

УДК 004.891.3: 004.3

Р. А. МАЛЯРЧУК, Т. О. ГОВОРУЩЕНКО

Хмельницький національний університет, Україна

МНОЖИННІ ПРЕДСТАВЛЕННЯ ТИПІВ ТА МЕТОДОЛОГІЙ РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Дана робота присвячена побудові множинних представлень типів програмного забезпечення (ПЗ) з врахуванням їх основних сутнісних та показникових характеристик та методологій розроблення ПЗ з врахуванням їх основних принципів. Розроблені множинні представлення надають замовнику та розробнику можливість підбору необхідної методології для поставленої задачі шляхом порівняння множини необхідних характеристик ПЗ розроблюваного типу з множинами принципів, які забезпечуються різними методологіями. Множинні представлення типів та методологій розроблення ПЗ використовуватимуться надалі при моделюванні та реалізації процесу вибору методології розроблення ПЗ, для чого слід встановити важливість (вагові коефіцієнти) характеристик для кожного типу ПЗ та принципів для кожної методології розроблення ПЗ.

Ключові слова: програмне забезпечення (ПЗ), програмний проект, типи програмного забезпечення, методологія розроблення ПЗ, характеристики ПЗ, принципи методологій розроблення ПЗ.

Вступ

Наразі для створення повнофункціональних програмних додатків програмістам доводиться використовувати комплексні засоби розроблення, зокрема, методологію розроблення програмного забезпечення (ПЗ). Методологія розроблення ПЗ – це набір методів та критеріїв оцінки, які використовуються для постановки задачі, планування, контролю і для досягнення поставленої мети [1].

Сьогодні найбільш відомими та використовуваними є наступні методології: «швидкі» Agile (SCRUM, XP, DSDM, Kanban та інші), прогнозуючі та ітеративні Rational Unified Process (RUP), каскадні (Waterfall), а також Microsoft Solutions Framework (MSF), яка поєднує принципи каскадної та спіральної моделей [2].

Отже, існує велика кількість методологій, але відсутні чіткі стандартизовані критерії вибору методології для розроблення ПЗ певного типу (лише у роботі [3] введено декілька метрик, які дозволяють частково формалізувати процес підбору методології), в той час як від вибору методології розроблення ПЗ залежить успішність реалізації програмного проекту різних розмірів – табл. 1 [4].

При виборі методології розроблення ПЗ зараз не враховується тип та цільове призначення програмного проекту (автор [3] підкреслює важливість вибору не «найкращої» методології, а тієї, яка найкращим чином відповідає проектній задачі та колективу розробників). Але різні типи програмних проектів вимагають різних підходів, оскільки кожна категорія проектів має різні пріоритети і цілі, тому

важливою і актуальною є задача вибору методології розроблення для програмного забезпечення конкретного типу.

Таблиця 1

Порівняння успішності програмних проектів, реалізованих за каскадними (Waterfall) та AGILE- методологіями

Size	Method	Successful	Challenged	Failed
All	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

Тому метою даного дослідження є визначення основних характеристик різних типів ПЗ та основних принципів сучасних методологій розроблення ПЗ, на основі яких і відбуватиметься підбір необхідної методології для поставленої задачі шляхом порівняння множини необхідних характеристик ПЗ розроблюваного типу з множинами принципів, які забезпечуються різними методологіями розроблення.

1. Множинні представлення типів програмного забезпечення

Оскільки вибір методології розроблення слід виконувати з врахуванням основних характеристик ПЗ певного типу, то визначимо основні характерис-

тики різних типів ПЗ та подамо їх у вигляді множин для ПЗ кожного типу.

Прикладне ПЗ поділяється, як відомо, на ПЗ загального та спеціального призначення. Нашу увагу буде зосереджено на ПЗ спеціального призначення, яке використовується у специфічній діяльності користувачів. Надалі ми розглядатимемо наступні види ПЗ спеціального призначення: Web-додатки, мобільні додатки, програми електронного навчання, прикладні програми (для статистики та бухгалтерії), автоматизовані системи та інформаційні системи.

Представимо кожен розглядуваний тип ПЗ у формалізованому вигляді – у вигляді кортежу:

$$T = \langle EC_T, IC_T \rangle, \quad (1)$$

де EC_T - множина сутнісних характеристик, IC_T - множина показникових характеристик, які необхідно забезпечити при реалізації ПЗ конкретного типу (обираються зі стандартів та фахової літератури).

Тоді множина сутнісних характеристик для Web-додатків має вигляд (на основі [5-7]):

$$EC_{Web} = \{As, Csa, Csmrt, Cc, Cai, Par\}, \quad (2)$$

де As - можливість налагодження (параметрів броузера), Csa - наявність клієнт-серверної архітектури, $Csmrt$ - можливість керування контентом в режимі реального часу, Cc - можливість кешування контенту, Cai - можливість колективного доступу, Par - можливість внесення змін до вимог протягом всього життєвого циклу проекту.

