

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Дослідження особливостей покращення процесів розробки
програмного забезпечення SPI для малих
та середніх компаній.

Виконав(ла): студент(ка) 4 курсу, групи СНС-41
спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

(підпис)

Магей М.М.

(прізвище та ініціали)

Керівник

(підпис)

Мацюк О.В.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Литвиненко Я.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Хоміцький Б.В.

(прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
Боднарчук І.О.
(підпис) (прізвище та ініціали)

«21» червня 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

студенту Магей Микола Миколайович
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження особливостей покращення процесів розробки програмного забезпечення SPI для малих та середніх компаній

Керівник роботи к.т.н., доц. Мацюк О.В.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «07» лютого 2023 року № 4/7-133

2. Термін подання студентом завершеної роботи 21 червня 2023 р.

3. Вихідні дані до роботи Літературні джерела з тематики роботи

4. Зміст роботи (перелік питань, які потрібно розробити)

ВСТУП РОЗДІЛ 1. МОДЕЛІ ПОКРАЩЕННЯ ПРОЦЕСІВ РОЗРОБКИ ПЗ ДЛЯ МАЛИХ ТА СЕРЕДНІХ ПІДПРИЄМСТВ

1.1 Методи на основі самодіагностики якості програмного забезпечення 1.2 Модель на основі матриці процесу розробки програмного забезпечення (SPM) РОЗДІЛ 2. МЕТОДИКА ВПРОВАДЖЕННЯ ПРОЦЕСУ ПОКРАЩЕННЯ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В МАЛИХ ТА СЕРЕДНІХ КОМПАНІЯХ 2.1 Модель на основі ASPE-MSC 2.2 PRISMS: Підхід до вдосконалення програмного процесу для малих і середніх підприємств 2.3 Метод MESOPYME 2.4 Цільова метрика анкетуванням спрощена РОЗДІЛ 3. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ 3.1 Охорона праці та її актуальність в IT-сфері 3.2 Шкідлива дія шуму та вібрації і захист від неї ВИСНОВОК ПЕРЕЛІК ПОСИЛАНЬ

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Гурик О.Я., к.т.н., доц.		

7. Дата видачі завдання 24 січня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	24.01.23-27.01.23	<i>Виконано</i>
2.	Підбір джерел по темі роботи	28.01.23 – 01.04.23	<i>Виконано</i>
3.	Оформлення першого розділу	15.04.2023	<i>Виконано</i>
4.	Оформлення другого розділу	30.04.2023	<i>Виконано</i>
5.	Виконання завдання до підрозділу «Безпека життєдіяльності, основи охорони праці»	15.05.2023	<i>Виконано</i>
6.	Оформлення кваліфікаційної роботи	07.06.2023	<i>Виконано</i>
7.	Перевірка на плагіат	07.06.2023	<i>Виконано</i>
8.	Нормоконтроль	09.06.2023	<i>Виконано</i>
9.	Попередній захист кваліфікаційної роботи	11.06.2023	<i>Виконано</i>
10.	Захист кваліфікаційної роботи	22.06.2023	

Студент

(підпис)

Магей М.М.

(прізвище та ініціали)

Керівник роботи

(підпис)

Мацюк О.В.

(прізвище та ініціали)

АНОТАЦІЯ

Дослідження особливостей покращення процесів розробки програмного забезпечення SPI для малих та середніх компаній // Кваліфікаційна робота освітнього рівня "Бакалавр" // Магей Микола Миколайович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНс-41 // Тернопіль, 2023 // с. – 47, рис. – 4, табл. – 1, кресл. – 10, бібліогр. – 17.

Ключові слова: програмне забезпечення, інженерія програмного забезпечення, стандарти розробки ПЗ, зрілість компанії, процеси розробки ПЗ.

Удосконалення (покращення) процесів розробки програмного забезпечення є важливим у сучасному технологічному середовищі, що швидко розвивається. Ось кілька причин, чому це необхідно.

Підвищення якості: удосконалення процесів програмного забезпечення спрямоване на підвищення загальної якості програмних продуктів і послуг. Впроваджуючи надійні процеси, організації можуть виявляти та виправляти дефекти, зменшувати кількість помилок і підвищувати надійність і стабільність своїх програмних рішень. Це, у свою чергу, призводить до підвищення задоволеності клієнтів і кращих результатів бізнесу.

Зменшення витрат: удосконалення програмних процесів може призвести до економії коштів протягом життєвого циклу розробки програмного забезпечення. Завдяки впорядкуванню й оптимізації процесів організації можуть усунути неефективність, скоротити кількість повторних робіт і мінімізувати затримки проекту. Це призводить до зниження витрат на розробку та обслуговування, дозволяючи компаніям ефективніше розподіляти ресурси.

Прискорення виходу на ринок: на сучасному конкурентному ринку швидкість має вирішальне значення. Удосконалення процесу допомагає прискорити час, необхідний для доставки програмних продуктів або оновлень. Застосовуючи гнучкі методології, оптимізуючи процеси розробки та тестування та сприяючи співпраці, організації можуть скоротити цикли розробки та швидко реагувати на мінливі вимоги ринку.

Зменшення ризиків: удосконалення процесів програмного забезпечення включає методи управління ризиками, що гарантує виявлення потенційних ризиків і проблем і їх вирішення на ранніх етапах життєвого циклу розробки. Впроваджуючи надійні процеси забезпечення якості та тестування, організації можуть мінімізувати виникнення критичних помилок або збоїв, які можуть мати значні фінансові чи репутаційні наслідки.

Постійне вдосконалення. Удосконалення процесу — це не одноразове зусилля, а безперервна подорож. Це сприяє розвитку культури навчання, адаптації та інновацій в організаціях. Регулярно переглядаючи й удосконалюючи процеси, організації можуть визначати сфери для вдосконалення, застосовувати нові інструменти й технології та випереджати галузеві тенденції. Це сприяє довгостроковій стабільності та підтримує конкурентоспроможність організацій на ринку.

Відповідність і стандарти: у багатьох галузях є певні вимоги до відповідності та стандарти, яким мають відповідати програмні рішення. Удосконалення процесів допомагає організаціям дотримуватися цих правил і стандартів, гарантуючи, що програмне забезпечення розробляється у відповідний і безпечний спосіб. Це особливо важливо в таких секторах, як охорона здоров'я, фінанси та уряд, де конфіденційність і безпека даних є найважливішими.

Таким чином, удосконалення процесу програмного забезпечення є необхідним для підвищення якості, зниження витрат, прискорення часу виходу на ринок, пом'якшення ризиків, сприяння постійному вдосконаленню та

забезпечення відповідності галузевим стандартам. Інвестуючи в ініціативи з удосконалення процесів, організації можуть оптимізувати свою практику розробки програмного забезпечення та отримати конкурентну перевагу в динамічній і вимогливій індустрії програмного забезпечення.

ANNOTATION

Survey of Particularities of Improving SPI Software Development Processes for Small and Medium -Sized Companies // Qualification work of the educational level "Bachelor" // Mykola Mahey // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, Group CHc-41 // Ternopil, 2023 // p. – 47, fig. – 4, tables – 1, references – 17, posters – 10.

Keywords: software, software engineering, software development standards, company maturity, software development processes.

Software process improvement is essential in today's rapidly evolving technological landscape. Here are some reasons why it is necessary.

Quality Enhancement: Software process improvement aims to enhance the overall quality of software products and services. By implementing robust processes, organizations can identify and rectify defects, reduce errors, and improve the reliability and stability of their software solutions. This, in turn, leads to higher customer satisfaction and better business outcomes.

Cost Reduction: Improving software processes can result in cost savings throughout the software development lifecycle. By streamlining and optimizing processes, organizations can eliminate inefficiencies, reduce rework, and minimize project delays. This leads to lower development and maintenance costs, allowing companies to allocate resources more effectively.

Time-to-Market Acceleration: In today's competitive market, speed is crucial. Process improvement helps accelerate the time it takes to deliver software products or updates. By adopting agile methodologies, optimizing development and testing processes, and promoting collaboration, organizations can shorten development cycles and respond swiftly to changing market demands.

Risk Mitigation: Software process improvement incorporates risk management practices, ensuring that potential risks and issues are identified and addressed early in the development lifecycle. By implementing robust quality assurance and testing processes, organizations can minimize the occurrence of critical errors or failures that could have significant financial or reputational implications.

Continuous Improvement: Process improvement is not a one-time effort but a continuous journey. It fosters a culture of learning, adaptation, and innovation within organizations. By regularly reviewing and refining processes, organizations can identify areas for improvement, adopt new tools and technologies, and stay ahead of industry trends. This promotes long-term sustainability and keeps organizations competitive in the market.

Compliance and Standards: Many industries have specific compliance requirements and standards that software solutions must meet. Process improvement helps organizations adhere to these regulations and standards, ensuring the software is developed in a compliant and secure manner. This is particularly important in sectors such as healthcare, finance, and government, where data privacy and security are paramount.

