

## ОСНОВНІ ПРИНЦИПИ МЕТОДОЛОГІЙ РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

**Малярчук Р.О., Говорушенко Т.О.**

*Хмельницький національний університет; 29016, м.Хмельницький, вул.Інститутська,11;  
[sayhello2fly@gmail.com](mailto:sayhello2fly@gmail.com), [tat\\_yana@ukr.net](mailto:tat_yana@ukr.net)*

**Вступ.** Методологія розроблення програмного забезпечення (ПЗ) – це набір методів та критеріїв оцінки, які використовуються для постановки задачі, планування, контролю і для досягнення поставленої мети [1]. Процес розроблення описується моделлю, яка визначає послідовність найбільш загальних етапів та одержуваних результатів. Саме методологія визначає, як виконуватиметься розроблення ПЗ. Методології представляють собою ядро теорії керування розробленням ПЗ [2]. Сьогодні найбільш відомими та використовуваними є наступні методології: «швидкі» Agile (SCRUM, XP, DSDM, Kanban та інші), прогнозуючі та ітеративні Rational Unified Process (RUP), водоспадні (каскадні), Microsoft Solutions Framework (MSF), яка поєднує принципи каскадної та спіральної моделей, та інші [3].

**Постановка задачі.** Отже, існує велика кількість методологій, але відсутні чіткі стандартизовані критерії вибору методології для розроблення ПЗ певного типу (лише у роботі [4] введено декілька метрик, які дозволяють частково формалізувати процес підбору методології), а від вибору методології розроблення ПЗ залежить успішність реалізації програмного проекту. 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 розробниками та замовниками, зводити до мінімуму витрати на супровід ПЗ. Алістер Коберн у [4] підкреслює важливість вибору не «найкращої» методології розроблення, а тієї, яка найкращим чином відповідає проектній задачі та колективу розробників. Отже, 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.

**Основні принципи методологій розроблення ПЗ.** Методологія розроблення ПЗ – це система принципів, а також сукупність ідей, понять, характеристик, методів, способів та засобів, які визначають стиль розроблення ПЗ [2]. Аналіз фахової літератури дозволив визначити основні принципи таких методологій розроблення ПЗ, як: SCRUM, KANBAN, DSDM, MSF, RUP, XP та каскадної (waterfall).

Основні принципи методологій представлено у таблицях 1 і 2.

Таблиця 1. Основні принципи методологій SCRUM, KANBAN, DSDM, MSF

SCRUM	KANBAN	DSDM	MSF
Невеликі команди розробників Розбиття на ітерації (спринти) тривалістю 30 днів Незмінність функцій під час однієї ітерації Суворе дотримання термінів випуску чергового релізу (навіть якщо не вдається реалізувати весь планований функціонал)	Орієнтація на задачі Візуалізація розроблення Обмеження одночасних робіт на кожному етапі розроблення Вимірювання часу циклу (середній час на одну задачу) Оптимізація процесу	Вивчення можливості реалізації ПЗ та галузі його застосування Розбиття процесу на три взаємозв'язаних цикли (цикл функційної моделі, цикл проектування та конструювання, цикл реалізації) Активна взаємодія з користувачами Часті випуски версій (релізів) Самостійність розробників в прийнятті рішень Тестування протягом всього циклу робіт Короткі ітерації (2-6 тижнів кожна) Висока якість Адаптованість до змін у вимогах	Єдине бачення проекту Керування компромісами Гнучкість (готовність до змін) Концентрація на бізнес-пріоритетах Заохочення вільного спілкування всередині проекту Створення базових версій Розбиття стадії на 5 фаз (концепція, планування, розроблення, стабілізація, впровадження) Масштабованість Відсутність жорстких інструкцій