Множина показникових характеристик для Web-додатків має вигляд (на основі [6-8]):

$$IC_{Web} = \left\{ \begin{array}{l} Cp, Pr, Ub, Ia, Ss, Sa, Is, Rt, Bw, At, \\ Om, Qu, Rb, Sb, Sui \end{array} \right\} \quad (3)$$

де Cp - висока кросплатформність, Pr - висока продуктивність, Ub - висока зручність використання, Ia - висока інтерактивність, Ss - висока безпека та захищеність, Sa - відносно невеликий розмір додатку, Is - висока швидкість взаємодії, Rt - малий час відгуку, Bw - висока пропускна здатність, At - відносно невеликий трафік додатку, Om - відносно невеликий обсяг оперативної пам'яті, Qu - велика кількість одночасно працюючих користувачів, Rb - висока надійність (доступ до всіх функцій за будь-яких умов, тобто всі умови повинні бути враховані при реалізації), Sb - висока масштабованість (здатність збільшувати продуктивність за потреби), Sui - висока стандартизованість інтерфейсу користувача,

Мобільні додатки представимо у вигляді наступної множини необхідних сутнісних характеристик (на основі [9-11]):

$$EC_{Mob} = \{Uuis, Par\}, \quad (4)$$

де $Uuis$ - одноманітність стилю інтерфейсу користувача, Par - можливість внесення змін до вимог протягом всього життєвого циклу проекту.

Множина показникових характеристик для мобільних додатків має вигляд (на основі [10-12]):

$$IC_{Mob} = \left\{ \begin{array}{l} Ts, Rt, Ds, Ieds, Om, Ub, Pr, Emm, \\ Rb, Dus, Qd, Qsc, Cts, Ini, Ss \end{array} \right\}, \quad (5)$$

де Ts - малий час запуску додатка, Rt - малий час відгуку додатку, Ds - високий ступінь спеціалізації додатку (фокусування уваги на конкретних задачах, цільова орієнтація), $Ieds$ - високий рівень взаємодії із зовнішніми джерелами інформації та зовнішнім ПЗ, Om - відносно невеликий обсяг оперативної пам'яті, Ub - висока зручність використання, Pr - висока продуктивність, Emm - високоефективне керування пам'яттю, Rb - висока надійність, Dus - високий ступінь задоволення запитів та потреб користувача, Qd - відносно мала кількість рішень, Qsc - велика кількість контенту на екрані, Cts - низька складність виконання задач, Ini - висока інтуїтивність інтерфейсу, Ss - висока безпека та захищеність.

Програми електронного навчання можна представити у вигляді наступної множини сутнісних характеристик (на основі [13-15]):

$$EC_{E-learn} = \left\{ \begin{array}{l} Fbu, Ucs, Nfpoi, Mcs, Ru, \\ Upp, Or, Fsd, Par \end{array} \right\}, \quad (6)$$

де Fbu - зворотній зв'язок з користувачем, Ucs - використання єдиних стандартів, $Nfpoi$ - наявність нових форм представлення та організації інформації, Mcs - модульність структури курсів, Ru - багаторазове використання (здатність гнучко використовувати навчальні об'єкти у різному контексті та в багатьох навчальних курсах), Upp - персоналізація сторінки користувача, Or - організація звітності, Fsd - можливість формування статистичних даних, Par - можливість внесення змін до вимог протягом всього життєвого циклу проекту.

Множина показникових характеристик для програм електронного навчання має вигляд (на основі [13-15]):

$$IC_{E-learn} = \{Ub, Qua, Ae, Ack, Sst, Io, Fio, Da, Iu\}, \quad (7)$$

де U_b - висока зручність використання, Q_{ua} - висока якість, A_e - висока доступність навчання (можливість знаходити та одержувати доступ до навчальних об'єктів з різних джерел, доступність для людей з особливими потребами), A_{sk} - висока достовірність сертифікації знань, S_{st} - висока підтримка студентів, I_o - висока сумісність (можливість використовувати навчальні об'єкти, розташовані в різних місцях, створені різними інструментами та на різних платформах), F_{io} - висока функційна сумісність (збереження роботоздатності при використанні різного програмного та апаратного забезпечення), D_a - велика довговічність (здатність витримувати будь-які технологічні зміни без переробки або зміни внутрішньої структури навчального об'єкту), I_u - висока взаємодія між користувачами.

Прикладні програми для статистики та бухгалтерії представимо у вигляді наступної множини необхідних сутнісних характеристик (на основі [16, 17]):

$$EC_{Appl} = \{As, Mt, Ud, Ass, Odb, Fsd, Or, Pad, Par\}, \quad (8)$$

де As - можливість налагодження (на конкретні умови використання), Mt - наявність засобів керування (які дозволяють обирати конкретні можливості з передбачених), Ud - наявність і достатність документації для користувача, Ass - можливість післяпродажного обслуговування, Odb - централізація даних в єдиній базі, Fsd - можливість формування статистичних даних, Or - організація звітності, Pad - можливість статистичного опрацювання та аналізу даних, Par - можливість внесення змін до вимог протягом всього життєвого циклу проекту.

