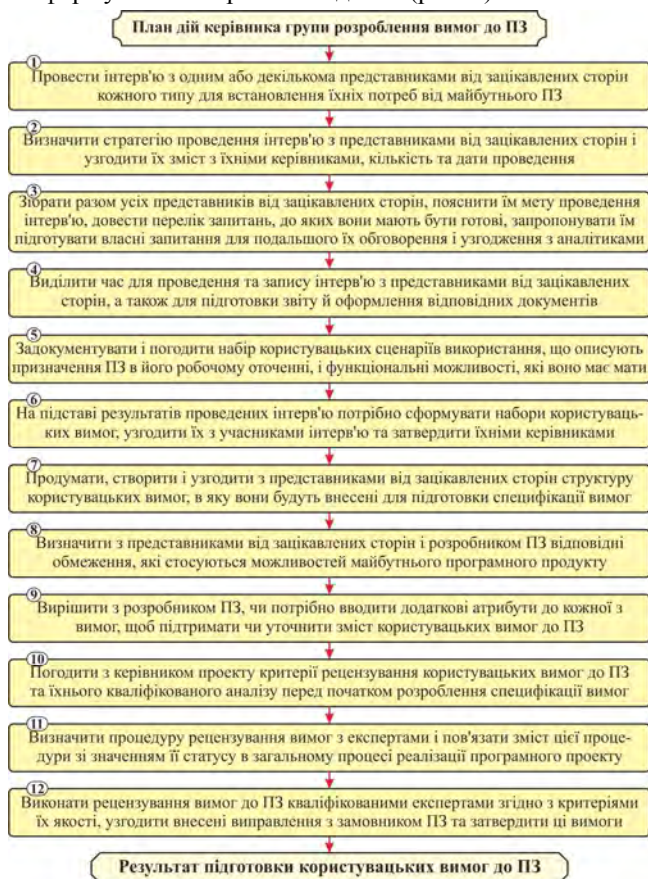


Рис. 3. План послідовності виконання робіт для отримання користувацьких вимог до ПЗ

В цьому плані дій заслуговує уваги процедура рецензування вимог відповідними експертами і взаємний зв'язок змісту цієї процедури зі статусами вимог в загальному процесі реалізації програмного проекту. Приклад процедури рецензування вимоги подано на рис. 4 у вигляді діаграми переходів станів статусу вимоги до ПЗ. З рисунку видно, що початковим статусом вимоги є значення "Запропоновано". Вимогу в цьому статусі потрібно перевірити аналітиком і тільки після

позитивного результату вона може перейти в статус "Перевірено". Потім цю вимогу оцінює та затверджує керівництво, після чого вона набуває статусу "Узгоджено". Вважається, що в будь-який момент вимога, яка була в статусі "Активна", може змінити свій статус на "Відхилена" чи "Виконана". Важливо, щоб критерії перевірки цієї вимоги на кожному кроці можливої зміни її статусу були обов'язково визначені заздалегідь.

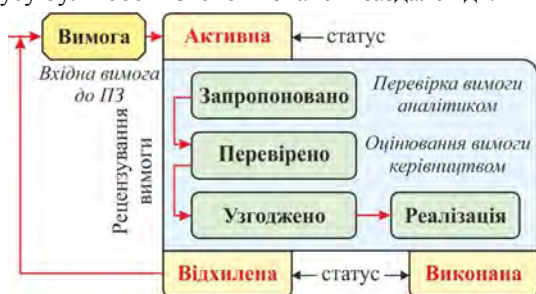


Рис. 4. Схема переходів стану статусу користувацької вимоги

Отже, наведений вище план послідовності виконання робіт для розроблення користувацьких вимог до ПЗ передбачає потребу прийняття керівником проекту цілого ряду організаційних рішень. Підготовка та обґрунтування цих рішень входить до обов'язків керівника проекту, оскільки він має узгодити цей план з керівництвом зацікавлених сторін і основним замовником ПЗ.

Всі обмеження, які стосуються планування послідовності виконання робіт, потрібно оцінити з позиції їх доцільності та здійсненості. Наприклад, зацікавлені сторони, як завжди, наполягатимуть на введенні ПЗ в експлуатацію в дуже стислі терміни і з низькою собівартістю виконання робіт, хоча реально виконати таке протисто неможливо (див. рис. 1), що і потрібно оцінити керівнику проекту.

