

УДК 004.891.3: 004.3

Р.А. МАЛЯРЧУК, Т.О. ГОВОРУЩЕНКО
Хмельницький національний університет

МОДЕЛЮВАННЯ ПРОЦЕСУ ВИБОРУ МЕТОДОЛОГІЇ РОЗРОБЛЕННЯ ДЛЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ РІЗНИХ ТИПІВ

У статті вперше виконано моделювання процесу вибору методології розроблення для програмного забезпечення різних типів, яке дало можливість зробити висновки щодо найбільш та найменш підходящих методологій розроблення для програмного забезпечення різних типів.

Ключові слова: програмне забезпечення (ПЗ), методологія розроблення ПЗ, характеристики ПЗ різних типів, ключові принципи методологій розроблення ПЗ.

R.A. MALYARCHUK, T.O. HOVORUSHCHENKO
Khmelnitsky National University

MODELLING OF SELECTION OF SOFTWARE DEVELOPMENT METHODOLOGY FOR SOFTWARE OF DIFFERENT TYPES

The article describes the dependence of the successful implementation of software of different types and sizes from selection of the software development methodology. In the article the modelling of selection of software development methodology for software of various types was conducted by comparing the previously identified characteristics of different types of software (Web-applications, mobile applications, e-learning programs, applications for statistics and accounting), automated systems and information systems) and the key principles of modern software development methodologies (SCRUM, KANBAN, DSDM, MSF, RUP, XP, Waterfall). Conducted modelling has made it possible to draw conclusions about the most and least appropriate methodologies for development of software of different types.

Keywords: software, software development methodology, characteristics of different types of software, key principles of software development methodologies.

Вступ

Наразі існує велика кількість методологій розроблення ПЗ, але відсутні чіткі стандартизовані критерії вибору методології для розроблення ПЗ певного типу, в той час як від вибору методології розроблення ПЗ залежить успішність реалізації програмного проекту різних типів та розмірів – рис. 1 [1].

CHAOS RESOLUTION BY AGILE VERSUS WATERFALL

SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small size Projects	Agile	56%	38%	4%
	Waterfall	44%	45%	11%

The resolution of all software projects from FY2011-2016 within the new CHROS database, segmented by the agile process and waterfall method. The total number of software projects is over 10,000.

Рис. 1. Порівняння успішності програмних проектів, реалізованих за каскадними та AGILE-методологіями

При виборі методології розроблення ПЗ зараз не враховується тип та цільове призначення програмного проекту, хоча автор [2] підкреслює важливість вибору не «найкращої» методології, а тієї, яка найкращим чином відповідає проектній задачі та колективу розробників. Відтак 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.

У [3] було визначено основні характеристики різних типів ПЗ (Web-додатків, мобільних додатків, програм електронного навчання, прикладних програм (для статистики та бухгалтерії), автоматизованих систем та інформаційних систем) і основних принципів сучасних методологій розроблення ПЗ (SCRUM, KANBAN, DSDM, MSF, RUP, XP, Waterfall), 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. Тому метою даного

дослідження є моделювання процесу вибору методології розроблення програмного забезпечення для програмних проектів різних типів.

Моделювання процесу вибору методології розроблення програмного забезпечення для програмних проектів різних типів

У [3] автори представили кожен розглядуваний тип ПЗ у формалізованому вигляді – у вигляді пари $T_i = \langle EC_{T_i}, IC_{T_i} \rangle$, де EC_{T_i} – множина сутнісних характеристик ПЗ i -го типу, IC_{T_i} – множина показникових характеристик, які необхідно забезпечити при реалізації ПЗ i -го типу.

Крім цього, у [3] авторами було представлено у вигляді множин M_j основних принципів також і всі розглядувані методології розроблення ПЗ.