Множина показникових характеристик для таких програм має вигляд (на основі [16, 17]):

$$IC_{Appl} = \{U_b, Eem, Pa, Vl, Ssf, Uo, Mp, Io, Udf, Ss\}, \quad (9)$$

де U_b - висока зручність використання, Eem - висока зручність розширення та модифікації, Pa - висока пакетність (повне покриття потреб предметної галузі), Vl - висока універсальність (дозволяє розв'язувати більшість задач певної предметної галузі), Ssf - висока відповідність існуючим стандартам, Uo - висока орієнтація на користувача, Mp - висока модульність реалізації програми, I_o - висока сумісність (з іншим ПЗ фірми-розробника), Udf - висока уніфікація форматів представлення різних даних, Ss - висока безпека та захищеність.

Автоматизовані системи можна представити у вигляді наступної множини необхідних сутнісних характеристик (на основі [18-20]):

$$EC_{AS} = \{Ud, As, Par\}, \quad (10)$$

де Ud - наявність та достатність документації для користувача, As - можливість налагодження системи, Par - можливість внесення змін до вимог протягом всього життєвого циклу проекту.

Множина показникових характеристик для автоматизованих систем має вигляд (на основі [18-20]):

$$IC_{AS} = \left\{ \begin{array}{l} Em, Rd, Scpd, Qd, Lc, Rm, Dvd, \\ Csp, Ssf, Io, Fc, Rb, Ub, Eem \end{array} \right\}, \quad (11)$$

де Em - висока ефективність керування об'єктом та виробництвом, Rd - висока релевантність даних для прийняття рішень, $Scpd$ - висока швидкість збирання та опрацювання даних, Qd - відносно мала кількість рішень, Lc - високий рівень контролю, Rm - висока оперативність керування процесом, Dvd - високий ступінь обґрунтованості прийнятих рішень, Csp - низькі витрати на допоміжні процеси, Ssf - висока відповідність існуючим стандартам, I_o - висока сумісність (організаційна, технічна, програмна, математична, лінгвістична), Fc - високий рівень функційної повноти, Rb - висока надійність, Ub - висока зручність використання, Eem - висока зручність розширення та модифікації.

Інформаційні системи представимо у вигляді наступної множини необхідних сутнісних характеристик (на основі [21, 22]):

$$EC_{IS} = \left\{ \begin{array}{l} Shst, Fsd, Or, Pad, Cai, \\ Ati, Par, Odb, Rtw, Cmm \end{array} \right\}, \quad (12)$$

де $Shst$ - можливість розв'язання лише добре структурованих задач (для яких можна розробити алгоритм), Fsd - можливість формування статистичних даних, Or - організація звітності, Pad - можливість статистичного опрацювання та аналізу даних, Cai - можливість колективного доступу, Ati - відсутність технічних інтерфейсів між підсистемами (необхідність використання для цього людей), Par - можливість внесення змін до вимог протягом всього життєвого циклу проекту, Odb - централізація даних в єдиній базі, Rtw - близький до реального часу режим роботи, Cmm - спільна модель керування для різних галузей.

Множина показникових характеристик для інформаційних систем (на основі [21, 22]):

$$IC_{IS} = \left\{ Scpd, Rds, Larw, Ub, Rirf, Lpp, Us, Dm, \right. \\ \left. Ld, Dus, Dich, Efir, Eem, Rb, Ss \right\}, \quad (13)$$

де Scpd - висока швидкість збирання та опрацювання даних (швидкодія), Rds - висока надійність зберігання даних, Larw - високий рівень автоматизації рутинної роботи, Ub - висока зручність використання, Rirf - високий рівень готовності системи до впливу випадкових факторів (відмови, помилки, збої, зловмисні дії), Lpp - високий рівень участі людини в інформаційних процесах, Us - висока унікальність станів інформаційної системи, Dm - висока динамічність (зміна структури та станів елементів в часі), Ld - високий рівень розподіленості (просторове розташування компонентів системи), Dus - високий ступінь задоволення запитів та потреб користувача, Dich - високий рівень повноти, корисності, цінності, точності, достовірності та актуальності інформації, Efir - висока ефективність опрацювання інформації, Eem - висока зручність розширення та модифікації, Rb - висока надійність, Ss - високий рівень безпеки та захищеності.

Повнота побудованих множинних представлень різних типів ПЗ забезпечується використанням для їх побудови діючих стандартів, які регламентують розроблення ПЗ різних типів. Побудовані множинні представлення будуть використовуватись надалі при моделюванні та реалізації процесу вибору методології розроблення для ПЗ певного типу.