2. Контроль за ходом виконання робіт. Зазвичай, контроль за ходом виконання робіт можна починати відразу ж після затвердження плану послідовності дій. Очевидними точками контролю є кінцеві дати виконання проміжних кроків цього плану. На більш ранніх кроках виконання робіт реалізація проміжних завдань пов'язана з підготовкою і проведенням інтерв'ю, а також документуванням їх результатів. Ці завдання доволі просто контролювати в ході їх виконання.

Для успішного моніторингу за всіма іншими діями керівнику проекту можуть допомогти такі основні контрольні точки:

- з'ясування потреб зацікавлених сторін від майбутнього ПЗ;
- формування наборів користувацьких вимог до ПЗ;
- визначення структури специфікації вимог до ПЗ;
- визначення атрибутів кожної з вимог до ПЗ;
- формулювання вимог до ПЗ згідно з встановленими шаблонами;
- визначення процедури рецензування вимог до ПЗ згідно з переліком критеріїв їх якості.

Якщо структуру специфікації вимог до ПЗ сформульовано, то стає доволі легко визначити, де вимоги вже сформульовано, а де в структурі (в документі) поки що залишаються незаповнені місця. Такі "дірки" можна легко знайти й вжити заходів для їх усунення. Якщо набір атрибутів для кожної вимоги визначено, то вже набагато легше контролювати хід заповнення цих атрибутів відповідними значеннями.

Під час рецензування вимог до ПЗ кваліфікованими експертами теж доволі просто здійснювати їхній моніторинг, використовуючи для цього результати переві-

ки вимог на відповідність прийнятним критеріям, а також контролювати їхній поточний статус.

3. Процес управління змінами вимог до ПЗ. На початкових етапах розроблення користувацьких вимог до ПЗ обов'язково присутній період швидких і інтенсивних змін цих вимог як з боку замовників, так і з боку виконавців програмного проекту. Очевидно, протягом цього періоду немає значних підстав вводити формальну процедуру управління змінами, оскільки ситуація зі станом вимог на цих етапах дуже динамічна й зміни у вимоги можуть як вносити, так і приймати під час їх виникнення, тобто "на ходу". Однак, завдання керівника проекту полягає ще й в тому, щоб вчасно визначити момент, коли період внесення змін у вимоги поступово стабілізується й настає потреба вводити формальну процедуру управління вимогами до ПЗ. Зазвичай, цей момент настає тоді, коли всі вимоги пройшли повний цикл рецензування та перейшли в статус "Узгоджено" (див. рис. 4).

Управління змінами вимог до ПЗ є життєво необхідним процесом розроблення як користувацьких, так і системних вимог, в т.ч. й похідних вимог на різних рівнях. Рівень формальності використання процедури внесення змін у вимоги залежить від етапу, на якому в цей момент може знаходитися реалізація програмного проекту. Тут важливо взяти під контроль такі основні етапи:

- визначені користувацькі вимоги використано в тендері для конкурентного вибору виконавця програмного проекту;
- укладання контракту з компанією-виконавцем щодо процесу розроблення ПЗ;
- завершення процедур розроблення специфікацій користувацьких і системних вимог до ПЗ і початок розроблення програмного коду;
- виконання приймальних тестів і реалізація експлуатаційних випробувань ПЗ;
- експлуатація ПЗ в промислових умовах.

Цей перелік етапів, на яких може знаходитися програмний проект, визначає деяку послідовність контрольних точок, на кожній з яких відповідальність керівника проекту за раніше прийняті рішення поступово збільшується. Отже, що далі просувається програмний проект, то вищого рівня набуває формальність процедури внесення змін у вимоги до ПЗ, і тим більшими мають ставати витрати, пов'язані з реалізацією запропонованих змін.

Вважається, що на якому б етапі реалізації не перебував програмний проект, процес управління вимогами до ПЗ передбачає обов'язкове виконання таких кроків:

- 1) зареєструвати запропоновану зміну вимоги до ПЗ;
- 2) визначити вплив запропонованої зміни на всі вимоги верхнього й нижнього рівнів, встановити вартість внесених змін і потрібні для цього ресурси;
- 3) вирішити – прийняти або відхилити запропоновану зміну вимоги до ПЗ – і, якщо прийняти, то:
- 4) визначити момент, коли потрібно реалізовувати запропоновану зміну вимоги, а також всі вимоги верхнього і нижніх рівнів.