Таблиця 2. Основні принципи RUP, XP та каскадної (waterfall) методологій

RUP	XP	Каскадна (waterfall)
<p>Компонентна архітектура</p> <p>Робота над проектом у згуртованій команді (ключова роль в архітекторів)</p> <p>Рання ідентифікація та неперервне усунення можливих ризиків</p> <p>Концентрація на виконанні вимог замовників</p> <p>Очікування змін у вимогах в процесі розроблення (гнучкість)</p> <p>Постійне забезпечення якості на всіх етапах</p> <p>Мінливість ступеня формалізації (залежно від потреб проекту), за рахунок чого RUP використовується як для невеликих та коротких, так і для великих та складних проектів</p>	<p>Низький ступінь формалізації</p> <p>Малі, спільно розташовані команди розробників (4-10 чоловік)</p> <p>Постійний зв'язок із зацікавленими сторонами проекту</p> <p>«Парне» програмування</p> <p>Короткий цикл зворотнього зв'язку</p> <p>Розроблення через тестування (автоматичне)</p> <p>Неперервний процес, неперервна інтеграція</p> <p>Часті випуски версій</p> <p>Просте та зрозуміле розроблення</p> <p>Колективне володіння кодом</p> <p>Єдиний стандарт кодування</p> <p>Швидке формування плану роботи та його оновлення (по мірі уточнення цілей проекту)</p> <p>Ітеративність (короткі ітерації)</p> <p>Метафора системи</p>	<p>Відсутня можливість внесення змін до вимог протягом життєвого циклу проекту (низька гнучкість)</p> <p>Жорстка послідовність проходження етапів</p> <p>Високий ступінь формалізації</p> <p>Висока зручність для замовника</p> <p>Прозорість процесу розроблення</p> <p>Низькі ризики</p> <p>Ясність процесу розроблення</p> <p>Простота та зручність керування (навіть масштабними проектами)</p> <p>Можливість ранніх оцінок вартості, термінів та ефективності проекту</p>

**Висновки.** Застосування саме коректно підібраної методології розроблення ПЗ дозволяє в типовому циклі розроблення ПЗ усувати розбіжності та відсутність зв'язків між членами команди розробників і кінцевими користувачами., підключати нових людей до проекту на будь-якій стадії розроблення, створювати пакет документації, який надалі спрощує супровід і доповнення програмного продукту, зводити до мінімуму витрати на доопрацювання, модифікацію та супровід ПЗ. Наразі вибір методології розроблення сьогодні залежить від: масштабу проекту; критичності проекту; кількості учасників проекту; ступеня новизни проекту; очікуваної тривалості проекту; вимог замовника; стандартів, прийнятих в компанії-розробника; знань і досвіду проектної команди; уподобань або вимог замовника; складності проекту; стабільності та зрілості процесів у компанії; особистих якостей співробітників. Отже, зараз 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 розробниками та замовниками, зводити до мінімуму витрати на супровід ПЗ.

На основі аналізу фахової літератури було визначено основні принципи методологій розроблення ПЗ. Для подальшої організації процесу ефективного вибору методології розроблення ПЗ (або, швидше, компанії-розробника, яка використовує підходящу методологію) потрібно встановити важливість (вагові коефіцієнти) принципів для кожної методології розроблення ПЗ, які відображають роль та пріоритетність кожного принципу (наприклад, для каскадної методології розроблення ПЗ визначальним є принцип «відсутня можливість внесення змін до вимог протягом життєвого циклу проекту»). 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.

#### ЛИТЕРАТУРА

1. Хаф Л. Методологии разработки программного обеспечения [Электронный ресурс] / Л.Хаф. – Режим доступа: <http://compress.ru/article.aspx?id=11321>
2. Методологии разработки программного обеспечения [Электронный ресурс] – Режим доступа: <http://habrahabr.ru/sandbox/43802/>
3. Говорушенко Т.О. Аналіз процесу вибору технології проектування, методології та середовища розроблення програмного забезпечення / Т.О.Говорушенко, Р.А.Малярчук // Вісник Хмельницького національного університету – 2014. - №6. – С.186-196
4. Коберн А. Быстрая разработка программного обеспечения / А.Коберн – М.: Лори, 2002. – 314 с.