

КОНЦЕПЦІЯ ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ, МЕТОДОЛОГІЙ ТА СЕРЕДОВИЩ РОЗРОБЛЕННЯ ДЛЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ РІЗНИХ ТИПІВ

Р.А.Малярчук, Т.О.Говоруценко

Хмельницький національний університет, Хмельницький, Україна
tat_yana@ukr.net, sayhello2fly@gmail.com

Вступ

З аналізу галузі створення програмного забезпечення (ПЗ) випливає, що наразі для створення повнофункціональних програмних додатків програмістам доводиться використовувати комплексні засоби розроблення - технологію проектування ПЗ, методологію розроблення ПЗ та середовище розроблення ПЗ.

Питанням оцінювання ефективності технологій проектування, методологій та середовища розроблення для ПЗ певного типу слід приділяти підвищену увагу: як показує досвід, без ефективної технології/методології/середовища навіть невеликі програмні проекти навряд чи можуть бути успішними.

Статистика успішності програмних проектів за даними The Standish Group International [1] представлена на рис. 1.

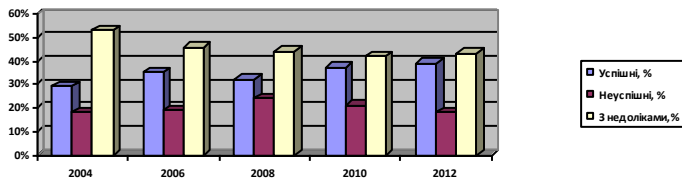


Рис.1. Статистика успішності впровадження програмних проектів за даними The Standish Group International [1]

Аналіз даних з рис. 1 дав можливість побачити, що частка проектів з недоліками є досить сталою величиною і складає мінімум 42%. При цьому лише невелика кількість (максимум 39% за 1994-2012 роки) проектів має необхідні функціональні можливості при існуючих обмеженнях за вартістю та терміном [1].

З проведеного аналізу відомих технологій проектування, методологій та середовищ розроблення [2] очевидним є існування значної кількості технологій/методологій/середовищ за відсутності універсальної технології/методології/ середовища, яка підходила б для будь-якого програмного проекту, а також за відсутності чітких стандартизованих критеріїв оцінювання та вибору технології проектування, методології та середовища розроблення ПЗ.

Тому при всій множині існуючих технологій/методологій/середовищ кількість програмних проектів з недоліками та неуспішних програмних проектів залишається досить високою (61% у 2012 році), а процес розроблення ПЗ залишається недетермінованим, і результат його завжди невідомий. Такий стан справ можна пояснити наступною причиною в контексті технологій/методологій/середовищ - їх неефективністю для ПЗ певного типу.

Отже, задача оцінювання ефективності технологій проектування, методологій та середовищ розроблення для ПЗ різних типів є актуальною на етапі проектування, оскільки від ефективності технології/методології/середовища залежить майбутня успішність програмного проекту.

Сучасний стан галузі вибору технології проектування, методології та середовища розроблення ПЗ

Галузеві публікації вказують на часте використання тієї чи іншої *технології проектування* для ПЗ певного типу:

1) технологія RUP часто використовується при створенні систем автоматизованого проектування, CASE-систем, систем з елементами штучного інтелекту, а також систем підтримки структурно-параметричного синтезу об'єктів [3];

2) технологія Borland застосовується як софтверними компаніями, які реалізують великі дорогі програмні проекти, так і невеликими софтверними компаніями завдяки демократичним цінам багатьох редакцій цієї технології [4];

3) технологія Oracle призначена для швидкого створення та розгортання Інтернет-додатків, динамічних Web-порталів та розгортання Web-сервісів [5].

Дані відомості можна використовувати як рекомендації щодо використання технології проектування для ПЗ різних типів, але в галузевих публікаціях відсутня інформація про те, які (тип та кількість) з цих програмних проектів є успішними, тобто невідомо, чи є ефективною та чи інша технологія для ПЗ певного типу.