На *першому кроці* запропонована зміна має містити обґрунтування потреби її внесення (причину) і посилання на вимогу (можливо, на декілька), яка має бути змінена, видалена або додана. Потрібно, щоб ім'я ініціатора (автора), який вносить зміни, було зареєстроване як один з обов'язкових атрибутів вимоги.

На *другому кроці* визначають вплив зміни вимоги на хід розроблення ПЗ, потенційно спричинений запропо-

нованою зміною, який залежатиме від етапу, на якому в певний момент знаходиться програмний проект. Для оцінювання такого впливу необхідна інформація про те, як саме і наскільки вимога, зміни до якої запропоновано, стосується спрямованих "вниз" інформаційних потоків. Тобто, потрібно встановити, як ця вимога пов'язана з іншими вимогами нижчих рівнів, деякими (не)функціональними вимогами, а також, можливо, з розробленою частиною ПЗ, або її функціональністю, що знаходиться вже в експлуатації.

Також тут потрібно фінансистам розрахувати трудовитрати та вартість внесення змін, потрібні для цього додаткові ресурси як матеріальні, так і людські, а також терміни реалізації змін. Ці додаткові терміни потрібно узгодити з запланованими термінами реалізації програмного проекту і, за потреби, відкоректувати їх, а також передбачити санкції інвесторів проекту в разі їх затримання.

На *третьому кроці* рішення має вже приймати спеціальна програмна команда із внесення змін. Склад і форма організації роботи такої команди залежать від специфіки діяльності ІТ-компанії (див. розд. 2), розмаху та величини програмного проекту, етапу конструювання ПЗ чи стану його тестування або етапу його промислової експлуатації. Якщо зміни до вимоги не прийнято, то ці зміни відхиляють, вимога залишається в попередній інтерпретації та статусі, після чого ініціатору зміни надають детальне пояснення причини її неприйняття.

Якщо ж зміни у вимогу узгоджені та прийняті, то процес управління вимогами до ПЗ переходить на *четвертий крок*. У деяких випадках потрібно, щоб зміна була внесена й реалізована негайно, незалежно від вартості виконуваних робіт і їх тривалості. В інших випадках зміну можна відкласти до реалізації наступної версії ПЗ. Безперечно, процедура внесення змін у вимоги може містити будь-яку кількість проміжних етапів, наявність яких диктує специфіка технологічного процесу розроблення ПЗ і тими обставинами, що склалися на цей момент під час реалізації програмного проекту.



Рис. 5. Діаграма переходів стану змін у вимогу до ПЗ

Процес управління вимогами до ПЗ можна подати у вигляді діаграми переходів стану вимоги (рис. 5). Також важливо з самого початку визначити, чи змінюватиметься статус вимоги, стосовно якої потрібно вносити зміни, щоб відобразити цей факт у відповідній документації. Існує два різних підходи до вирішення цього питання. Одна група фахівців вважає, що взаємозв'язок між зміною статусу й самою вимогою так чи інакше зберігається в межах пропозиції на внесення зміни і, як наслідок, немає потреби змінювати статус самої вимоги. Інша група вважає, якщо керівник проекту приймає рішення внести зміну, то конкретна вимога якраз і стає тим об'єктом, до якого потрібно вносити зміни, і, як наслідок, значення статусу цієї вимоги також має змі-

нитися. Зроблено це для того, щоб за статусом вимоги можна було в певний момент визначити статус запропонованої зміни.

Проте, головна проблема процесу управління змінами вимог до ПЗ для компанії-замовника виникає в ситуації, коли декілька модифікацій одного програмного продукту мають загальні для них вимоги, а запропоновані зміни стосуються однієї або декількох цих вимог. У цьому випадку поява запиту на зміну вимог змушує керівника проекту шукати відповіді на такі запитання:

- чи має внесена зміна стосуватися всіх модифікацій програмного продукту?;
- коли ця зміна має бути внесена в кожну з модифікацій програмного продукту?