In summary, software process improvement is necessary to enhance quality, reduce costs, accelerate time-to-market, mitigate risks, promote continuous improvement, and ensure compliance with industry standards. By investing in process improvement initiatives, organizations can optimize their software development practices and gain a competitive edge in the dynamic and demanding software industry.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. МОДЕЛІ ПОКРАЩЕННЯ ПРОЦЕСІВ РОЗРОБКИ ПЗ ДЛЯ МАЛИХ ТА СЕРЕДНІХ ПІДПРИЄМСТВ.....	12
1.1 Методи на основі самодіагностики якості програмного забезпечення	12
1.1.1 Розширена анкета зрілості (EMQ).....	13
1.1.2 Матриця цілей, діяльності та відповідальності (GAR)	14
1.1.3 Направлена анкета	16
1.1.4 Оцінка результату самодіагностики	16
1.2 Модель на основі матриці процесу розробки програмного забезпечення (SPM)	16
РОЗДІЛ 2. МЕТОДИКА ВПРОВАДЖЕННЯ ПРОЦЕСУ ПОКРАЩЕННЯ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В МАЛИХ ТА СЕРЕДНІХ КОМПАНІЯХ	22
2.1 Модель на основі ASPE-MSC	22
2.2 PRISMS : Підхід до вдосконалення програмного процесу для малих і середніх підприємств.....	27
2.3 Метод MESOPYME	29
2.4 Цільова метрика анкетуванням спрощена.....	30
РОЗДІЛ 3. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	33
3.1 Охорона праці та її актуальність в ІТ-сфері.....	33
3.2 Шкідлива дія шуму та вібрації і захист від неї.....	37
ВИСНОВОК.....	43
ПЕРЕЛІК ПОСИЛАНЬ	45

ВСТУП

Для покращення якості продукції розроблено моделі та стандарти якості. Якість програмного забезпечення визначається як усі характеристики продукту, які впливають на його здатність задовольняти явні та неявні потреби/вимоги користувача (ISO/IES 9126). Таким чином, якість програмного забезпечення визначається як відповідність чітко задокументованим стандартам розробки та неявним характеристикам, які очікуються від усього професійно розробленого програмного забезпечення. Отже, це визначення передбачає три вимоги до забезпечення якості, яким повинен відповідати розробник [1]:

- Спеціальні формальні вимоги, які стосуються в основному результатів програмної системи.
- Стандарти якості програмного забезпечення, зазначені в контракті.
- Належна практика розробки програмного забезпечення (GSEP), що відображає найсучасніші професійні практики, яких має дотримуватися розробник, навіть якщо це прямо не зазначено в контракті.

Те, як ми розробляємо програмне забезпечення, впливає на його якість, а отже, процес розробки програмного забезпечення є одним із найважливіших факторів у визначенні якості розроблюваного продукту.

Програмний процес – це набір дій разом із обмеженнями впорядкування серед них, таким чином, якщо дії виконуються належним чином і відповідно до обмежень упорядкування, буде отримано бажаний результат. Бажаним результатом є високоякісне програмне забезпечення за низьку вартість. Оскільки кожен проект розробки програмного забезпечення є екземпляром процесу, за яким він слідує, це, по суті, процес, який визначає очікувані результати проекту [2]. Удосконалення процесу автоматично призводить до підвищення якості продукту (програмного забезпечення).

Сьогодні лише деякі організації, що займаються програмним забезпеченням, досягають високого рівня якості процесу розробки. Значна

кількість програмного забезпечення виробляється в усьому світі малими та середніми підприємствами (МСП) із кількістю працівників від 1 до приблизно 50. Існує потреба в малих компаніях програмного забезпечення в багатьох країнах і для вдосконалення процесу програмного забезпечення.

Термін «мала установа» визначається як організація чи компанія, що складається з менше ніж приблизно 100 осіб і проект із менш ніж приблизно 20 осіб [3]. Як зазначено на веб-сайті Інституту розробки програмного забезпечення (SEI) для невеликих налаштувань, основним аспектом, який слід враховувати в цих середовищах, є те, що кількість ресурсів, які використовуються для підтримки зусиль щодо вдосконалення процесу, становитиме великий відсоток операційного бюджету організації.

Переважає більшість виробників програмного забезпечення, які ще не впровадили методологію вдосконалення процесу програмного забезпечення, сплачують високі витрати на виробництво та обслуговування систем, і тому витісняються з глобального ринку, не перебуваючи на такому ж рівні конкурентоспроможності, як компанії, які володіють методами покращення процесів розробки ПЗ.

Існує декілька моделей удосконалення процесу програмного забезпечення, таких як інтеграція моделі зрілості можливостей (CMMI), вдосконалення процесу програмного забезпечення та визначення можливостей (SPICE) і норми ISO 9000 від Міжнародної організації стандартизації (ISO). Ці моделі забезпечують шаблони якості, які компанія повинна впроваджувати для покращення процесу розробки програмного забезпечення. На жаль, успішна реалізація таких моделей, як правило, неможлива в контексті малих і середніх організацій програмного забезпечення, оскільки вони не здатні нести витрати на впровадження цих програм вдосконалення процесів програмного забезпечення.

Належне впровадження методів розробки програмного забезпечення є складним завданням для невеликих організацій, оскільки вони часто працюють з обмеженими ресурсами та суворими часовими обмеженнями. Культурні

проблеми, такі як опір змінам з боку співробітників або керівництва, які вважають додаткову роботу, необхідну для забезпечення якості, марним і складним тягарем, покладеним на команду, що розвивається. Національна культура також впливає на методи вдосконалення процесу. Через бюджетні обмеження послуги консалтингової організації для покращення якості програмного забезпечення неможливі; Проте потреба в забезпечення якості для розроблюваної програми стає все більш очевидною, і менеджери прагнуть її досягти через впровадження міжнародних стандартів якості, які в довгостроковій перспективі призводять до зниження собівартості виробництва.

У наступних розділах роботи обговорюються деякі моделі вдосконалення процесів програмного забезпечення для малих і середніх організацій. Пізніше ці моделі порівняємо. Зрештою в роботі пропонуються напрямки майбутніх досліджень у цій галузі.

РОЗДІЛ 1. МОДЕЛІ ПОКРАЩЕННЯ ПРОЦЕСІВ РОЗРОБКИ ПЗ ДЛЯ МАЛИХ ТА СЕРЕДНІХ ПІДПРИЄМСТВ

Щоб отримати перевагу в постійно зростаючому висококонкурентному світі розробки програмного забезпечення, для організації важливо регулярно контролювати процес розробки програмного забезпечення. Для організації важливо постійно вдосконалювати цей процес на основі відгуків від різних зацікавлених сторін. Деякі моделі вдосконалення процесів програмного забезпечення для малих і середніх організацій обговорюються в цьому розділі.

1.1 Методи на основі самодіагностики якості програмного забезпечення

Першим кроком до реалізації будь-якої стратегії вдосконалення процесу розробки програмного забезпечення є оцінка поточного рівня якості процесу та пошук потенційних областей покращення. Ця методологія для самодіагностики базується на концепціях, цілях і діях, визначених моделлю зрілості можливостей (СММ), яка може використовуватися малою або мікроорганізацією, як частина плану внутрішнього аудиту до офіційної оцінки. Для малої чи середньої організації важко оцінити свої поточні можливості за допомогою методу оцінки SCAMPI A (єдиного методу в наборі продуктів СММІ, який може призвести до рейтингу), оскільки це займає більше часу та споживає більше ресурсів.

Наймати професійних оцінювачів обходиться дорого, а співробітникам компанії доводиться витратити час на ці оцінки поза роботою, що дорого коштує компанії. Ця методологія може дати простий діагноз або допомогти створити план дій. Вона намагалася адаптувати СММ для невеликих організацій шляхом додавання деяких додаткових анкет, щоб допомогти організаціям зрозуміти свій поточний статус і знайти сфери для вдосконалення. За допомогою цих анкет малі та середні організації зможуть використовувати СММ. Основна мета цієї

методології полягає в зборі фактів, які допомагають зробити «моментальний знімок» поточних процесів, що здійснюються організацією, щоб можна було це зрозуміти, а потім мета полягає в тому, щоб визначити сильні сторони та сфери вдосконалення, таким чином визначаючи ступінь завершеності кожної з КРА (ключових областей процесу) моделі СММ. Щоб зібрати цю інформацію, пов'язану з поточними процесами організації, дослідники створили три анкети [4]:

1. Розширена анкета зрілості (EMQ).
2. Матриця цілей, діяльності та відповідальності (GAR).
3. Спрямована анкета.

СММ описує дії та практики, необхідні для досягнення рівня зрілості в рамках процесів розробки програмного забезпечення. Компанія, яка бажає вдосконалити свої процеси за допомогою СММ, повинна досягати згаданих рівнів поступово, безперервно та по зростанню; тобто нижні рівні є основою для верхніх рівнів. Першим кроком є оцінка рівня зрілості організації, і для цього офіційні анкети розроблені як список неієрархічних, бінарних запитань, щоб визначити рівень зрілості компанії було просто. Коли відомий поточний рівень зрілості, ця модель допомагає визначити ключові практики, необхідні для підвищення зрілості процесів.

1.1.1 Розширена анкета зрілості (EMQ)

EMQ базується на анкеті зрілості, розробленій SEI. Ця анкета містить кілька запитань про конкретні КРА, на які можна відповісти «так», «ні», «не знаю» та «не застосовано». Усі запитання стосуються цілей та зобов'язань кожного КРА, але не стосуються конкретних заходів, які мають бути виконані для досягнення цих цілей. Пряме застосування опитувальника зрілості в малих і мікроорганізаціях призводить до того, що більшість запитань мають відповіді «ні». Основна причина такої тенденції полягає в тому, що багато із

запропонованих в анкеті цілей досягаються лише частково. Тому дослідники модифікували EMQ, дозволяючи відповіді «так», «ні» та «частково» (для опису невиконаних цілей). Мета EMQ полягає в тому, щоб керувати системним адміністратором невеликої організації, щоб він чи вона могли відповідати на запитання у спосіб, який точніше відображає статус організації. За допомогою цієї анкети адміністратор може визначити цілі та зобов'язання, які виконуються лише частково, щоб вони могли бути включені до плану вдосконалення керівником аудиту, який відповідає за процес самодіагностики.