2. Множинні представлення методологій розроблення програмного забезпечення

Методологія розроблення ПЗ – це система принципів, а також сукупність ідей, понять, характеристик, методів, способів та засобів, які визначають стиль розроблення ПЗ [1]. Представимо кожну розглядану методологію розроблення ПЗ у формалізованому вигляді – у вигляді множинних представлень, елементами яких є основні принципи, ідеї, поняття, характеристики методології (обирались зі стандартів та фахової літератури).

Представимо методологію SCRUM у вигляді наступної множини її ключових принципів (на основі [23-25]):

$$M_{SCRUM} = \{Scd, Ps, Ncfi, Cti\}, \quad (14)$$

де Scd - невеликі команди розробників, Ps - розбиття на ітерації (спринти) тривалістю 30 днів, Ncfi - незмінність функцій під час однієї ітерації,

Cti - суворе дотримання термінів випуску чергового релізу (навіть якщо не вдається реалізувати весь планований функціонал).

Методологію KANBAN можна представити у вигляді наступної множини її ключових принципів (на основі [23, 26, 27]):

$$M_{KANBAN} = \{To, Dv, Lcw, Mct, Po\}, \quad (15)$$

де To - орієнтація на задачі, Dv - візуалізація розроблення, Lcw - обмеження одночасних робіт на кожному етапі розроблення, Mct - вимірювання часу циклу (середній час на одну задачу), Po - оптимізація процесу.

Представимо методологію DSDM у вигляді наступної множини її основних принципів (на основі [23, 28, 29]):

$$M_{DSDM} = \{Sfr, Prc, Aiu, Frv, Ad, Tac, Si, Qua, Acr\}, \quad (16)$$

де Sfr - вивчення можливості реалізації ПЗ та галузі її застосування, Prc - розбиття процесу на три взаємозв'язаних цикли (цикл функційної моделі, цикл проектування та конструювання, цикл реалізації), Aiu - активна взаємодія з користувачами, Frv - часті випуски версій (релізів), Ad - самостійність розробників в прийнятті рішень, Tac - тестування протягом всього циклу робіт, Si - короткі ітерації (2-6 тижнів кожна), Qua - висока якість, Acr - адаптованість до змін у вимогах.

Методологію MSF можна представити у вигляді наступної множини її основних принципів (на основі [23, 30, 31]):

$$M_{MSF} = \{Orv, Cm, Par, Vrc, Pfc, Cbv, Psf, Sb, Lsi\}, \quad (17)$$

де Orv - єдине бачення проекту, Cm - керування компромісами, Par - гнучкість (готовність до змін), Vrc - концентрація на бізнес-пріоритетах, Pfc - заохочення вільного спілкування всередині проекту, Cbv - створення базових версій, Psf - розбиття стадії на 5 фаз (концепція, планування, розроблення, стабілізація, впровадження), Sb - масштабованість, Lsi - відсутність жорстких інструкцій.

Представимо методологію RUP у вигляді наступної множини її ключових принципів (на основі [23, 32, 33]):

$$M_{RUP} = \{Cmpa, Ut, Ier, Frr, Par, Qua, Vfd\}, \quad (18)$$

де Cmpa - компонентна архітектура, Ut - робота над проектом у згуртованій команді (ключова роль в архітекторів), Ier - рання ідентифікація та неперервне усунення можливих ризиків, Frr -

концентрація на виконанні вимог замовників, Раg - очікування змін у вимогах в процесі розроблення (гнучкість), Qua - постійне забезпечення якості на всіх етапах, Vfd - мінливість ступеня формалізації (залежно від потреб проекту), за рахунок чого RUP використовується як для невеликих та коротких, так і для великих та складних проектів.

Методологію XP можна представити у вигляді наступної множини її ключових принципів (на основі [23, 34, 35]):

$$M_{XP} = \left\{ \text{Lfd, Scpt, Cmsh, Pp, Fsf, Tdd, Cig, Frv, Smpd, Cco, Ocst, Fmpp, Si, Ms} \right\}, \quad (19)$$

де Lfd - низький ступінь формалізації, Scpt - малі, спільно розташовані команди розробників (4-10 чоловік), Cmsh - постійний зв'язок із зацікавленими сторонами проекту, Pp - «парне» програмування, Fsf - короткий цикл зворотнього зв'язку, Tdd - розроблення через тестування (автоматичне), Cig - неперервний процес, неперервна інтеграція, Frv - часті випуски версій, Smpd - просте та зрозуміле розроблення, Cco - колективне володіння кодом, Ocst - єдиний стандарт кодування, Fmpp - швидке формування плану роботи та його оновлення (по мірі уточнення цілей проекту), Si - ітеративність (короткі ітерації), Ms - метафора системи.