Неважко вгадати найпоширенішу відповідь: зміну вимоги потрібно внести в усі модифікації програмного продукту, але не одночасно! Така ситуація вимагає введення в діаграму переходів (див. рис. 5) декілька додаткових станів, позаяк зміна вимоги має бути "адапована" за відношенням до кожної модифікації проекту для того, щоб первинний запит на зміну вимоги вважався відпрацьованим до кінця (рис. 6). Оскільки попередні та наступні модифікації програмного продукту мають свої власні базові та змінні вимоги, то не факт, що запропонована зміна до вимоги буде повністю чи частково стосуватися усіх модифікацій продукту. Тому процедуру внесення змін до вимоги спочатку потрібно повністю відпрацювати, наприклад, на поточній модифікації програмного продукту, а вже потім необхідно адаптувати цю процедуру на всіх інших модифікаціях продукту інколи навіть без значних змін.

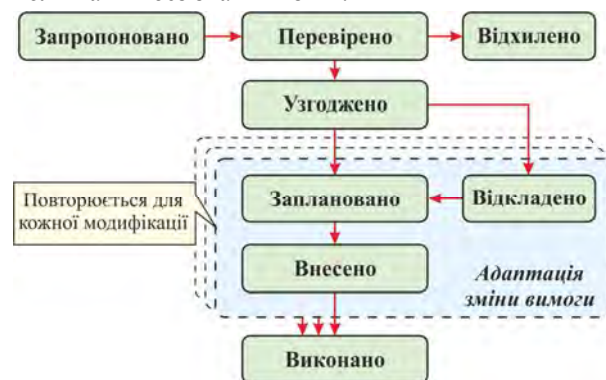


Рис. 6. Діаграма переходів стану зміни вимоги до кожної модифікації програмного продукту

З рис. 6 видно, що для кожної модифікації внесені зміни до вимоги мають мати свої власні значення станів "Заплановано", "Перевірено", "Відкладено", "Внесено". При цьому значення стану "Виконано" первинна зміна може прийняти тільки після відпрацювання його зі всіма модифікаціями програмного продукту. Однак, який би підхід аналітик не використав у своїй повсякденній діяльності, потрібно заздалегідь визначити набір значень, які потрібно застосовувати для ідентифікації статусу змін, і попередньо вирішити, чи будуть ці значення якось впливати на статус рецензування залученої вимоги (чи будуть вони змінювати її статус).

Детальний аналіз особливостей управління вимогами до ПЗ в процесі діяльності компанії його виконавця та виробника, що згодом дасть змогу підготувати відповідні плани виконання завдань програмного проекту та ефективно управляти процесом його реалізації, наведено в роботі (Hrytsiuk, 2018).

Отже, робота з визначення вимог у компанії-замовника зосереджена, переважно, в області наявних проблем, де бізнес-аналітик визначає користувацькі вимоги до ПЗ. Це творчий процес, який передбачає два кроки вперед і три кроки назад, а тому на самому початку реалізації програмного проекту його керівнику надзвичайно важко точно визначити потрібні йому ресурсні можливості та часові обмеження. Однак, після визначення всіх зацікавлених сторін, розроблення та узгодження всіх користувацьких сценаріїв використання, планування послідовності виконання робіт всі подальші наміри можна проводити більш акуратно і прискіпливо до кожного завдання проекту.

На початкових етапах розроблення користувацьких вимог до ПЗ процедура внесення змін має незначний ступінь формалізації, яка постійно збільшується в міру просування етапів реалізації програмного проекту.

Висновки

Встановлено технологічні особливості управління процесами розроблення вимог до ПЗ та внесення змін до них, методи і засоби яких дають змогу проконтролювати стан їх виконання та встановити статус кожної з вимог протягом усіх етапів реалізації програмного проекту. За результатами дослідження можна зробити такі основні висновки.

1. Проаналізовано основні поняття про план виконання завдань програмного проекту та управління процесом його реалізації, що допомогло зрозуміти суть пропонуваніх методів і засобів здійснення контролю за станом виконання кожного із завдань та відстеження їхнього статусу на усіх етапах реалізації програмного проекту.