1.1.2 Матриця цілей, діяльності та відповідальності (GAR)

Успіх моделі на основі CMM залежить від повного досягнення певних цілей і зобов'язань для кожного КРА. Ці цілі включають вимірювання, документування, перевірку та перевірку діяльності для забезпечення якості програмного забезпечення. Між цілями, видами діяльності та здібностями існує тісний зв'язок, який не так одразу видно з опису стандарту CMM.

На сьогоднішній день не існує спеціального стандарту ISO для моделі зрілості можливостей (CMM). CMM – це структура вдосконалення процесів, розроблена Інститутом розробки програмного забезпечення (SEI) при Університеті Карнегі-Меллона. Він надає набір найкращих практик для організацій щодо вдосконалення процесів розробки програмного забезпечення.

Однак модель CMM вплинула на розробку різних міжнародних стандартів. Наприклад, ISO/IEC 15504, також відомий як SPICE (покращення процесів програмного забезпечення та визначення можливостей), є стандартом, який визначає структуру для оцінки та вдосконалення процесів розробки програмного забезпечення. Хоча ISO/IEC 15504 не базується безпосередньо на CMM, ISO/IEC 15504 має спільні риси з CMM і забезпечує подібний підхід для оцінки можливостей процесу.

ISO/IEC 15504 є частиною стандарту ISO/IEC 12207, який визначає процеси життєвого циклу розробки програмного забезпечення. Хоча ці стандарти не стосуються CMM, їх можна використовувати в поєднанні з CMM або іншими моделями вдосконалення процесів для оцінки та вдосконалення можливостей розробки програмного забезпечення в організації.

Отже, очевидно, що засвоєння та інтерпретація такої кількості інформації є тривалим і складним процесом. Щоб полегшити завдання адміністраторів програмного забезпечення, пропонується матриця. Ця матриця включає зв'язок між здібностями (змінними), видами діяльності (практиками, пов'язаними з кожним КРА), цілями та зобов'язаннями (цілі, яких потрібно досягти в кожному КРА), а також відповідальними особами (клієнтом, аналітиком вимог, програмним забезпеченням). інженерна група, менеджер, група забезпечення якості) для кожного КРА.

Коли ця матриця використовується як засіб контролю протягом життєвого циклу проекту, вона працює не тільки як діагностика якості проекту, а й як керівництво до дій, яке допомагає керівнику проекту виправляти недоліки та досягати поставлених цілей. КРА, таким чином, покращує загальний процес.

Матрицю GAR можна автоматизувати за допомогою експертної системи шляхом перетворення зв'язків між правилами, відповідальною особою (зацікавленою особою) та діяльністю в правила висновків, які можуть бути викликані базою знань, що складається з відповідей на запитання всередині матриці. Експертна система може допомогти внутрішньому або зовнішньому аудитору в оцінці зрілості процесів розробки шляхом визначення ступеня задоволеності кожного КРА відповідно до CMM. Таким чином, маючи базу знань і заповнивши анкету, ми могли отримати оцінку стану організації без зайвих зусиль, крім документального оформлення діяльності. Оригінальний CMM представляється як текст. GAR забезпечує візуальне представлення, яке спрощує розуміння величезної кількості інформації, необхідної для оцінки цілей.

1.1.3 Направлена анкета

Останнім форматом методології самодіагностики є направлена анкета, за допомогою якої провідний аудитор може побудувати базу знань. Ця анкета має сутність оригінальної анкети зрілості від СММ, але в цьому випадку кожне нове запитання генерується на основі відповіді на попередні запитання, тому нове запитання може бути спрямоване на доповнення інформації, отриманої раніше, або на підтвердження такої інформації. Звідси і назва анкети – направлена. У будь-якому випадку марні питання відкидаються.

1.1.4 Оцінка результату самодіагностики

Результати, отримані за допомогою опитувальників, дають відповідь на основне запитання: чи досягнуто ключові технологічні зони, необхідні СММ для певного рівня? Для кожного КРА є чотири можливі відповіді: КРА повністю досягнуто, частково досягнуто, не досягнуто або воно не застосовується. Частково досягнуті або недосягнуті КРА є сферами можливостей для покращення, і це має бути частиною плану дій.

1.2 Модель на основі матриці процесу розробки програмного забезпечення (SPM)

SPM надає організації ранжований список дій, які можна ввести в стратегію вдосконалення процесу розробки програмного забезпечення. SPM базується на розгортанні функції якості (QFD). Ця модель допомагає організації визначити відносну важливість програмних процесів. Для процесів з високим пріоритетом практики, над якими потрібно працювати, визначаються матрицею програмного процесу. Модель SPM використовує самооцінку всередині організацій і демонструє такі характеристики [5]:

1. Відноситься до бізнес-цілей компанії.
2. Зосереджено на найважливішому програмному процесі..
3. Забезпечує максимальне співвідношення ціни та якості
4. Пропонує вдосконалення, які мають максимальний ефект за якомога короткий час.
5. Забезпечує швидке повернення інвестицій (ROI).
6. Орієнтований на процес.
7. Відноситься до інших моделей програмного забезпечення.
8. Є гнучким і простим у використанні.

Використання моделі QFD в області інженерії програмного забезпечення для проектування архітектури ПЗ використовували також в свої роботах вчені з ТНТУ [6, 7, 8, 9].

SPM можна використовувати для створення стратегії вдосконалення на основі QFD. QFD – це спосіб гарантувати якість дизайну, поки продукт все ще знаходиться на стадії проектування, а також як «система якості, орієнтована на надання продуктів і послуг, які задовольняють клієнтів».

Щоби фіксувати і моторити думку клієнта протягом життєвого циклу виробництва, QFD використовує низку матриць, які перетворюють твердження клієнта в кінцевий продукт. Для використання доступні різні моделі, і, згідно з [10], модель, адаптована Американським інститутом стандартів (чотирьохфазна модель) і містить чотири матриці, ймовірно, є найбільш широко описаною та використовуваною моделлю в Сполучених Штатах. SPM базується на першій матриці цієї моделі, будинку якості.

Спочатку збирається «голос клієнта» та вимірюється відносна важливість кожної вимоги клієнта. У матриці якості ці вимоги використовуються для визначення конструктивних характеристик, які мають найбільший вплив на вимоги замовника. Хоча QFD складається з багатьох матриць, основна увага часто приділяється цій матриці, оскільки її використання окремо може мати

значний вплив на процес розробки продукту. Матриця зазвичай розбивається на шість областей:

- вимоги замовника (WHAT/ЩО);
- характеристики конструкції (HOW/ЯК);
- загальна важливість вимог замовника;
- взаємозв'язок між вимогами замовника та конструктивними характеристиками;
- важливість конструктивних характеристик;
- взаємозв'язки між характеристиками конструкції.

Використовуючи QFD, модель програмного процесу розглядається як клієнт, де програмні процеси є вимогами клієнта. Ці процеси були визначені з літератури про процеси розробки програмного забезпечення. Приклади процесів:

1. Визначення та документування процесів.
2. Систематична оцінка придатності програмного процесу постачальників.
3. Системне впровадження проектування програмного забезпечення.
4. Систематичне планування робочого процесу проекту та кошторису.

Характеристики проектування – це практики, яких необхідно дотримуватися, щоб процеси були успішними. Ці практики також були визначені в літературі про процеси програмного забезпечення. Приклади практик:

- Тестування роботи клієнта перед впровадженням програмного забезпечення.
- Прототип або імітація критичних елементів програмного забезпечення.
- Ведення та відстеження історії продуктів.

Вирішальною частиною розробки матриці процесу розробки програмного забезпечення було визначення зв'язків між процесами та практиками. Ті, які прямо згадуються в літературі, були легко ідентифіковані. Використовуючи висновки експертів і різні статистичні методи, було виявлено інші взаємозв'язки

між процесами та практиками, що призвело до розробки та перевірки матриці процесу розробки програмного забезпечення, яка потім була перевірена в галузі.

Щоб невелика компанія використовувала будь-яку модель процесу розробки програмного забезпечення з користю, необхідно, щоб зусилля були мінімальними. SPM надає йому загальний розділ, який був заповнений раніше та може використовуватися в компанії. Надається анкета для оцінки поточної ефективності, запланованої майбутньої ефективності та важливості для компанії кожного процесу. З точки зору компанії, все, що їм потрібно надати, це вимірювання для розрахунку загальної важливості програмного процесу, враховуючи наступне [5]:

1. Поточні можливості, оцінені за допомогою анкети для самооцінки.
2. Майбутні можливості як дані керівництва.
3. Важливість програмного процесу для бізнесу.
4. Конкурентний аналіз/
5. Використовуйте ринок для конкретних вимог компанії, наприклад, сертифікація ISO.

Дозвіл керівництву вибирати, включати цифри для аналізу конкуренції та впливу на ринок, забезпечує гнучкість моделі. Використовуючи стандартні розрахунки QFD і посилаючись на таблицю 1.1, яка містить конкретні дані компанії, важливість процесу (D) розраховується згідно формули 1.1.

$$D = [1 + (B - A) * 0,2] * C. \quad (1.1)$$

У прикладі, наведеному в таблиці 1.1, можна побачити, що вони впорядковані від найбільш важливого до найменш важливого:

1. Розробка робочого проекту.
2. Розробка системних вимог і дизайну.
3. Приймальні випробування систем.

Використовуючи значення взаємозв'язків між процесами та практиками, важливість кожної практики розраховується як у формулі (1.2):

$$w = \sum I_i * R_i, \quad (1.2)$$

де I_i – важливість процесу;

R_i – вага взаємозв'язку.