Представимо каскадну (waterfall) методологію у вигляді наступної множини її основних принципів (на основі [23, 36, 37]):

$$M_{\text{Waterfall}} = \{ \text{Npar, Rss, Hfd, Ub, Tdp, Rr, Cdp, Sm, Ee} \}, \quad (20)$$

де Npar - відсутня можливість внесення змін до вимог протягом життєвого циклу проекту (низька гнучкість), Rss - жорстка послідовність проходження етапів, Hfd - високий ступінь формалізації, Ub - висока зручність для замовника, Tdp - прозорість процесу розроблення, Rr - низькі ризики, Cdp - ясність процесу розроблення, Sm - простота та зручність керування (навіть масштабними проектами), Ee - можливість ранніх оцінок вартості, термінів та ефективності проекту.

Висновки

Наразі до проблеми вибору методології розроблення ПЗ підходять без врахування специфіки софтверної компанії та програмного проекту. Але саме застосування методології, яка найкращим чином відповідає проектній задачі та колективу розробників, дозволить усунути розбіжності між розробника-

ми та замовниками, зводити до мінімуму витрати на супровід ПЗ.

На основі аналізу стандартів та фахової літератури було визначено основні сутнісні та показникові характеристики типів ПЗ та основні принципи методологій розроблення ПЗ. Запропоновано множинні представлення типів програмного забезпечення (ПЗ) з врахуванням їх основних характеристик та методологій розроблення ПЗ з врахуванням їх ключових принципів. Розроблені множинні представлення надають замовнику та розробнику можливість підбору необхідної методології для поставленої задачі шляхом порівняння множини необхідних характеристик ПЗ розроблюваного типу з множинами принципів, які забезпечуються різними методологіями.

Для подальшого моделювання та реалізації процесу вибору методології розроблення ПЗ (або, швидше, компанії-розробника, яка використовує підходящу методологію) потрібно встановити важливість (вагові коефіцієнти) характеристик для кожного типу ПЗ та принципів для кожної методології розроблення ПЗ, які відображають роль та пріоритетність характеристики і принципу відповідно. Визначення вагових коефіцієнтів дасть можливість оцінити ефективність та зробити обґрунтований вибір методології для розроблення ПЗ певного типу. Саме на вирішення такої задачі і будуть спрямовані подальші зусилля авторів.

Література

1. Хаф, Л. *Методологии разработки программного обеспечения [Электронный ресурс] / Л. Хаф. – Режим доступа: <http://compress.ru/article.aspx?id=11321>. – 15.02.2016.*
2. Говоруценко, Т. О. *Аналіз процесу вибору технології проектування, методології та середовища розроблення програмного забезпечення [Текст] / Т. О. Говоруценко, Р. А. Мальярчук // Вісник Хмельницького національного університету. – 2014. - № 6. – С. 186-196.*
3. Коберн, А. *Быстрая разработка программного обеспечения [Текст] / А. Коберн. – М. : Лори, 2002. – 314 с.*
4. Standish Group 2015 *Chaos Report [Electronic resource]. – Access mode: www.infoq.com/articles/standish-chaos-2015 - 15.02.2016.*
5. ISO 40500:2012. *Information technology. W3C Web Content Accessibility Guidelines (WCAG) 2.0 [Text]. – Introduced 15.10.2012. – ISO/IEC, 2012. – 2 p.*
6. ISO 9241-151:2008. *Ergonomics of human-system interaction. Part 151: Guidance on World Wide Web user interfaces [Text]. – Introduced 15.05.2008. – ISO/IEC, 2008. – 49 p.*
7. ISO/IEC 40240:2011. *Information technology.*

W3C Web Services Addressing 1.0. Core [Text]. – Introduced 01.09.2011. – ISO/IEC, 2011. – 1 p.

8. ISO/IEC 40260:2011. Information technology. W3C Web Services Addressing 1.0. Metadata [Text]. – Introduced 01.09.2011. – ISO/IEC, 2011. – 1 p.

9. Standards for Web Applications on Mobile (2015) [Electronic resource]. – Access mode: <https://www.w3.org/Mobile/mobile-web-app-state/> - 15.02.2016.

10. Mobile Web Application Best Practices (2010) [Electronic resource]. – Access mode: <https://www.w3.org/TR/mwabp/> - 15.02.2016.

11. Mobile Web Best Practices 1.0 (2008) [Electronic resource]. – Access mode: <https://www.w3.org/TR/mobile-bp/> - 15.02.2016.

12. W3C mobileOK Basic Tests 1.0 (2008) [Electronic resource]. – Access mode: <https://www.w3.org/TR/mobileOK-basic10-tests/> - 15.02.2016.

13. Bianco, A. M. Standards for e-learning [Electronic resource] / A. M. Bianco, M. D. Marsico, M. Temperini. – Access mode: http://www2.tisip.no/quis/public_files/wp5-standards-for-elearning.pdf - 15.02.2016.