Тоді процес вибору (підбору) методології розроблення ПЗ для програмних проектів різних типів промодельюємо за допомогою операції перетину множин сутнісних та показникових характеристик ПЗ кожного типу із множинами основних принципів всіх розглядуваних методологій розроблення ПЗ:

1) для Web-додатків:

$$EC_{Web} \cap M_{SCRUM} = \{Par, Cmrt, Fbu\},$$

$$IC_{Web} \cap M_{SCRUM} = \{Ia\},$$

де Par – можливість внесення змін до вимог протягом всього життєвого циклу проекту, $Cmrt$ – можливість керування розробкою та контентом в режимі реального часу, Fbu – зворотній зв'язок з користувачем, Ia – висока інтерактивність (високий рівень взаємодії із зовнішніми джерелами інформації та зовнішнім ПЗ);

$$EC_{Web} \cap M_{KANBAN} = \{Cmrt, Par, Dv, Fbu\},$$

$$IC_{Web} \cap M_{KANBAN} = \{Po_{Web}\},$$

де Dv – візуалізація розроблення, $Po_{Web} = \{Pr, Sa, Is, Rt, Bw, Om, Sb\}$ – оптимізація процесу (Pr – висока продуктивність, Sa – відносно невеликий розмір додатку, Is – висока швидкість взаємодії, Rt – малий час відгуку, Bw – висока пропускну здатність, Om – відносно невеликий обсяг оперативної пам'яті, Sb – висока здатність збільшувати продуктивність за потреби);

$$EC_{Web} \cap M_{DSDM} = \{Cmrt, Par, Sfr, Fbu, Tac\},$$

$$IC_{Web} \cap M_{DSDM} = \{Ia, Qua\},$$

де Sfr – вивчення можливості реалізації ПЗ та галузі його застосування, Tac – тестування протягом всього циклу робіт, $Qua = \{Ub, Ss, Rb, Fc\}$ – висока якість (згідно стандарту ISO25010 [4], зручність використання, безпека та захищеність, надійність та функційна повнота – є основними характеристиками якості ПЗ);

$$EC_{Web} \cap M_{MSF} = \{Par, Fbu, Bpc\},$$

$$IC_{Web} \cap M_{MSF} = \{Sb\};$$

$$EC_{Web} \cap M_{RUP} = \{Par, Fbu\},$$

$$IC_{Web} \cap M_{RUP} = \{Qua, Ier\},$$

де Ier – рання ідентифікація та неперервне усунення можливих ризиків;

$$EC_{Web} \cap M_{XP} = \{Cmrt, Par, Fbu, CmsH\},$$

$$IC_{Web} \cap M_{XP} = \{Cig\},$$

де $CmsH$ – постійний зв'язок із зацікавленими сторонами проекту, Cig – неперервна інтеграція;

$$EC_{Web} \cap M_{Waterfall} = \emptyset,$$

$$IC_{Web} \cap M_{Waterfall} = \{Ub\};$$

2) для мобільних додатків:

$$EC_{Mob} \cap M_{SCRUM} = \{Par, Cmrt, Fbu\},$$

$$IC_{Mob} \cap M_{SCRUM} = \{Ia\};$$

$$EC_{Mob} \cap M_{KANBAN} = \{Par, Cmrt, Dv, Fbu, Ds\},$$

$$IC_{Mob} \cap M_{KANBAN} = \{Ds, Po_{Mob}\},$$

де Ds – орієнтація на задачі, $Po_{Mob} = \{Ts, Rt, Om, Pr, Emm, Cts\}$ – оптимізація процесу (Ts – малий час запуску додатка, Rt – малий час відгуку додатку, Om – відносно невеликий обсяг оперативної пам'яті, Pr – висока продуктивність, Emm – високоєфективне керування пам'яттю, Cts – низька складність виконання задач);

$$EC_{Mob} \cap M_{DSDM} = \{Par, Cmrt, Sfr, Fbu, Tac\},$$

$$IC_{Mob} \cap M_{DSDM} = \{Ia, Qua\};$$

$$EC_{Mob} \cap M_{MSF} = \{Par, Fbu, Bpc\},$$

$$\begin{aligned}
 IC_{Mob} \cap M_{MSF} &= \emptyset ; \\
 EC_{Mob} \cap M_{RUP} &= \{Par, Fbu\} , \\
 IC_{Mob} \cap M_{RUP} &= \{Qua, Ier\} ; \\
 EC_{Mob} \cap M_{XP} &= \{Par, Cmrt, Fbu, Cmsh\} , \\
 IC_{Mob} \cap M_{XP} &= \{Cig\} ; \\
 EC_{Mob} \cap M_{Waterfall} &= \emptyset , \\
 IC_{Mob} \cap M_{Waterfall} &= \{Ub\} ;
 \end{aligned}$$