Оцінювання ефективності *методології розроблення* взагалі вимагає врахування багатьох різноманітних умов та факторів [6]:

1) масштаб проекту; 2) критичність проекту; 3) кількість та розподіл повноважень учасників проекту; 4) ступінь новизни проекту; 5) очікувана тривалість проекту; 6) вимоги замовника; 7) специфіка та складність проекту.

Знов-таки галузеві публікації дають можливість зробити наступні висновки щодо використання методологій проектування:

1) для великих, розрахованих на тривалий термін, проектів, як правило, використовуються методології з більшим ступенем

формалізації – традиційні та ітеративні методології, в той час як для менш масштабних проєктів, розрахованих на оперативну реалізацію, підходять менш формалізовані гнучкі методології [6];

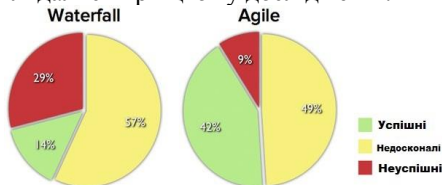
2) методологія RUP використовується для масштабних та тривалих проєктів [7];

3) методологія MSF є оптимальною для великих проєктів, які вимагають дотримання балансу між ресурсами, часом розроблення та можливостями [7];

4) методологія SCRUM ідеальна для невеликих та середніх проєктів, особливо, якщо під час процесу розроблення очікується внесення численних змін [8];

5) методологія XP розрахована на невеликі проєкти, в яких не виникає необхідності у створенні детальної документації та регламентації всіх кроків розроблення [9].

Отже, галузеві публікації знову не відображають кількості успішних проєктів, розроблених з використанням тієї чи іншої методології. Єдине джерело, яке вносить хоч якусь ясність у це питання, це діаграма порівняння успішності програмних проєктів, реалізованих за каскадними та AGILE-методологіями (рис.2) [10], яка показує, що AGILE-проєкти втричі успішніші за каскадні проєкти, але з неї незрозуміло, яку саме з гнучких методологій було використано для успішних програмних проєктів, а також проєкти якого типу розглядались при цьому дослідженні.



Джерело: The CHAOS Manifesto, The Standish Group, 2012.

Рис.2. Порівняння успішності програмних проєктів, реалізованих за каскадними та AGILE- методологіями

Оцінювання ефективності *середовища розроблення ПЗ* теж зводиться до багатокритеріальної задачі і далеко не очевидне. Галузеві публікації показують наступне використання середовищ розроблення для ПЗ різних типів:

1) середовище C++ Builder дозволяє створювати як консольні додатки, так і додатки з графічним інтерфейсом користувача [11];

2) Code::Blocks використовується для багатоцільових програмних проєктів [12];

3) MonoDevelop дає можливість розробникам швидко створювати додатки для кишенькових пристроїв та web-додатки

ASP.NET для Linux, Windows, MacOSX; дозволяє створення web-проектів на XSP [13];

4) Microsoft Visual Studio дозволяє розробляти як консольні додатки, так і додатки з графічним інтерфейсом, а також web-сторінки, web-додатки та web-служби [14];

5) NetBeans підтримує розроблення для платформ J2SE і J2EE [15].

Галузеві публікації також не показують, скільки програмних проектів стали успішними в результаті розроблення в тому чи іншому середовищі.

При оцінюванні технології проектування, методології та середовища розроблення ПЗ розробники виділяють наступні проблеми [16]: 1) відсутність єдиної бази знань про технології/методології/середовища; 2) відсутність єдиного стандарту для оцінювання технологій/методологій/середовищ; 3) відсутність єдиних критеріїв, за якими оцінюються технології/методології/середовища для ПЗ різних типів. Оцінювання ефективності технології/методології/середовища може підвищити якість ПЗ, а також знизити витрати на його розроблення.