14. SCORM 2004: 4th edition [Electronic resource] – Access mode: <https://www.adlnet.gov/adl-research/scorm/scorm-2004-4th-edition/> - 15.02.2016.

15. AICC Guidelines & Recommendations (AGR's) [Electronic resource]. – Access mode: www.aicc.org – 15.02.2016.

16. SSARS No. 3, 4, 13, 14, 19, 21. Statements on Standards for Accounting and Review Services [Electronic resource]. – Access mode: <http://www.aicpa.org/RESEARCH/STANDARDS/COMPILATIONREVIEW/Pages/compilation%20and%20review%20standards.aspx> – 15.02.2016.

17. SPSS Statistics Standard [Electronic resource] – Access mode: <http://www-03.ibm.com/software/products/en/spss-stats-standard>. - 15.02.2016.

18. ISO 16100-1:2009. Industrial automation systems and integration. Manufacturing software capability profiling for interoperability. Part 1: Framework [Text]. – Revises ISO 16100-1:2002 : introduced 15.12.2009. – ISO/IEC, 2009. – 45 p.

19. ISO 16100-2:2003. Industrial automation systems and integration. Manufacturing software capability profiling for interoperability. Part 2: Profiling methodology [Text]. – Introduced 01.11.2003. – ISO/IEC, 2003. – 17 p.

20. ISO 16100-6:2011. Industrial automation systems and integration. Manufacturing software capability profiling for interoperability. Part 6: Interface services and protocols for matching profiles based on multiple capability class structures [Text]. – Introduced 15.06.2011. – ISO/IEC, 2011. – 70 p.

21. ITAF: A Professional Practices Framework for IS Audit/Assurance [Electronic resource]. – Access mode: http://www.isaca.org/Knowledge-Center/Research/Documents/ITAF-3rd-Edition_fm_k_Eng_1014.pdf. – 15.02.2016.

22. ISO/IEC 20000-1:2011. Information technology. Service management. Part 1: Service management system requirements [Text]. – Revises ISO 20000-1:2005 : introduced 15.04.2011. – ISO/IEC, 2011. – 26 p.

23. ISO/IEC 90003:2014. Software engineering. Guidelines for the application of ISO 9001:2008 to computer software [Text]. – Revises ISO 90003:2004 : introduced 15.12.2014. – ISO/IEC, 2014. – 54 p.

24. The Scrum Guide [Electronic resource]. – Access mode: <http://www.scrumguides.org/scrum-guide.html>. - 15.02.2016.

25. Schwaber, K. Agile Project Management with SCRUM: Developer Best Practices [Text] / K. Schwaber. – Redmond: Microsoft Press, 2004. – 188 p.

26. Cimorelli, S. KANBAN for the Supply Chain: Fundamental Practices for Manufacturing Management [Text] / S. Cimorelli. – Boca Raton: CRC Press, 2013. – 128 p.

27. The Kanban System [Electronic resource]. – Access mode: <http://isoconsultantpune.com/kanban-system/> - 15.02.2016.

28. DSDM Atern Handbook (2008) [Electronic resource]. – Access mode: <https://www.dsdm.org/content/delivering-quality>. - 15.02.2016.

29. Roberts, B. Dynamic Systems Development Method. The Standard for Rapid Application Development [Text] / B. Roberts. // Software Quality. – 2001. – Part 1. – P. 16-33.

30. Microsoft Solutions Framework Overview [Electronic resource] – Access mode: <https://msdn.microsoft.com/en-us/library/jj161047%28v=vs.120%29.aspx> – 15.02.2016.

31. Villafiorita, A. Introduction to Software Project Management [Text] / A. Villafiorita. – Boca Raton: CRC Press, 2014. – 268 p.

32. Rational Unified Process: Best Practices for Software Development Teams [Electronic resource]. – Access mode: https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf - 15.02.2016.

33. Shuja, A. K. IBM Rational Unified Process Reference and Certification Guide [Text] / A. K. Shuja, J. Krebs. – New York: IBM Press, 2008. – 336 p.

34. The Rules of Extreme Programming [Electronic resource]. – Access mode: <http://www.extremeprogramming.org/rules.html>. - 15.02.2016.

35. Beck, K. Extreme Programming Explained: Embrace Change [Text] / K. Beck, C. Andres. – Indianapolis: Addison-Wesley, 2004. – 224 p.

36. The Standard Waterfall Model for Systems Development [Electronic resource]. – Access mode: <http://scrum.kaverjody.com/wp-content/uploads/2013/01/2-nasa-waterfall-2005.pdf>. - 15.02.2016.

37. Selby, R.W. Software Engineering: Barry W. Boehm's Lifetime Contributions to Software Development, Management, and Research [Text] / R. W. Selby. – New York: John Wiley & Sons, 2007. – 818 p.