3) для програм електронного навчання:

$$\begin{aligned}
 EC_{E-learn} \cap M_{SCRUM} &= \{Par, Fbu\} , \\
 IC_{E-learn} \cap M_{SCRUM} &= \{Ia\} ; \\
 EC_{E-learn} \cap M_{KANBAN} &= \{Par, Dv, Fbu\} , \\
 IC_{E-learn} \cap M_{KANBAN} &= \{Po_{E-learn}\} ,
 \end{aligned}$$

де $Po_{E-learn} = \{Io, Fio\}$ – оптимізація процесу (Io – висока сумісність як можливість використовувати навчальні об'єкти, розташовані в різних місцях, створені різними інструментами та на різних платформах), Fio – висока функційна сумісність як збереження роботоздатності при використанні різного програмного та апаратного забезпечення);

$$\begin{aligned}
 EC_{E-learn} \cap M_{DSDM} &= \{Par, Sfr, Fbu, Tac\} , \\
 IC_{E-learn} \cap M_{DSDM} &= \{Ia, Qua\} ; \\
 EC_{E-learn} \cap M_{MSF} &= \{Par, Fbu, Ucs\} , \\
 IC_{E-learn} \cap M_{MSF} &= \emptyset ,
 \end{aligned}$$

де Ucs – єдине бачення проекту (використання єдиних стандартів);

$$\begin{aligned}
 EC_{E-learn} \cap M_{RUP} &= \{Par, Fbu, Cmpa\} , \\
 IC_{E-learn} \cap M_{RUP} &= \{Qua, Ier\} ; \\
 EC_{E-learn} \cap M_{XP} &= \{Par, Fbu, Cmsh\} , \\
 IC_{E-learn} \cap M_{XP} &= \{Cig\} ; \\
 EC_{E-learn} \cap M_{Waterfall} &= \{Hfd\} , \\
 IC_{E-learn} \cap M_{Waterfall} &= \{Ub\} ,
 \end{aligned}$$

де Hfd – високий ступінь формалізації;

4) для прикладних програм (для статистики та бухгалтерії):

$$\begin{aligned}
 EC_{Appl} \cap M_{SCRUM} &= \{Par, Fbu\} , \\
 IC_{Appl} \cap M_{SCRUM} &= \emptyset ; \\
 EC_{Appl} \cap M_{KANBAN} &= \{Par, Dv, Fbu, Ds\} , \\
 IC_{Appl} \cap M_{KANBAN} &= \{Po_{Appl}\} ,
 \end{aligned}$$

де $Po_{Appl} = \{Io, Udf\}$ – оптимізація процесу (Io – висока сумісність (з іншим ПЗ фірми-розробника), Udf – висока уніфікація форматів представлення різних даних);

$$\begin{aligned}
 EC_{Appl} \cap M_{DSDM} &= \{Par, Sfr, Fbu, Tac\} , \\
 IC_{Appl} \cap M_{DSDM} &= \{Qua\} ; \\
 EC_{Appl} \cap M_{MSF} &= \{Par, Fbu, Ucs, Bpc\} , \\
 IC_{Appl} \cap M_{MSF} &= \{Sb\} ; \\
 EC_{Appl} \cap M_{RUP} &= \{Par, Fbu, Cmpa\} , \\
 IC_{Appl} \cap M_{RUP} &= \{Frr, Qua, Ier\} ,
 \end{aligned}$$