2. Виявлено проблеми, які виникають під час управління процесом розроблення вимог до ПЗ, що дало змогу запобігти непередбачені втрати і неприємні інциденти, які можуть виникнути під час деяких етапів реалізації програмного проекту.

3. Встановлено деякі особливості управління вимогами до ПЗ в процесі діяльності компанії його замовника, виконавця та виробника, що згодом дасть змогу підготувати відповідні плани виконання завдань програмного проекту та ефективно управляти процесом його реалізації.

4. Зроблено відповідні висновки та надано рекомендації щодо практичного використання запропонуваніх методів і засобів управління процесом розроблення вимог до ПЗ.

Перелік використаних джерел

- Berenbach, B., Paulish, D., Katzmeier, J., & Rudorfer, A. (2009). *Software & Systems Requirements Engineering: In Practice*. New York: McGraw-Hill Professional.
- Bevz, O. M., Papinov, V. M., & Skadan, Yu. A. (2010). *Designing software control systems: Teach. manual Part 1: Fundamentals of Object-Oriented Design*. Vinnytsia: VNTU, 125 p. Retrieved from: <http://posibnyky.vntu.edu.ua/bevz/zm.html>. [In Ukrainian].
- Bronshtein, I. N., & Semendiaev, K. A. (1986). *Handbook of mathematics for engineers and students of universities*, (11th ed.). Moscow: Publishing "Science", Main edition of physics and mathematical literature, 544 p. [In Russian].
- Cockburn, A. (2001). *Writing Effective Use Cases*. Addison-Wesley, 270 p.
- Dick, J., Hull, E., & Jackson, K. (2017). *Requirements Engineering*. (4rd ed.) Springer. <https://doi.org/10.1007/978-3-319-61073-3>
- Futrell, R. T., Shafer, D. F., & Shafer, L. I. (2002). *Quality Software Project Management*. Prentice Hall, 1639 p.
- Hovorushchenko, T. O. (2018). *Teoretychni ta prykladni zasady informatsiinoi tekhnologii otsiniuvannia dostatnosti informatsii shchodo yakosti u spetsyfikatsiakh vymoh do prohramnoho zabezpechennia. Abstract of Doctoral Dissertation for Technical Sciences* (05.13.06 – Information technologies). Lviv: Ukrainska akademiia drukarstva, 43 p. [In Ukrainian].
- Hrytsiuk, Yu. I. (2018). *Analysis of Software Requirements: Tutorial*. Lviv: Publishing House of Lviv Polytechnic, 460 p. [In Ukrainian].
- Hrytsiuk, Yu. I., & Leshkevych, I. F. (2017). The Problems of Definition and Analysis of Software Requirements. *Scientific Bulletin of UNFU*, 27(4), 148–158. <https://doi.org/10.15421/40270433>
- Hull, E., Jackson, K., & Dick, D. (2005). *Development and management of requirements. Practical user manual*. (2nd ed.), 230 p. Retrieved from: http://www.swd.ru/files/share/DOORS/books/eBook_RU_Requirements_Engineering.pdf. [In Russian].
- Jones, C., & Bonsignour, O. (2012). *The economics of software quality*. Boston: Pearson Education, 588 p.
- Kharchenko, A., Galay, I., & Yatsyshyn, V. (2011). The method of quality management software. *VII International Conference on Perspective Technologies and Methods in MEMS Design: Proceedings*, (pp. 82–84), May 11–14. Polyana (Ukraine).
- Kharchenko, O., & Yatsyshyn, V. (2009). Rozrobka ta keruvannia vymohamy do prohramnoho zabezpechennia z vy-korystanniam modeli yakosti PZ. *Visnyk Ternopilskoho derzhavnogo tekhnichnoho universytetu*, 14(1), 201–207. [In Ukrainian].
- Kharchenko, V. S., Netkacheva, E. I., Orekhova, A. A., et al. (2012). *CASE-otcenka kriticheskikh programnykh sistem: monografiia*, (In 3 vol.), Vol. 1: Bezopasnost, (Kharchenko, V. S. Scientific Ed.). Kharkov: NAU "KhAI", 301 p. [In Russian].
- Konorev, B. M. (Ed.), Manzhos, Iu. S., Kharchenko, V. S. (Ed.) et al. (2009). *Invariantno-orientirovannaia otenka kachestva programnogo obespechennia kosmicheskikh sistem: monografiia*. Kharkov: NAU "KhAI", 224 p. [In Russian].
- Kornipaev, Iliia. (2013). *Trebovaniia dlia programmnoho obespechennia. Rekomendatsii po sboru i dokumentirovaniu*. Moscow: OZON Status, 118 p. [In Russian].
- Kroll, P., & Krachten, F. (2004). *Rational Unified Process – eto legko. Rukovodstvo po RUP dlia praktikov: per. s angl.* Moscow: KUDITC-OBRAZ. 432 p. [In Russian].
- Laplante, Phil. (2009). *Requirements Engineering for Software and Systems* (1st ed.). Redmond, WA: CRC Press, 268 p.
- Lavrishcheva, E. M. (2013). *Software Engineering kompiuternykh sistem. Paradigmy, tekhnologii i CASE-sredstva programmirovaniia*. Kyiv: Naukova dumka, 283 p. [In Russian].
- Leffingwell, D., & Widrig, D. (1999). *Managing Software Requirements: A Unified Approach*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA
- Maedche, A., Botzenhardt, A., & Neer, L. (2012). *Software for People: Fundamentals, Trends and Best Practices (Management for Professionals)*. Springer-Verlag Berlin Heidelberg, 293 p.
- Maievskiy, D., & Kozina, Iu. (2015). Gde i kogda formiruetsia kachestvo programmnoho obespechennia? *Elektrotekhnicheskies i kompiuternye sistemy*, 18, 55–59. [In Russian].
- Mansilla, D., Pollo-Cattaneo, M., Britos, P., & Garcia-Martinez, R. (2013). A Proposal of a Process Model for Requirements Elicitation in Information Mining Projects. *Lecture Notes in Business Information Processing*, 4, 165–173. https://doi.org/10.1007/978-3-642-36611-6_13
- Matciashchek, L. A. (2002). *Analiz trebovaniy i proektirovaniia sistem. Razrabotka informatsionnykh sistem s ispolzovaniem UML*. Moscow: Izd. dom "Viliams", 432 p. [In Russian].
- McConnell, S. (2013). *Sovershennyi kod. Master-klass*. Moscow: Izdvo "Russkaia redaktsiia". 896 p. [In Russian].
- Mishchenko, V. O., Pomorova, O. V., & Hovorushchenko, T. A. (2012). *CASE-otcenka kriticheskikh programnykh sistem: monografiia*, (In 3 vol.), Vol. 1: Kachestvo. (Kharchenko, V. S. Scientific Ed.). Kharkov: Natc. aerokosmicheskii universitet "KhAI". 201 p. [In Russian].