References

1. Khaf, L. *Metodologii razrabotki programmno-go obespecheniya* [Software development methodologies]. Available at: <http://compress.ru/article.aspx?id=11321> (accessed 15.02.2016).
2. Hovorushchenko, T. O., Malyarchuk, R. A. *Analiz protsesu vyboru tekhnologii proektivannya, metodologii ta seredovyshcha rozroblennya programnogo zabezpechennya* [Analysis of process of selection of software design technology, development methodology and environment]. Transactions of Khmelnytsky National University, 2014, no. 6, pp. 186-196.
3. Kobern, A. *Bystraya razrabotka programmno-go obespecheniya* [Rapid software development]. Moscow, Lori Publ. House, 2002. 314 p.
4. *Standish Group 2015 Chaos Report*. Available at: <http://www.infoq.com/articles/standish-chaos-2015> (accessed 15.02.2016).
5. *ISO 40500:2012. Information technology. W3C Web Content Accessibility Guidelines (WCAG) 2.0*. Introduced 15.10.2012. ISO/IEC, 2012. 2 p.
6. *ISO 9241-151:2008. Ergonomics of human-system interaction. Part 151: Guidance on World Wide Web user interfaces*. Introduced 15.05.2008. ISO/IEC, 2008. 49 p.
7. *ISO/IEC 40240:2011. Information technology - W3C Web Services Addressing 1.0. Core*. Introduced 01.09.2011. ISO/IEC, 2011. 1 p.
8. *ISO/IEC 40260:2011. Information technology. W3C Web Services Addressing 1.0. Metadata*. Introduced 01.09.2011. ISO/IEC, 2011. 1 p.
9. *Standards for Web Applications on Mobile* (2015). Available at: <https://www.w3.org/Mobile/mobile-web-app-state/> (accessed 15.02.2016).
10. *Mobile Web Application Best Practices* (2010). Available at: <https://www.w3.org/TR/mwabp/> (accessed 15.02.2016).
11. *Mobile Web Best Practices 1.0* (2008). Available at: <https://www.w3.org/TR/mobile-bp/> (accessed 15.02.2016).
12. *W3C mobileOK Basic Tests 1.0* (2008). Available at: <https://www.w3.org/TR/mobileOK-basic10-tests/> (accessed 15.02.2016).
13. Bianco, A. M., Marsico, M. D., Temperini, M. *Standards for e-learning*. Available at: http://www2.tisip.no/quis/public_files/wp5-standards-for-elearning.pdf (accessed 15.02.2016).
14. *SCORM 2004: 4th edition*. Available at: <https://www.adlnet.gov/adl-research/scorm/scorm-2004-4th-edition/> (accessed 15.02.2016).
15. *AICC Guidelines & Recommendations (AGR's)*. Available at: www.aicc.org (accessed 15.02.2016)
16. *SSARS No. 3, 4, 13, 14, 19, 21. Statements on Standards for Accounting and Review Services*. Available at: <http://www.aicpa.org/RESEARCH/STANDARDS/COMPILATIONREVIEW/Pages/compilation%20and%20review%20standards.aspx> (accessed 15.02.2016).
17. *SPSS Statistics Standard*. Available at: <http://www-03.ibm.com/software/products/en/spss-stats-standard> (accessed 15.02.2016).
18. *ISO 16100-1:2009. Industrial automation systems and integration. Manufacturing software capability profiling for interoperability. Part 1: Framework. Revises ISO 16100-1:2002* : introduced 15.12.2009. ISO/IEC, 2009. 45 p.
19. *ISO 16100-2:2003. Industrial automation systems and integration. Manufacturing software capability profiling for interoperability. Part 2: Profiling methodology*. Introduced 01.22.2003. ISO/IEC, 2003. 17 p.
20. *ISO 16100-6:2011. Industrial automation systems and integration. Manufacturing software capability profiling for interoperability. Part 6: Interface services and protocols for matching profiles based on multiple capability class structures*. Introduced 15.06.2011. ISO/IEC, 2011. 70 p.
21. *ITAF: A Professional Practices Framework for IS Audit/Assurance*. Available at: http://www.isaca.org/Knowledge-Center/Research/Documents/ITAF-3rd-Edition_fm_k_Eng_1014.pdf (accessed 15.02.2016).
22. *ISO/IEC 20000-1:2011. Information technology. Service management. Part 1: Service management system requirements. Revises ISO 20000-1:2005*. Introduced 15.04.2011. ISO/IEC, 2011. 26 p.
23. *ISO/IEC 90003:2014. Software engineering. Guidelines for the application of ISO 9001:2008 to computer software. Revises ISO 90003:2004*. Introduced 15.12.2014. ISO/IEC, 2014. 54 p.
24. *The Scrum Guide*. Available at: <http://www.scrumguides.org/scrum-guide.html> (accessed 15.02.2016).
25. Schwaber, K. *Agile Project Management with SCRUM: Developer Best Practices*. Redmond, Microsoft Press Publ., 2004. 188 p.
26. Cimorelli, S. *KANBAN for the Supply Chain: Fundamental Practices for Manufacturing Management*. Boca Raton, CRC Press Publ., 2013. 128 p.
27. *The Kanban System*. Available at: <http://isoconsultantpune.com/kanban-system/> (accessed 15.02.2016).
28. *DSDM Atern Handbook* (2008). Available at: <https://www.dsdm.org/content/delivering-quality> (accessed 15.02.2016).
29. Roberts, B. *Dynamic Systems Development Method, The Standard for Rapid Application Development*. Software Quality, 2001, part 1, pp. 16-33.
30. *Microsoft Solutions Framework Overview*. Available at: <https://msdn.microsoft.com/en-us/library/jj161047%28v=vs.120%29.aspx> (accessed 15.02.2016).
31. Villafiorita, A. *Introduction to Software Project Management*. Boca Raton, CRC Press Publ., 2014. 268 p.