У випадку тестових даних про «запити користувачів, що визначають вимоги», будемо мати:

$$(3,6 * 9) + (7,0 * 9) + (2 * 0) = 95,4.$$

Це надає наступний пріоритетний список практик:

1. Запити користувачів визначають вимоги.
2. Визначити інтерфейси програмної системи.
3. Розробка і документування верифікації одиниць

Практики з найвищими цінностями є найважливішими, і тому пропонується, щоб над ними в першу чергу попрацювали в організації.

Таблиця 1.1 – Дані для розрахунку важливості характеристик якості програмної системи

Практики процесу	Запити користувачів визначають вимоги	Визначити інтерфейси програмної системи	Розробити та документувати одиничну сертифікацію	Поточні досягнення (1 – 5)	Майбутнє досягнення (1 – 5)	Поліпшення до компанії (1 – 5)	Важливість процесу (1 – 5)
				A	B	C	D
Розробка системних вимог і дизайну	•	•		2	3	3	3,6
Розробка робочого проекту	•		°	2	4	5	7
Ємність системи _ тестування		∇	∇	3	3	2	2
Важливість практики	95,4	34,4	14,6				

З цього встановлюються пріоритети, які необхідно включити до будь-якого плану дій з удосконалення процесу розробки програмного забезпечення, і вони можуть допомогти організації визначити свою стратегію вдосконалення. Повний SPM надає організації ранжований список дій, які можна ввести в стратегію вдосконалення програмного процесу. Цей ранжований список можна об'єднати з показниками витрат і ефективними розрахунками за часом, таким чином враховуючи ці фактори при визначенні плану дій для організації.

РОЗДІЛ 2. МЕТОДИКА ВПРОВАДЖЕННЯ ПРОЦЕСУ ПОКРАЩЕННЯ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В МАЛИХ ТА СЕРЕДНІХ КОМПАНІЯХ

2.1 Модель на основі ASPE-MSС

Процес впровадження процесу покращення розробки програмного забезпечення в малих та середніх компаніях (ASPE-MSС) визначається шляхом інтеграції та адаптації існуючих підходів до характеристик малих програмних компаній. Як показано на рисунку 2.1, основними етапами підходу є діагностика, стратегічний аналіз, визначення та реалізація, які можуть виконуватися ітеративним та поступовим способом, щоб крок за кроком встановити один або більше процесів у межах організації.

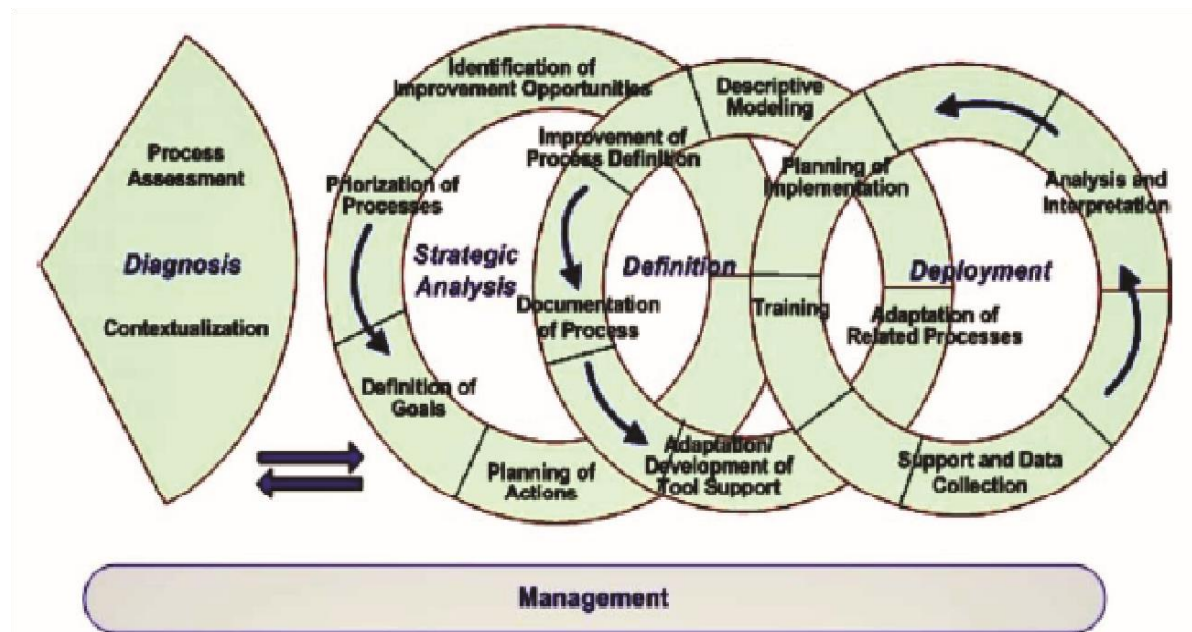


Рисунок 2.1 – Модель підходу на основі ASPE-MSС

Крім того, цей підхід також охоплює управління створенням програмного процесу (процесів), включаючи планування, моніторинг, контроль і аналіз після виводу з експлуатації програмного забезпечення. Встановлення процесу в контексті невеликих компаній виконується інженером-технологом, як правило,

зовнішнім консультантом, якщо компанія не має компетенції в цій галузі. У цьому випадку один співробітник організації повинен бути призначений (принаймні неповний робочий день) помічником інженера-технолога, щоб забезпечити передачу знань і полегшити встановлення процесу. Іншими учасниками є спонсор, як правило, один із директорів компанії, і всі виконавці процесу, які беруть участь у відповідному процесі(ах).

Основними етапами підходу, відповідно до [11], є наступні етапи.

Планування. На початку налагодження процесу планується на високому рівні. Пізніше, під час стратегічного аналізу, план переглядається, доповнюється та адаптується відповідно до прийнятих рішень. Під час планування всі залучені сторони інформуються та мотивуються, і необхідно отримати зобов'язання. При необхідності асистента інженера-технолога навчають основним аспектам створення та вдосконалення програмного процесу.

Етап 1: Діагностика . Метою цього етапу є контекстуалізація організації та отримання високорівневого знімка фактичного процесу розробки програмного забезпечення. Таку базову лінію можна встановити за допомогою оцінки процесу розробки програмного забезпечення, використовуючи, наприклад, MARES, метод оцінки процесу, сумісний із ISO/IEC 15504, розроблений для малих компаній. Результати оцінювання, що мають відношення до встановлення процесу, можуть включати:

- контекстний опис організації;
- опис високого рівня фактичного програмного процесу із зазначенням того, які процеси виконуються та як;
- цільові профілі можливостей процесів із зазначенням високопріоритетних процесів і рівня їх можливостей, якого необхідно досягти, щоб досягти бізнес-цілей організації та вдосконалення;
- спостереження та/або оцінені профілі можливостей процесів, а також сильні та слабкі сторони ідентифікованого процесу(ів) програмного

забезпечення, включаючи вказівку на потенційні ризики та можливості покращення.

Етап 2: Стратегічний аналіз. Мета цього етапу полягає в тому, щоб визначити обсяг і визначити пріоритетність процесів-кандидатів, які мають бути встановлені на основі результатів діагностики та відповідно до бізнес-цілей організації та вдосконалення.

Це можна зробити, використовуючи, наприклад, адаптацію техніки аналізу SWOT (сильні сторони/слабкі сторони/можливості/загрози), що пов'язує важливість процесів та їхню оцінку/розраховану здатність. На основі прийнятих рішень визначаються вимірювані цілі, які мають бути досягнуті шляхом встановлення вибраного(их) процесу(ів) відповідно до цілей бізнесу та покращення організації.

Дії плануються, переглядаючи та доопрацьовуючи план. Це також включає визначення заходів, необхідних для моніторингу прогресу відповідно до визначених цілей. Стратегічний аналіз зазвичай виконується директором(ами) компанії під керівництвом інженера-технолога та помічника. У кожному ітераційному циклі рішення повинні переглядатися. Якщо будь-яка інформація або артефакти, які будуть створені на етапах 1 і 2, уже доступні, результати, можливо, потрібно просто переглянути та, якщо необхідно, завершити або оновити.

Етап 3: Означення. Метою цього етапу є визначення вибраного(их) програмного(их) процесу(ів) у формі посібника з процесу для підтримки виконавців. Як правило, визначення вибраного(их) процесу(ів) починається з описового моделювання фактичного(их) процесу(ів) на місці. Ця діяльність складається з фази ознайомлення з процесом і фази детального виявлення. Під час фази ознайомлення з процесом отримується огляд процесу розробки програмного забезпечення та його загальної структури, взаємодії та послідовності, який документується, наприклад, на схемі процесу. На наступному кроці визначаються ролі, компетенції та відповідальність, пов'язані

з кожною діяльністю. Визначаються пов'язані робочі продукти та виявляються шаблони для цих артефактів.

Мета процесу визначається у кількісних термінах, що визначають заходи відповідно до цілей якості та управління, а також цілей, яких має досягти організація процесу. Це також включає визначення характеристик продуктивності для процесу та пов'язаних з ним робочих продуктів. Початкове представлення процесу потім завершується під час виявлення деталей процесу, таких як критерії входу/виходу, використані методи та інструменти, рекомендації щодо адаптації, приклади, попередження щодо потенційних проблем, необхідна інфраструктура, контрольні списки тощо. Це робить інженер проекту та асистент у тісній співпраці з усіма виконавцями процесу, щоб зафіксувати їх розуміння та ноу-хау щодо процесу шляхом інтерв'ю, спостережень або семінарів і дослідження існуючих артефактів. Після визначення початкової версії процесу як є, представлення процесу може бути проаналізовано, реструктуровано, завершено або адаптовано з метою покращення процесу відповідно до цілей організації [12].