- Myers, G., Badzhett, T., & Sandler, K. (2012). *Iskusstvo testirovaniia programm*, (3rd ed.). Moscow: OOO "ID Viliams", 272 p. [In Russian].
- Pomorova, O. V., & Hovorushchenko, T. O. (2013). Intelligent Assessment and Prediction of Software Characteristics at the Design Stage. *American Journal of Software Engineering and Applications (AJSEA)*, 2(2), 25–31. Retrieved from: <http://article.sciencepublishinggroup.com/pdf/10.11648.j.ajsea.20130202.11.pdf>.
- Project. (2008). A Guide to the Project Management Body of Knowledge (4th ed.). Project Management Institute. Retrieved from: <https://marketplace.pmi.org/Pages/ProductDetail.aspx?GMProduct=00101388701>
- Sommerville, I. (2002). *Inzheneriia programmnogo obespecheniia*, (6ix ed.). Moscow: Izd. dom "Viliams", 624 p. [In Russian].
- Stellman, Andrew; Greene, Jennifer. (2005). *Applied Software Project Management*. O'Reilly Media. Retrieved from: <http://www.stellman-greene.com/about/applied-software-project-management/>
- Sutcliffe, A. (1998). Scenario-based requirements analysis. *Requirements Engineering*, 3(1), 48–65. <https://doi.org/10.1007/bf02802920>
- Wiegiers, K., & Betti, D. (2014). Development of Software Requirements. (Trans. from English). (3rd ed.). Moscow: Russian Edition; St. Petersburg: BHV-Petersburg, 736 p. [In Russian].
- Wiegiers, Karl E. (2003). *Software Requirements (2nd ed.) (Developer Best Practices)*. Redmond, WA: Microsoft Press, 544 p.
- Yakovyna, V. S. (2012). Vplyv funktsii aktyvatsii RBF neironnoi me-rezhii na efektyvnist prohnozuvannya kilkosti vidmov programmoho zabezpechennia. *Visnyk Natsionalnoho universytetu "Lvivska politekhnika". Seriia: "Kompiuterni nauky ta informatsiini tekhnologii"*, 732, 36–39. [In Ukrainian].