де Frr – концентрація на виконанні вимог замовників (повне покриття потреб предметної галузі, високий ступінь задоволення запитів та потреб користувача);

$$\begin{aligned}
 EC_{Appl} \cap M_{XP} &= \{Par, Fbu, Cmsh\} , \\
 IC_{Appl} \cap M_{XP} &= \{Cig\} ; \\
 EC_{Appl} \cap M_{Waterfall} &= \{Hfd\} , \\
 IC_{Appl} \cap M_{Waterfall} &= \{Ub\} ;
 \end{aligned}$$

5) для автоматизованих систем:

$$EC_{AS} \cap M_{SCRUM} = \{Par, Fbu\},$$

$$IC_{AS} \cap M_{SCRUM} = \emptyset;$$

$$EC_{AS} \cap M_{KANBAN} = \{Par, Fbu, Ds\},$$

$$IC_{AS} \cap M_{KANBAN} = \{Po_{AS}\},$$

де $Po_{AS} = \{Io, Rm, Em, Scpd\}$ – оптимізація процесу (Io – висока сумісність (організаційна, технічна, програмна, математична, лінгвістична), Rm – висока оперативність керування процесом, Em – висока ефективність керування об'єктом та виробництвом, $Scpd$ – висока швидкість збирання та опрацювання даних);

$$EC_{AS} \cap M_{DSDM} = \{Par, Sfr, Fb, Tac\},$$

$$IC_{AS} \cap M_{DSDM} = \{Qua\};$$

$$EC_{AS} \cap M_{MSF} = \{Par, Fbu, Bpc\},$$

$$IC_{AS} \cap M_{MSF} = \emptyset;$$

$$EC_{AS} \cap M_{RUP} = \{Par, Fbu\},$$

$$IC_{AS} \cap M_{RUP} = \{Qua, Ier\};$$

$$EC_{AS} \cap M_{XP} = \{Par, Fbu, CmsH\},$$

$$IC_{AS} \cap M_{XP} = \{Cig\};$$

$$EC_{AS} \cap M_{Waterfall} = \{Hfd\},$$

$$IC_{AS} \cap M_{Waterfall} = \{Ub, Rr, Rss, Sm, Qd\};$$

де Rr – високий ступінь обґрунтованості прийнятих рішень (низькі ризики), Rss – жорстка послідовність проходження етапів (високий рівень контролю), Sm – висока ефективність, простота та зручність керування, Qd – відносно мала кількість рішень;

б) для інформаційних систем:

$$EC_{IS} \cap M_{SCRUM} = \{Par, Fbu, Cmrt\},$$

$$IC_{IS} \cap M_{SCRUM} = \emptyset;$$

$$EC_{IS} \cap M_{KANBAN} = \{Par, Fbu, Cmrt\},$$

$$IC_{IS} \cap M_{KANBAN} = \{Po_{IS}\},$$

де $Po_{IS} = \{Scpd, Dm, Efip\}$ – оптимізація процесу ($Scpd$ – висока швидкість збирання та опрацювання даних (швидкодія), Dm – висока динамічність (зміна структури та станів елементів в часі), $Efip$ – висока ефективність опрацювання інформації);

$$EC_{IS} \cap M_{DSDM} = \{Par, Sfr, Fbu, Cmrt, Tac\},$$

$$IC_{IS} \cap M_{DSDM} = \{Qua\};$$

$$EC_{IS} \cap M_{MSF} = \{Par, Fbu, Bpc\},$$

$$IC_{IS} \cap M_{MSF} = \emptyset;$$

$$EC_{IS} \cap M_{RUP} = \{Par, Fbu\},$$

$$IC_{IS} \cap M_{RUP} = \{Qua, Frr, Ier\};$$

$$EC_{IS} \cap M_{XP} = \{Par, Fbu, Cco, Cmrt, CmsH\},$$

$$IC_{IS} \cap M_{XP} = \{Cig\};$$

де Cco – колективне володіння кодом, можливість колективного доступу;

$$EC_{IS} \cap M_{Waterfall} = \emptyset,$$

$$IC_{IS} \cap M_{Waterfall} = \{Ub\}.$$

Враховуючи виконане моделювання для описаних у [3] типів ПЗ та методологій, сформулюємо наступні висновки:

1) для Web-додатків найбільш підходящою є методологія DSDM, а найменш підходящою є каскадна (Waterfall) методологія;

2) для мобільних додатків найбільш підходящими є методології DSDM та KANBAN, а найменш підходящою є каскадна (Waterfall) методологія;

3) для програм електронного навчання найбільш підходящою є методологія DSDM, а найменш підходящою є каскадна (Waterfall) методологія;

4) для прикладних програм найбільш підходящою є методологія RUP, а найменш підходящою є каскадна (Waterfall) методологія;

5) для автоматизованих систем найбільш підходящою є методологія каскадна (Waterfall)

методологія, а найменш підходящою є методологія Scrum;

б) для інформаційних систем найбільш підходящими є методології DSDM та XP, а найменш підходящою є каскадна (Waterfall) методологія.

Таке моделювання не є абсолютно ефективним, оскільки воно не враховує важливість характеристик для ПЗ різних типів; крім цього, не враховує також протилежності деяких характеристик – наприклад, *Par* – можливість внесення змін до вимог протягом всього життєвого циклу проекту (висока гнучкість), а *Npar* – відсутня можливість внесення змін до вимог протягом життєвого циклу проекту (низька гнучкість).

Висновки

Від вибору методології розроблення ПЗ залежить успішність реалізації програмного проекту різних типів та розмірів. При виборі методології розроблення ПЗ зараз не враховується тип та цільове призначення програмного проекту, хоча кожна категорія проектів має різні пріоритети і цілі, тому важливою і актуальною є задача вибору методології розроблення для програмного забезпечення конкретного типу.

У статті було проведено моделювання процесу вибору методології розроблення програмного забезпечення для програмних проектів різних типів шляхом порівняння раніше визначених основних характеристик різних типів ПЗ (Web-додатків, мобільних додатків, програм електронного навчання, прикладних програм (для статистики та бухгалтерії), автоматизованих систем та інформаційних систем) і основних принципів сучасних методологій розроблення ПЗ (SCRUM, KANBAN, DSDM, MSF, RUP, XP, Waterfall).

Виконане моделювання дало можливість зробити висновки щодо найбільш та найменш підходящих методологій розроблення для ПЗ різних типів. Але моделювання не є абсолютно ефективним, оскільки воно не враховує важливість певних характеристик для ПЗ різних типів та ключових принципів для методологій; крім цього, не враховує також протилежності деяких характеристик – наприклад, *Par* - можливість внесення змін до вимог протягом всього життєвого циклу проекту (висока гнучкість), а *Npar* - відсутня можливість внесення змін до вимог протягом життєвого циклу проекту (низька гнучкість).

Тому перспективою для подальших досліджень авторів є:

- 1) визначення вагових коефіцієнтів характеристик ПЗ та ключових принципів методологій;
- 2) розроблення методу вибору ефективної методології розроблення для ПЗ різних типів;
- 3) розроблення інформаційної технології вибору ефективної методології розроблення для ПЗ різних типів.

На вирішення поставлених задач і будуть спрямовуватись подальші зусилля авторів.

Література

1. Standish Group 2015 Chaos Report. URL : <http://www.infoq.com/articles/standish-chaos-2015> - 15.02.2016
2. Коберн А. Быстрая разработка программного обеспечения / А. Коберн. – М. : Лори, 2002. – 314 с.
3. Малярчук Р. Множинні представлення типів та методологій розроблення програмного забезпечення / Р. Малярчук // Радіоелектронні і комп'ютерні системи. – Харків : НАУ “ХАІ”, 2016. – № 5 (79). – С. 70-77.
4. ISO/IEC 25010:2011. Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). System and software quality models. Revises ISO/IEC 9126-1:2001: introduced 01.03.2011. – ISO/IEC, 2011. – 34 p.

Рецензія/Peer review : 1.11.2016 р.

Надрукована/Printed : 15.12.2016 р.

Рецензент: д.т.н., професор, Сорокатиї Р.В.