Альтернативні рішення можна визначити з урахуванням найкращих практик, стандартів або еталонних моделей. Аналізуючи різні аспекти, можна вибрати відповідні альтернативи та інтегрувати їх у представлення процесу. У результаті цього етапу процес представляється у формі посібника з процесу, який потім переглядається виконавцями процесу. Після визначення процесу інструменти, що використовуються під час виконання процесу, повинні бути адаптовані або розроблені відповідно до визначення процесу. Крім того, пов'язані процеси, такі як процеси управління, мають бути переглянуті та, якщо необхідно, адаптовані. Цю фазу, можливо, доведеться повторити кілька разів, поки не буде визначено та/або повторено задовільне представлення процесу, щоб постійно адаптувати та покращувати представлення процесу.

Етап 4: Реалізація. По-перше, оцінка визначеного(их) процесу(ів) планується паралельно з їх впровадженням. Це включає перегляд та/або

визначення заходів для моніторингу та визначення ефективності та придатності процесу(ів), а також того, чи досягнуто очікуваних переваг.

План реалізації також переглядається та уточнюється. Необхідні ресурси, інформація та інфраструктура доступні та розподілені. Виконавці процесу мотивовані та навчені стосовно процесу (процесів), який(-и) буде впроваджено. На початку інженер-технолог і асистент уважно стежать за впровадженням, надають підтримку та збирають дані відповідно до запланованої оцінки. Зібрані дані та досвід аналізуються та інтерпретуються. На основі цих даних можна покращити представлення процесу або запустити нові цикли, що повторюють фазу діагностики та/або стратегічний аналіз.

Отриманий досвід щодо того, як визначення процесу адаптується до конкретних проектів, також може слугувати основою для постійної еволюції інструкцій з адаптації. У контексті невеликих компаній ми чітко не відокремлюємо пілотне впровадження від розгортання в масштабах організації, оскільки пілотування в більшості випадків через невеликий розмір вже відбувається в усій компанії. Єдина відмінність тут полягає в сильнішій зосередженості на допомозі та зборі даних на початку реалізації процесу(ів), який постійно перетворюється на менш інтенсивний регулярний моніторинг та обслуговування.

Моніторинг і контроль: Повне встановлення процесу (процесів) контролюється та управляється. Тому дані збираються та аналізуються інженером-технологом і помічником. За необхідності розпочинаються коригувальні дії та оновлюється план.

Прикінцеві дії: після завершення повного циклу встановлення процесу підхід до встановлення процесу оцінюється як основа для постійного вдосконалення. Це робиться шляхом збору та аналізу відгуків від виконавців процесу, спонсора, інженера-технолога та помічника на зустрічі для зворотного зв'язку або за допомогою анкет.

2.2 PRISMS : Підхід до вдосконалення програмного процесу для малих і середніх підприємств

PRISMS – це дослідницький проект, у якому команда з трьох дослідників працює разом із менеджерами та розробниками компаній-учасниць, консультуючи та допомагаючи з плануванням і впровадженням програм удосконалення процесів розробки програмного забезпечення протягом декількох років. Модель процесу PRISMS узагальнено на рисунку 2.2.

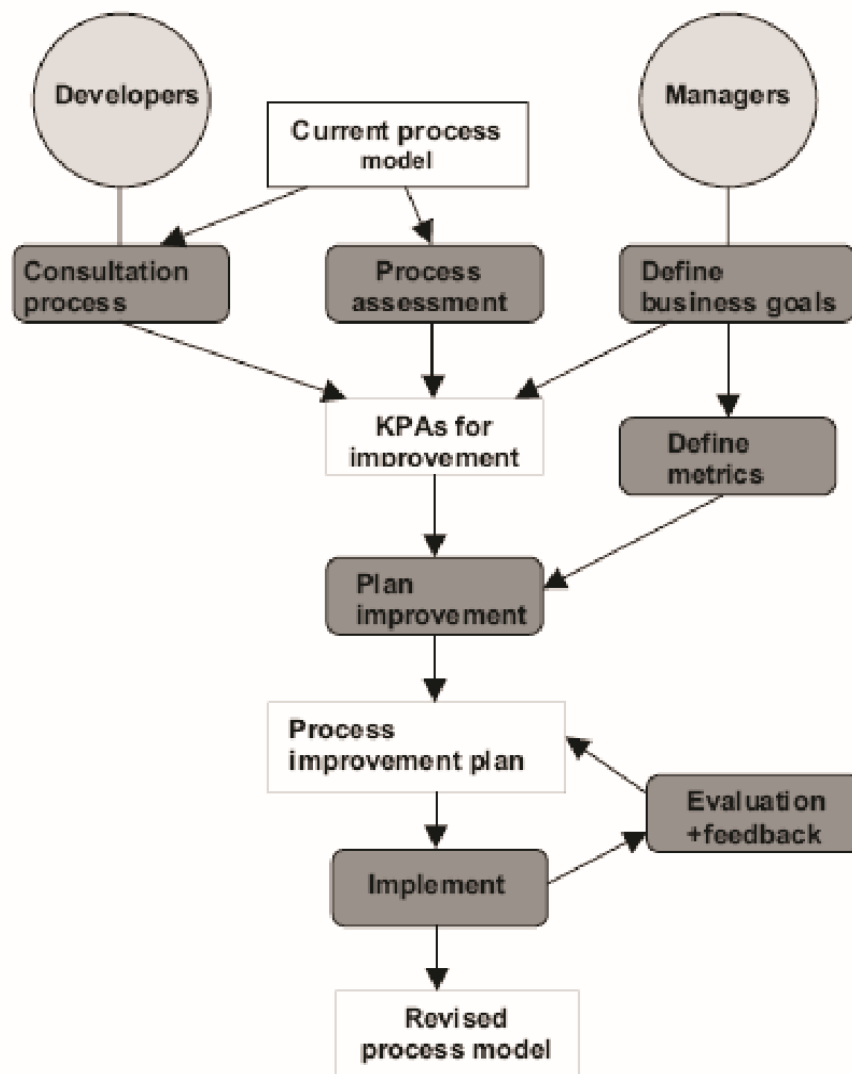


Рисунок 2.2 – Модель підходу до покращення процесів на основі PRISMS

Основними характеристиками процесу, згідно [13], є наступні елементи.

- Існуючий процес, незважаючи на неофіційне визначення, досліджується, і, якщо дозволяють ресурси, створюється явна модель. Це часто призводить до гарячої дискусії, оскільки виявляються різні інтерпретації існуючої специфікації процесу.

- На початку програми PRISMS бізнес-цілі визначаються керівництвом. Ці цілі керують більшою частиною подальшої діяльності, особливо вибором і пріоритезацією ключових областей процесу для вдосконалення та вибором вимірювань.

- Проводиться консультація із залученням усіх членів команд розробки. Це корисна вправа, яка використовує сильні сторони невеликих, гнучких команд, які є в невеликих організаціях. Сеанс мозкового штурму та/або опитування на основі анкети допомагають команді розробників взяти на себе відповідальність за програму SPI та брати участь у програмі з самого раннього етапу.

- Спеціалізована версія оцінки СММ проводиться членами дослідницької групи, перш за все, щоб допомогти визначити ключові області процесу, які необхідно вдосконалити. Це також вказує на рівень СММ процесу розробки програмного забезпечення, який часто є менш корисним для МСП, але все ж корисний як базова лінія для вимірювання майбутнього прогресу. Це узгоджується зі спостереженням, що рівень зрілості має бути мірою покращення, а не цілями для покращення.

- Використовуючи ці вхідні дані, КРА для вдосконалення визначаються та визначаються пріоритети. Головним критерієм тут має бути ступінь сприяння КРА у досягненні визначених бізнес-цілей. Одна компанія виявила корисним підхід зваженого відбору. Також може бути використаний матричний підхід процесу/практики.

- Вимірювання визначаються як невід'ємна частина процесу планування SPI. Керівники, як правило, прагнуть мати точніші способи відстеження ключових ресурсів і показників якості. Парадигма метрики

цільового питання може бути використана для вимірювання вибраних атрибутів на основі бізнес-цілей, визначених для програми SPI.

- План SPI періодично переглядається, і впроваджуються механізми збору відгуків від зацікавлених сторін.

2.3 Метод MESOPYME

MESOPYME – це метод постійного вдосконалення процесу розробки програмного забезпечення [14]. Він орієнтований на МСП і надає робочий посібник із впровадження вдосконалень.

Схема процесу показана на рисунку 2.3.

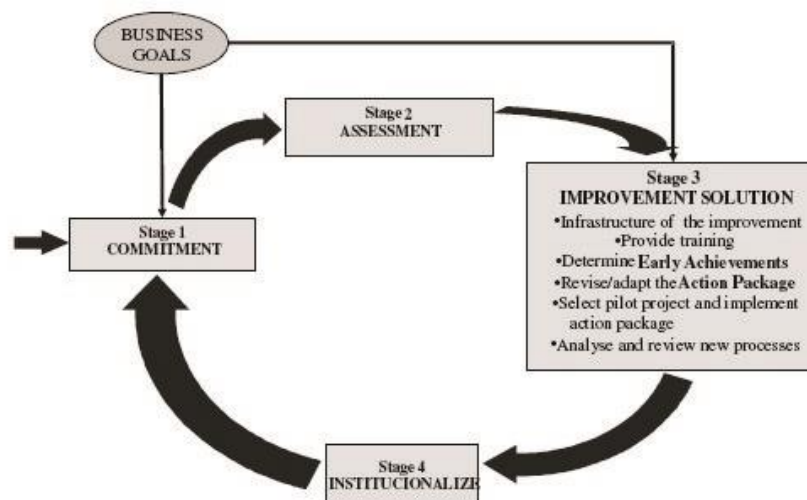


Рисунок 2.3 – Процес MESOPYME

MESOPYME було визначено з урахуванням загальної моделі SPI, визначеної з чотирма етапами.