Ю. И. Грыцюк, Е. А. Немова

Национальный университет "Львовская политехника", г. Львов, Украина

ОСОБЕННОСТИ УПРАВЛЕНИЯ ПРОЦЕССОМ РАЗРАБОТКИ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

Приведены технологические особенности управления процессами разработки требований к ПО и внесения изменений в них, методы и средства которых позволяют проконтролировать состояние их выполнения и установить статус каждого из требований в течение всех этапов реализации программного проекта. Проанализированы особенности планирования выполнения задач программного проекта и управления процессом его реализации, которые помогли понять суть предлагаемых методов и средств осуществления контроля за выполнением каждой из задач и отслеживания ее статуса на всех этапах реализации программного проекта. Выявлены проблемы, возникающие при управлении процессом разработки требований к ПО, что позволяет предотвратить непредвиденные потери и неприятные инциденты, которые могут возникнуть на некоторых этапах реализации программного проекта. Выяснено, что большинство выявленных проблем связанные прежде всего с: планированием последовательности выполнения работ; контролем за состоянием выполнения работ; управлением изменениями требований к ПО; отслеживанием статуса каждого требования. Характер проблем, с которыми сталкиваются IT-компании при управлении требованиями к ПО, зависит от специфики их деятельности. Исследованы некоторые особенности управления требованиями к ПО в процессе деятельности компании заказчика, исполнителя и производителя, что позволяет своевременно подготовить соответствующие планы выполнения задач программного проекта и эффективно управлять процессом их реализации. Установлено, что на начальных этапах разработки пользовательских требований к ПО процедура внесения изменений имеет незначительную степень формализации, которая должна постоянно увеличиваться по мере продвижения этапов реализации программного проекта.

Ключевые слова: программный проект; требования к программному обеспечению; определение требований; разработка требований; управление требованиями; рецензирование требований; спецификация требований; управление процессом разработки требований к программному обеспечению.

Yu. I. Hrytsiuk, E. A. Nemova

Lviv Polytechnic National University, Lviv, Ukraine

MANAGEMENT FEATURES PROCESS OF DEVELOPING SOFTWARE REQUIREMENTS

The technological features of management of the development process of software requirements and manage the process of making changes to the requirements. Such methods and management tools make it possible to monitor the state of compliance with the requirements and set the status of each requirement during all phases of the software project. The features task scheduling software project management and their implementation process. These features help to understand the essence of the proposed methods and tools for monitoring the implementation of each task and monitor its status at all stages of a software project. The problems that arise in managing the process of developing software requirements. They help to prevent unforeseen losses and unpleasant incidents that may occur in certain phases of the software project. The nature of the problems identified in the first place depends on the specifics of the IT-companies. It was found that most of the problems identified are related to: planning of the sequence of works; control of the execution status of works; Process change management requirements; tracking the status of each requirement.

We investigated some features to the software requirements management in the process of the client company, performer and producer. This allows time to prepare appropriate plans for the tasks of a software project and effectively manage the process of their implementation. It was found that the company-customer work to identify the requirements are concentrated in the area of the existing problems, where the business analyst defines user requirements for the software. It is a creative process that involves two steps forward and three steps back. Therefore, at the beginning of the implementation of a software project of its head it is very difficult to accurately determine the required resource capacity and time constraints. However, after the planning of work sequence, all further intentions can be carried out more accurately and thoroughly on each project task. It was established that in the initial stages of the development of user requirements amendment procedure has a slight degree of formalization. It should continue to increase as we move phases of a software project.

Keywords: software project; software requirements; definition of requirements; development of requirements; requirements management; review requirements; requirements specification; requirements development process management software.