32. *Rational Unified Process: Best Practices for Software Development Teams*. Available at: https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf (accessed 15.02.2016).

33. Shuja, A. K., Krebs, J. *IBM Rational Unified Process Reference and Certification Guide*. New York, IBM Press Publ., 2008. 336 p.

34. *The Rules of Extreme Programming*. Available at: <http://www.extremeprogramming.org/rules.html> (accessed 15.02.2016).

35. Beck, K., Andres, C. *Extreme Programming Explained: Embrace Change*. Indianapolis, Addison-Wesley, 2004. 224 p.

36. *The Standard Waterfall Model for Systems Development*. Available at: <http://scrum.kaverjody.com/wp-content/uploads/2013/01/2-nasa-waterfall-2005.pdf> (accessed 15.02.2016).

37. Selby, R. W. *Software Engineering: Barry W. Boehm's Lifetime Contributions to Software Development, Management, and Research*. New York, John Wiley & Sons Publ., 2007. 818 p.

Надійшла до редакції 15.02.2016, розглянута на редколегії 14.04.2016

МНОЖЕСТВЕННЫЕ ПРЕДСТАВЛЕНИЯ ТИПОВ И МЕТОДОЛОГИЙ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Р. А. Малярчук, Т. О. Говорущенко

Данная работа посвящена построению множественных представлений типов программного обеспечения (ПО) с учетом их существенных и показательных характеристик и методологий разработки ПО с учетом их основных принципов. Разработанные множественные представления обеспечивают заказчику и разработчику возможность подбора необходимой методологии для поставленной задачи путем сравнения множества необходимых характеристик ПО разрабатываемого типа с множествами принципов, которые обеспечиваются разными методологиями. Множественные представления типов и методологий разработки ПО будут использоваться в дальнейшем при моделировании и реализации процесса выбора методологии разработки ПО, для чего необходимо установить важность (весовые коэффициенты) характеристик для каждого типа ПО и принципов для каждой методологии разработки ПО.

Ключевые слова: программное обеспечение (ПО), программный проект, типы программного обеспечения, методология разработки ПО, характеристики ПО, принципы методологий разработки ПО.

SET REPRESENTATIONS OF TYPES AND DEVELOPMENT METHODOLOGIES OF SOFTWARE

R. A. Malyarchuk, T. O. Hovorushchenko

This paper is dedicated to the construction of the set representations of software types considering their basic essential and indicator characteristics, and the set representations of the software development methodologies considering their basic principles. The developed set representations provides to the customer and developer the possibility of selection of the necessary methodology for concrete task by comparing the set of the necessary software characteristics (for developed software type) with the sets of principles, that are provided by different methodologies. The set representations of software types and development methodologies will be used in the future for the modelling and realization of the process of software development methodology choice. For this we need to determine the importance (the weights) of the characteristics for each software type and of the principles for each software development methodology.

Key words: software, software project, software types, software development methodology, software characteristics, principles of the software development methodology.

Малярчук Роман Андрійович – аспірант кафедри системного програмування, Хмельницький національний університет, Хмельницький, Україна, e-mail: sayhello2fly@gmail.com.

Говорущенко Тетяна Олександрівна – канд. техн. наук, ст. науч. сотр., доцент кафедри системного програмування, Хмельницький національний університет, Хмельницький, Україна, e-mail: tat_yana@ukr.net.

Malyarchuk Roman Andriyovych – PhD Student of System Programming Department, Khmelnytsky National University, Khmelnytsky, Ukraine; e-mail: sayhello2fly@gmail.com.

Hovorushchenko Tetiana Oleksandrivna – PhD (Candidate of Technical Science), Senior Researcher, Associate Professor of System Programming Department, Khmelnytsky National University, Khmelnytsky, Ukraine; e-mail: tat_yana@ukr.net.