- Етап 1: Зобов'язання вдосконалюватися. Його метою є отримання підтримки вищого керівництва для реалізації проекту вдосконалення.
- Етап 2: Оцінка процесу розробки програмного забезпечення. Його мета полягає в тому, щоб отримати сильні та слабкі сторони процесу, оцінені

щодо моделі процесу розробки програмного забезпечення – СММ. З цієї оцінки вибираються процеси (зазвичай від одного до трьох), які потрібно покращити.

- Етап 3: рішення для покращення. Його мета – забезпечити необхідну інфраструктуру для вдосконалення (у вибраних процесах) і створити план, якому слід слідувати, щоб визначити та впровадити вдосконалення в цих вибраних процесах. Етап покращення рішення виконується шляхом застосування загального набору компонентів, який ми назвали пакетом дій.

Пакет дій – це загальне рішення для певної області процесу розробки програмного забезпечення, яке має бути налаштовано під компанію з урахуванням її бізнес-цілей та результатів оцінки. У деяких пілотних проектах реалізовано комплекс заходів.

- Етап 4: Інституціоналізація. Нарешті, вдосконалення має бути інституційним.

2.4 Цільова метрика анкетуванням спрощена

Основна ідея цієї моделі полягає в тому, що «вимірювання програмного забезпечення є необхідністю для контролю проектів програмного забезпечення та покращення якості». Ця модель базується на Goal Questionary Metric – GQM. «GQM Lightweight (спрощена)» – це налаштована модель GQM, яка поєднує в собі існуючу модель, адаптовану до конкретних характеристик і обмежень невеликих програмних компаній.

Метрична парадигма цільового питання була запропонована як цілеспрямований підхід для вимірювання продуктів і процесів у розробці програмного забезпечення. Це механізм для визначення та оцінки набору операційних цілей за допомогою вимірювання. GQM представляє системний підхід до адаптації та інтеграції цілей із моделями програмних процесів, продуктів і перспектив якості, що цікавлять, на основі конкретних потреб проекту та організації.

Ця модель є результатом адаптації існуючих моделей процесу GQM до специфічних характеристик і обмежень невеликих компаній, що займаються програмним забезпеченням. Схематично дана модель показана на рисунку 2.4.

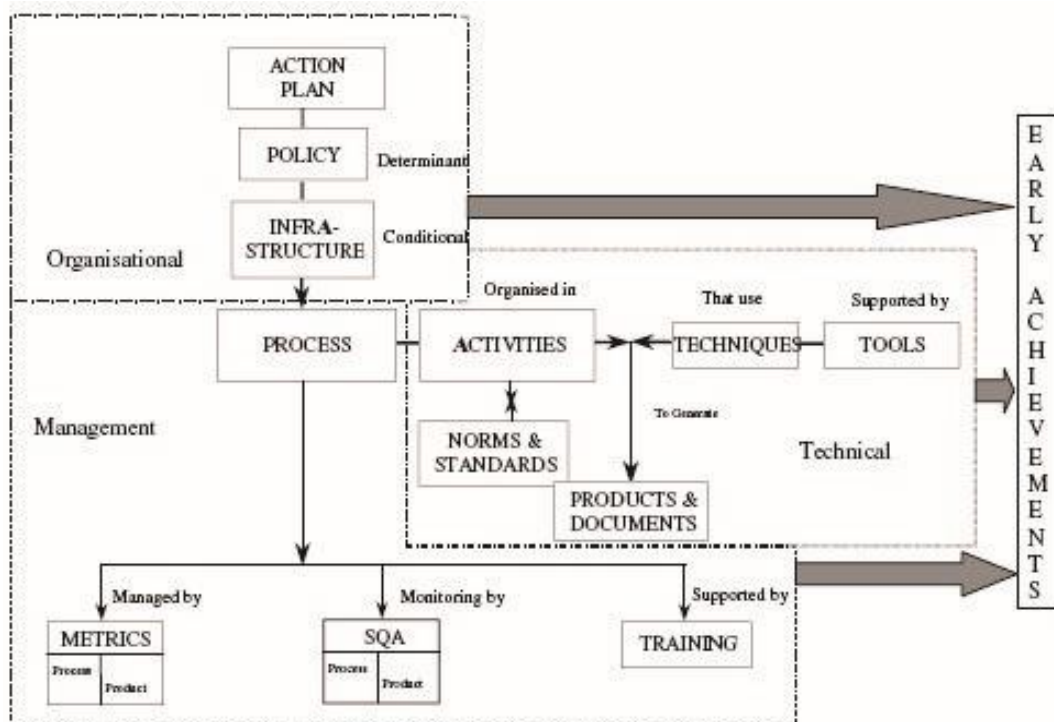


Рисунок 2.4 – Модель GQM Lightweight

Кожен із цих видів діяльності в рамках моделі було налаштовано з огляду на обмежені ресурси та переважну неформальність у невеликих компаніях, що займаються програмним забезпеченням.

Етап планування готує встановлення вимірювання програмного забезпечення в організації. У малих і середніх підприємствах не буде створено окремої групи вимірювань через невелику кількість працівників і неформальні структури організацій. Стартова сесія важлива для отримання схвалення та думки всіх людей. Сеанс зворотного зв'язку може бути корисним для мотивації. Визначення та вибір цілей вимірювання є легшим для МСП через те, що менше людей залучено до пілотного проекту, і часто процес визначення цілей є більш обмеженим.

На етапі збору даних дані збираються відповідно до процедур, зазначених у плані вимірювань. Розробивши відповідні інструменти збору даних та інтегрувавши їх у програмний процес, зусилля можна суттєво скоротити. Фаза інтерпретації спрямована на періодичний аналіз зібраних даних та інтерпретацію під час сеансів зворотного зв'язку за участю персоналу проекту за планом GQM «знизу вгору».

На етапі упаковки результати вимірювань, включаючи зібрані дані та їх інтерпретацію, аналізуються, упаковуються та зберігаються у спосіб, який відповідає організаційному контексту, щоб ці знання можна було повторно використовувати в майбутніх проектах програмного забезпечення та програмах вимірювання. Тут слід зосередитися на документуванні плану GQM, результатах сеансів зворотного зв'язку та звіті про витрати та вигоди.

РОЗДІЛ 3. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

3.1 Охорона праці та її актуальність в ІТ-сфері

Для підвищення ефективності системи управління охорони праці (СУОП) дуже важлива роль належить формуванню і розвитку інформаційної культури фахівців ІТ-технологій, яка впливає на удосконалення інформаційного контуру сучасних підприємств, дозволяє створювати надійні прогнози щодо стану умов праці, показників здоров'я та працездатності, виробничого травматизму і професійної захворюваності, визначати політику розвитку підприємств, установ та організацій на основі різноманітних стратегій охорони праці (інноваційні, маркетингові, інвестиційні, фінансові, технологічні, диверсифікаційні). Поряд з інформаційною культурою важливо використовувати в рамках СУОП «трикутник» її складових: правову, організаційну, управлінську.

В управлінні охороною праці потрібно реалізувати основні положення, окремі теоретико-методологічні підходи інформаційного менеджменту. Головну роль та відповідальність за стан СУОП мають нести фахівці служби охорони праці сучасного підприємства.

Сучасне суспільство називають постіндустріальним, постеконічним, інформаційним, оскільки йдеться про багатосторонні і кардинальні зміни у розвитку цивілізації.

Інформаційне суспільство передбачає докорінну зміну, яка полягає у перетворенні інформації і знань у головний професійно-виробничий потенціал особистості, соціуму і держави.

На постіндустріальному етапі розвитку суспільства вирішальним фактором стає інформація. Її домінування ініціювала науково-технічна революція, яку ще іменують інформаційною, оскільки нею охоплена будь-яка інтелектуальна діяльність, починаючи з інформаційних образів штучного

інтелекту у нових технологіях, економіки, і продовжуючи інформатизацією суспільства в умова світової глобалізації науки й освіти тощо.

Інформаційні технології розглядаються як потужний важіль економічного зростання України. Для цього необхідні значні стратегічні інвестиції у комп'ютерну та комунікаційну інфраструктуру, програми досліджень і розробок, освітню галузь [7].

Під інформаційною культурою розуміють сукупність, складову НІТ (новітні інформаційні технології), технологічну, правову, психологічну, соціологічну та ергономічну підсистеми, що сприяють спрямованому впливу на протікання соціальних процесів у суспільстві, колективі і вихованню свідомого відношення людини до праці, виконання прав та обов'язків [8].

Поняття інформаційної культури виникло в процесі активізації дослідницької уваги до механізмів інформаційного обміну у зв'язку зі значним підвищення ролі інформації в соціокультурних процесах суспільства, яке розглядають як інформаційне суспільство знань, де в центрі знаходяться інформаційні технології.

Робота з інформацією та інформаційна культура в цілому є одним з найважливіших компонентів спроб компанії управляти змінами. Є три принципові причини, в силу яких сьогодні необхідно дбати про інформаційну культуру компанії.

По-перше, вона все більше і більше стає найважливішою частиною загальної організаційної (корпоративної) культури компанії. Все більше компаній розуміють необхідність перетворень, орієнтованих на задоволення очікувань споживача. Щоб сьогодні впливати на майбутнє, потрібно уявляти собі на що вона буде схожа. А для цього потрібно працювати з різноманітною діловою, професійною, технологічною, соціальною, ринковою та політичною інформацією.