Концепція оцінювання ефективності технології проектування, методології та середовища розроблення для ПЗ різних типів

Зрозуміло, що кожна софтверна компанія використовує вже придбані та впроваджені технологію проектування, методологію та середовище(а) розроблення, для роботи з якими є достатня кількість фахівців. Навряд чи якась софтверна компанія погодиться купувати та впроваджувати іншу технологію/методологію/середовище, перенавчати своїх або наймати інших фахівців, яку система підтримки прийняття рішень визначила як більш переважну для ПЗ саме такого типу. Отже, розроблення системи підтримки прийняття рішень щодо вибору технології/методології/середовища не є нагальною задачею, тому що результати її роботи навряд чи використовуватимуть компанії-розробники.

Дослідницька задача полягає у іншому – визначити, на скільки вдало вдасться спроектувати та розробити ПЗ певного типу, використовуючи певні технологію проектування, методологію та середовище розроблення, тобто на скільки успішним буде програмний проект із використанням тієї чи іншої технології/методології/середовища. Отже, іншими словами, *актуальною науковою задачею* є визначення ефективності конкретної технології проектування, методології та середовища розроблення для ПЗ певного типу.

Якщо оцінювати ефективність за адитивним критерієм, то матимемо формулу вигляду (1):

$$E = \sum_{i=1}^n a_i \cdot x_i, \quad (1)$$

де x_i - характеристики ПЗ (наприклад, якість, надійність, зручність використання, вартість, тривалість, швидкодія): $x_i \in [0;1]$, де 0 – показує найменшу можливість реалізації характеристики з використанням певної технології/ методології/середовища, а 1 – найбільшу можливість реалізації характеристики з використанням певної технології/ методології/середовища; a_i - вагові коефіцієнти характеристик, які відображають роль та пріоритетність конкретної характеристики для даного ПЗ, тобто відображають роль характеристики для ПЗ даного типу з врахуванням цільового призначення та типу програмного проекту: нехай $a_i \in [-10;10]$, але кожний ваговий коефіцієнт a_i повинен обмежуватись своїм діапазоном в загальному діапазоні – саме цей діапазон вказуватиме важливість характеристики x_i для конкретного ПЗ (наприклад, для програмної системи електронного навчання характеристика «зручність використання» повинна мати ваговий коефіцієнт $a_1 \in [-10;10]$, а характеристика «швидкодія» може мати ваговий коефіцієнт $a_2 \in [-1;1]$).

Висновки

Вирішення задачі оцінювання ефективності конкретної технології проектування, методології та середовища розроблення для ПЗ певного типу допоможе розробнику оцінити, чи варто йому братись за той чи інший програмний проект за наявної технології/методології/ середовища (і не витратити часу та коштів на неуспішні програмні проекти), а замовнику – оцінити, наскільки вдалим буде програмний проект, якщо його реалізуватиме конкретна софтверна фірма, тобто дозволить зробити обґрунтований висновок про підходящість або невідходящість технології/методології/середовища, а відтак і софтверної компанії, яка їх використовує, для розроблення конкретного ПЗ або ПЗ певного типу на основі отриманого значення ефективності.

Для оцінювання ефективності технологій проектування, методологій та середовищ розроблення для ПЗ певного типу слід вирішити наступні задачі:

1) провести аналіз софтверних компаній України та світу за наступними критеріями: які вони використовують технології/методології/середовища; яке ПЗ розробляють; які

значення (ступені виконання) основних характеристик ПЗ забезпечують; типи та кількість успішних і неуспішних (через невідповідність технології/методології/середовища) проектів;

2) визначити значення основних характеристик x_i ПЗ з точки зору можливості їх реалізації з використанням певної технології/ методології/середовища на основі аналізу п.1;

3) визначити роль та пріоритетність характеристик для ПЗ кожного типу з врахуванням цільового призначення та типу програмного проекту, а також визначення діапазонів значень вагових коефіцієнтів a_i для ПЗ різних типів та різного призначення. Це є дуже важливою задачею, тому що саме вагові коефіцієнти a_i повинні не давати можливості компенсувати низьке значення зrealізованості важливої характеристики високим значенням зrealізованості іншої, менш важливої, характеристики (наприклад, для програмної системи електронного навчання високе значення можливості реалізації характеристики «швидкодія» не повинне в жодному разі компенсувати низького значення можливості реалізації характеристики «зручність використання», оскільки значущість цих характеристик абсолютно нерівнозначна для такої системи). Отже, значення вагових коефіцієнтів a_i повинні бути підібрані таким чином, щоб при неможливості (або низькій можливості) реалізації важливої характеристики конкретного ПЗ певною технологією/методологією/середовищем ефективність технології/методології/середовища для розроблення цього ПЗ дуже стрімко зменшувалась;