По-друге, інформаційні технології роблять можливим створення в компаніях комп'ютерних мереж, за допомогою яких йде спілкування між

менеджерами, але важливо знати, як люди використовують цю інформацію. Саме по собі створення такої мережі з усіма її робочими станціями і мультимедійними можливостями не гарантує того, що інформація буде використовуватися більш розумно і більш ефективно.

По-третє, для різних функціональних служб, підрозділів та робочих груп сучасних підприємств в сфері охорони праці інформаційна культура різна, а це означає відмінність методологічних підходів до процесів усвідомлення, збору, організації, обробки, поширення і використання інформації. Тому багато менеджерів погодяться з тим, що корпоративна інформаційна культура важлива для вироблення різних стратегій охорони праці та запровадження відповідних заходів з її вдосконалення.

Для деяких галузей, таких як розробка програмного забезпечення, інформаційна культура є необхідною умовою виживання, тому що зміна технологій в розробці програмного забезпечення відбувається кожні 6-8 місяців, а інвестиції на підготовку персоналу і освоєння нової технології величезні і у великих компаніях варіюються від 1,5 до 2 млрд. доларів на рік [17].

Аналіз свідчить, що інформатизація та інтеграція комунікаційного простору України сприяє різкому підвищенню інформаційної та професійної компетентності, ділової активності, стимулюванню конкуренції, створенню інноваційних підприємств та організацій, нових робочих місць, зниженню витрат на утримання управлінського апарату [16].

Поряд із задачами і здобутками окреслилися негативи використання інформаційних технологій:

1) надмірне інформаційне навантаження, суть якого полягає у тому, що кількість корисної інформації, яка надходить до мережі, перевищує психофізіологічні можливості її сприйняття людиною;

2) велика кількість інформації, яка сприймається, але не є корисною для фахівців в даний момент;

3) інформаційний голод, причиною якого є саме надлишок інформації, викликаний інформаційним перенавантаженням;

4) «інформоманія» як хвороба людини, яка робить останню знеособленою, залежною від перебування в інформаційному просторі і роботи з комп'ютером і чому вона віддає перевагу, уникаючи «живого» спілкування з людьми;

5) поява «кіберспільнот», що за своїми соціокультурними характеристиками набагато ближчі до представників інших культур у глобальному інформаційному просторі, ніж до своєї етнонаціональної спільноти чи решти населення, не охопленого Інтернетом;

б) індивідуалізм і дегуманізація способу життя «мешканців» Інтернету – відсутність готовності ділитися своїми знаннями.

Слід розуміти, що комп'ютерні технології, а особливо їх мережі істотно впливають на життєдіяльність людини, припускаючи глобалізацію і технократизацію суспільства. Але в ще більшій мірі цей вплив поширюється безпосередньо на центральну нервову систему, яка звикає працювати в дуже інтенсивному режимі багатозадачності, де вже переважають не тривалі логічні роздуми, а інтуїтивно-реактивні ланцюжки розумових формулювань у зв'язку з величезним обсягом оброблюваної щодня інформації, кількість якої зростає за експоненціальною швидкістю. Виникає припущення, що саме збільшення обсягу інформації та прискорення її обробки людиною може згубно вплинути на розвиток розумових здібностей людини.

Аналіз продуктивності розумової праці в найбільших за чисельністю фахівців ІТ-фірм показав, що велике значення з точки зору впливу на її результати має організаційна (корпоративна) культура. В цьому напрямі влаштовуються різні тимблдинги, заходи, тренінги для розвитку персоналу. Також кожен керівник повинен добре розуміти свого співробітника, що саме для нього важливо, що його мотивує. Важливо відвести потрібну роль відповідному співробітнику, щоб він виконував ті завдання, які йому цікаві.

На подібних тренінгах в тому числі повинна розглядатися інформаційна культура працівника, в освоєнні, володінні, мотивуванні, застосуванні, перетворенні інформації із застосуванням сучасних інформаційних технологій і використанням цих умінь в навчанні з охорони праці і в подальшій професійній діяльності. Особливо вони будуть корисні, як доповнення до існуючих інструктажів з охорони праці на підприємстві, або як контроль психологічного стану та взаємовідносин у колективі.

Інформаційна культура як інтегративне утворення абсолютно не зводиться до розрізнених знань, вмінь та навичок роботи за комп'ютером. Вона передбачає інформативну спрямованість цілісної особистості, яка володіє мотивацією до застосування і засвоєння нових даних. Інформаційну культуру можна розглядати, як одну з граней особистісного розвитку промислових робітників. Це шлях універсалізації якостей людини.

Оволодіння інформаційною культурою сприяє реальному розумінню особистістю свого місця, себе і своєї ролі у виробничому колективі. Вона має сприяти формуванню нового покоління фахівців інформаційного суспільства, який повинен володіти наступними навичками: виділення релевантної, значущої інформації, диференціації вихідних даних, розробки інформативних критеріїв її оцінки інформації, вміння використовувати її в рамках СУОП.

Сьогодні продовжує діяти стратегічне правило «Можливості комп'ютерної техніки обмежені тільки нашими уявленнями» [15].

3.2 Шкідлива дія шуму та вібрації і захист від неї

Для запобігання шкідливої дії шуму і вібрації на організм працюючих проводяться технічні, організаційні і медикопрофілактичні заходи.

Одним з основних технічних заходів є зменшення при експлуатації та на стадії проектування, конструювання обладнання причин шуму і вібрації в самому джерелі утворення. Досягають цього завдяки використанню

раціональної конструкції обладнання, заміни ударної дії деталей і машин коливальною, з'єднання елементів гнучкими зв'язками, врівноважування обертових частин механізмів, заміни металевих деталей пластмасовими, забезпечення різних власних частот коливань механізму з частотою збуджуючої сили. Аеродинамічний шум може бути зменшений застосуванням глушників та повітропроводів зі змінним перерізом. Шум трансформаторів (електромагнітний шум) знижується, якщо застосувати листи заліза як складових осердя трансформатора з малою магнітострикцією, серцевини.

Якщо неможливо ізолювати чи знизити шум і вібрацію самого джерела, потрібно:

- ізолювати джерело шуму або вібрації від навколишнього середовища засобами вібро- та звукоізоляції
- раціонально планувати виробничі приміщення, що мають інтенсивні джерела шуму;
- збільшувати звукопоглинання внутрішніх поверхонь приміщення шляхом звукопоглинальних покриттів.

Принцип роботи звукоізоляційних екранів оснований на відбиванні звукової хвилі від різних екранів, стін, кожухів обладнання. Шумливі агрегати слід закривати звукоізоляційними кожухами з виводом назовні органів керування та контрольних приладів. Звукоізоляційні екрани виготовляють з металу, деревини, пластмаси та інших щільних матеріалів. Екрани зсередини покривають звукопоглинаючими матеріалами (скловатою пінополіуретаном), а по периметру кожуха – віброізоляційними підкладками (гума).

Вихідними даними для розрахунку параметрів необхідного екрану є спектр шуму, який необхідно ослабити, кількість екранів, через які проходить шум, їх площа, акустичні характеристики приміщення.

За розрахованими значеннями необхідної звукової ізоляційної здатності екрану підбирається матеріал конструкції й екрану.

Принцип звукопоглинання оснований на явищі трансформації коливальної енергії звуку в теплову через втрати при терті. Найбільші втрати при терті мають пористі, волокнисті і перфоровані матеріали: поролон, пемзолітові і деревоволокнисті плити тощо.

Енергія звукової хвилі переходить у теплову енергію, причому, ефект звукоізоляції збільшується з ростом частоти звукової хвилі. Звукопоглинаючими матеріалами оббивають стелі, стіни. Щоб одержати ефективну звукоізоляцію, найбільш доцільно застосовувати багатошарові огороження з м'якими прошарками (мінеральна вата).

Важливим технічним рішенням у забезпеченні виробничих умов є вдосконалення ручних віброінструментів. Для цього використовують віброгасіння, змінюють ударний вузол, проводять балансування частин, що обертаються.

Послаблення локальної вібрації і передачі вібрації на підлогу і сидіння досягається засобами віброізоляції і вібропоглинання, застосуванням пружинних і гумових амортизаторів, прокладок тощо. Для обмеження поширення вібрацій через ґрунт, між фундаментом і ґрунтом залишають повітряні проміжки, які називаються акустичними розривами.

В останні роки знаходять застосування динамічні віброгасники, в яких створюються вібрації, що співпадають по частоті і протилежні по фазі вібрації машини, коливання якої необхідно зменшити.

До організаційних заходів по боротьбі з шумом та вібрацією на виробництві відносяться: впровадження раціонального режиму праці і відпочинку, обмеження часу роботи при використанні ручного інструменту, який створює вібрацію.

Глушники звуку застосовуються для зменшення шуму аеродинамічних установок (вентиляторів, пневмоінструментів, газотурбінних, дизельних, компресорних установок). Вони поділяються на активні, які поглинають звукову енергію, що на них постувила, і реактивні, які відбивають цю енергію. Потужні

джерела шуму як правило розміщують в окремих приміщеннях, які віддалені від постійних робочих місць.

Ізоляційні kabіни або екрани застосовують як екрани робочих місць для зменшення зовнішніх шумів.

Якщо не вдається зменшити рівень шуму і вібрації на робочому місці до нормативних значень та необхідно використовувати засоби індивідуального захисту: рукавиці, взуття, навушники, м'які шоломи, які зменшують рівень звукового тиску на 40-50 дБ.

У процесі виробництв, експлуатації і зберігання радіоелектронних засобів можуть виникати механічні і динамічні дії, що характеризуються широким діапазоном частот коливань, а також амплітудою, прискоренням і часом дії. Рівень механічних дій визначається умовами транспортування й експлуатації.