4) обґрунтовано обрати компоненти (статистичні або інтелектуальні) для визначення значень основних характеристик x_i і, особливо, для визначення діапазонів та значень вагових коефіцієнтів a_i - в залежності від обсягу отриманої інформації;

5) на основі обраних компонентів розробити моделі, методи та програмні засоби оцінювання ефективності конкретної технології проектування, методології та середовища розроблення для ПЗ певного типу.

Список використаних джерел

1. CHAOS Manifesto: Think Big, Act Small, 2013 // [Electronic resource] - Access mode: <http://www.versionone.com/assets/img/files/CHAOSManifesto2013.pdf>
2. Т.О.Говорущенко. Аналіз процесу вибору технології проектування, методології та середовища розроблення програмного забезпечення / Т.О.Говорущенко, Р.А.Маларчук //

Вісник Хмельницького національного університету – Хмельницький: ХНУ, 2014. - №6

3. RUP – рациональный унифицированный процесс разработки программного обеспечения // [Электронный ресурс] – Режим доступа: <http://www.structuralist.narod.ru/dictionary/rup.htm>
4. М.Фаулер. Рефакторинг: улучшение существующего кода / М.Фаулер. – СПб.: Символ-Плюс, 2012. – 489 с.
5. Oracle Developer Suite // [Electronic resource] – Access mode: <http://www.interface.ru/oracle/OracleDS10g.htm>
6. Разработка программного обеспечения: Методологии разработки программного обеспечения // [Электронный ресурс] – Режим доступа: <http://bms-soft.com.ua/ru/services/razrabotka-programmnogo-obespecheniya/metodologii-razrabotki-po>
7. Р. К. Мартин. Быстрая разработка программ: принципы, примеры, практика / Р. К. Мартин, Д. В. Ньюкирк, Р. С. Косс – М.: Вильямс, 2004. – 752 с.
8. Обзор методологии разработки программного обеспечения SCRUM // [Электронный ресурс] – Режим доступа: <http://www.dpgrup.ru/methodology-scrum.htm>
9. Разработка программного обеспечения: Методологии разработки программного обеспечения // [Электронный ресурс] – Режим доступа: <http://bms-soft.com.ua/ru/services/razrabotka-programmnogo-obespecheniya/metodologii-razrabotki-po>
10. Agile Succeeds Three Times More Often Than Waterfall // [Electronic resource] – Access mode: <http://www.mountaingoatsoftware.com/blog/agile-succeeds-three-times-more-often-than-waterfall>
11. Д.Холингворт. Borland C++ Builder 6. Руководство разработчика / Д.Холингворт, Б.Сворт, М.Кэшмэн, П.Густавсон.— М.: «Вильямс», 2004. — 976 с.
12. Code::Blocks features // [Electronic resource] – Access mode: <http://www.codeblocks.org/features>
13. MonoDevelop // [Electronic resource] – Access mode: <http://monodevelop.com/>
14. Н.Рендольф. Visual Studio 2010 для профессионалов / Н.Рендольф, Д.Гарднер, М.Минутилло, К.Андерсон — М.: «Диалектика», 2011. —1184 с.
15. Монахов В.В. Язык программирования Java и среда NetBeans / Монахов В.В. – СПб.: БХВ-Петербург, 2011. – 704 с.
16. Т.О.Говорущенко. Підтримка вибору середовища програмування для системного програмного забезпечення / Т.О.Говорущенко, Р.А.Малярчук, В.В.Лаврінчук // ІТСП-2014. III Всеукраїнська НПК молодих учених та студентів. Збірник наукових праць - Хмельницький: Гонта А.С., 2014 - с.372-381