Необхідно розрізняти два види механічних дій: удари і вібрації. Удар виникає, коли апаратура отримує швидко зміну прискорення (піддаються удару входи кабелів, джгути, резистори, конденсатори, напівпровідникові діоди і тріоди, силові трансформатори, дроселі тощо). Вібрації – довготривалі знакозмінні процеси, які впливають на роботу апаратури при безпосередньому контакті з джерелом коливань або через повітряне середовище.

У результаті дії вібрацій і удару можуть бути наступні ушкодження апаратури: порушення герметичності через псування паяльних, зварних і клеєних швів і появи тріщин у метало-скляних спаях; повне руйнування корпусів або окремих їх частин через механічний резонанс або циклічну втому; обривання монтажних зв'язків, відшарування багат шарових друкованих плат, руйнування підставок; вихід з ладу електричних контактів; модуляція розмірів хвилеводних трактів; коаксіальних кабелів, конденсаторів змінної ємності, коливальних контурів, електровакуумних приладів, зміщення положення органів настроювання і управління.

Під впливом вібрацій може статись зміна параметрів напівпровідникових приладів, вольт амперних характеристик діодів, транзисторів. Все це призводить до руйнування конструкцій за рахунок явищ втоми.

Радіоелектронна апаратура (РЕА) повинна мати віброміцність, вібростійкість, ударостійкість.

Захист РЕА здійснюється наступними групами методів:

- зменшується інтенсивність джерел вібрації шляхом балансування, зменшення зазорів, віброізоляції джерела вібрацій;
- зменшується величина дій, що передається апаратом шляхом віброізоляції, демпфірування, виключення резонансів, активного віброзахисту за допомогою ексцентриків, маятників, гіроскопів;
- використання найбільш добротні і жорсткі компоненти і вузли;
- застосовуються амортизатори.

Захист часом, захист віддалю, усунення джерела тепловиділення, теплоізоляція, охолодження гарячої поверхні, забезпечення тепловіддачі тіла людини та індивідуальні засоби захисту.

Захист часом передбачає обмеження часу перебування робітника в зоні дії інфрачервоного випромінювання. Потужність випромінювання можна знизити за рахунок конструкторських і технологічних рішень (змінюю нагрівання виробів у нагрівальних пічках індукційним нагріванням та ін.) і за рахунок покриття поверхні, яка нагрівається, тепло ізолювальним матеріалом.

Якщо теплоізоляція неможлива, тоді захист від прямої дії інфрачервоного випромінювання здійснюється екрануванням.

Екрани можуть бути прозорими, напівпрозорими і непрозорими.

У свою чергу вони поділяються на тепловідбивальні, тепловідвідні та теплопоглинальні; стаціонарні і нестаціонарні.

Застосовують також прозору водяну завісу у вигляді суцільної тонкої водяної плівки. Вода є активним поглиначем інфрачервоного випромінювання.

Перегрівання людини попереджують раціональним режимом пиття, режимом праці та гідро процедурами. Спецодяг виготовляється з незаймистого, стійкого до інфрачервоного випромінювання, м'якого і повітронепроникного матеріалу (тканина з металевим покриттям відбиває 90% інфрачервоного випромінювання).

Для захисту очей застосовують світлофільтри зі спеціального жовто-зеленого або синього скла.

Першочергові заходи – це конструкторські і технологічні рішення, які виключають генерацію або понижують інтенсивність випромінювання. Спеціальні засоби захисту (екранування джерел випромінювання, фарбування стін у світлі кольори) попереджують розповсюдження і знижують інтенсивність цих випромінювань у виробничих приміщеннях. Очі захищають окулярами або щитками зі склом – світлофільтром. Для захисту шкіри використовують мазі з речовинами – світлофільтрами для цих променів (салол, саліцилово-метиловий ефір та ін.), а також спецодяг з бавовняних тканин і грубововняного сукна. Руки захищають рукавицями.

ВИСНОВОК

Оскільки ці моделі SPI важко порівняти через їхні відмінні характеристики, ми спробували знайти деякі важливі, але загальні атрибути, щоб ми могли знайти порівняльний погляд на всі доступні моделі SPI. Ці моделі для малих і середніх організацій базуються на деяких існуючих методах, таких як CMM, GQM, QFD тощо. Ці підходи адаптовані та спрощені або шляхом включення додаткових анкет (у моделі самодіагностики), або матриці (у моделі SPM), або інструкцій з процесу (в ASPE-MSD), або пакетів дій (у MESOPYME), щоб їх можна було використовувати малими та середніми організаціями.

Одним із ключових моментів є те, що всі методи, крім моделі самодіагностики, враховують бізнес-цілі організації під час складання плану SPI. Більше того, ці методи (за винятком самодіагностики) є достатньо гнучкими, тому хоча методи визначення та встановлення пріоритетів у сферах покращення пропонуються, але організації також можуть вибрати будь-який інший метод.

Крім того, організації мають можливість вибрати більш важливі для них процеси для плану SPI. Ці методи не тільки визначають, що потрібно покращити, але й надають дорожні карти для того, як це зробити.

Ці моделі SPI спеціально розроблені для малих і середніх підприємств, оскільки ці організації не мають ресурсів і не можуть нести витрати на впровадження CMMI, SPICE тощо, але чи може мала чи середня організація впровадити ці методи без допомоги зовнішньої якості консультант ще не доведено.

Що стосується рівня знань в області SPI, то самодіагностика та MESOPYME не вимагають великого досвіду, тоді як інші моделі потребують багато знань та досвіду, щоб оцінити поточні можливості процесу. Малі та середні організації, як правило, не мають людей, відданих лише якісній роботі.

Людина виконує багато ролей у цих організаціях; наприклад, люди, які розробляють програмне забезпечення, також відповідають за ініціативу SPI. Ці

люди можуть мати або не мати досвіду роботи з ініціативою SPI, тому їм не легше використовувати будь-яку з цих моделей без допомоги зовнішнього консультанта.

Метод самодіагностики легший у використанні, оскільки він дає список КРА, які досягнуті частково або не досягнуті зовсім для досягнення певного рівня СММ. Щоб досягти цих КРА, список дій також може бути згенерований за допомогою матриці GAR, але те, як реалізувати ці дії, не обговорюється в цьому методі, що може бути проблематичним для малої чи середньої організації. Його можна використовувати лише для самооцінки перед офіційною оцінкою.

Малі та середні організації, що займаються розробкою програмного забезпечення, стикаються з серйозними проблемами у забезпеченні якості через обмежені ресурси для інвестицій у забезпечення такої інфраструктури.

Існуючі моделі процесу розробки програмного забезпечення нелегко застосувати в малих і середніх організаціях і потребують розгалуженої інфраструктури. У цьому контексті ми вивчили доступні моделі вдосконалення процесів розробки програмного забезпечення для малих і середніх організацій, що розробляють програмне забезпечення, і порівняли їхні важливі характеристики. Кожна модель має свої переваги та обмеження. Організації повинні вибрати конкретну модель процесу, зважаючи на свої цілі, обмеження та очікування. Подальша робота в цій галузі спрямована на виконання тематичних досліджень та емпіричної перевірки в реальних умовах розробки програмного забезпечення. Було б цікаво вивчити вплив і порівняння цих підходів на якість програмного процесу.

ПЕРЕЛІК ПОСИЛАНЬ

1. Galin, D. (2004). *Software quality assurance, from theory to implementation* (1st ed.). Pearson, Addison-Wesley.
2. Jalote, P. (2002). *An integrated approach to software engineering* (2nd ed.). Narosa Publishing House.
3. Software Engineering Institute. (n.d.). *Improving processes in small settings: A research initiative of the SEI's IPRC*. Retrieved December 12, 2007
4. Herrera, E.M., & Trejo Ramirez, R.A. (2003). A methodology for self-diagnosis for software quality assurance in small and medium-sized industries in Latin America. *The Electronic Journal on Information Systems in Developing Countries*, 15(4), 1-13.
5. Richardson, I. (2002). SPI models: What characteristics are required for small software development companies? *Software Quality Journal*, 10, 101-114.
6. Harchenko, A., Yatcyshyn, V., & Bodnarchuk, I. *The Investigation and Standardization of Quality Indices of Architecture of Software Systems*.
7. Боднарчук, І., & Харчченко, О. (2011). Проблема формального виділення атрибутів для оцінювання якості архітектури програмних систем. Збірник тез доповідей XV наукової конференції Тернопільського національного технічного університету імені Івана Пулюя, 62-62.
8. Alexandr, H., Ihor, B., Iryna, H., & Vasyl, Y. *Software Architecture Design on the Base of Method of Hierarchic Optimization*. In *Proceeding of VIIIth International Conference on Perspective Technologies and Methods in MEMS Design* (pp. 39-40).
9. Harchenko, A., Bodnarchuk, I., & Halay, I. (2013, September). *Decision support system of software architect*. In *2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)* (Vol. 1, pp. 265-269). IEEE.

10. Cohen, L. (1995). Quality function deployment: How to make QFD work for you. USA: Addison Wesley.
11. Gresse von Wangenheim, C. (2002). Planning and executing GQM based software measurement (Tech. Rep. No. LQPS001.01E). Sao Jose, Brazil: UNIVALI.
12. Scott, L., Zettel, J., & Hamann, D. (2000). Supporting process engineering in practice: An experience based scenario. In Proceedings of the Conference on Quality Engineering in Software Technology (CONQUEST) (pp. 160-169), Germany.
13. Allen, P., Ramachandran, M., & Abushama, H. (2003, November 6-7). PRISMS: An approach to software process improvement for small to medium enterprises. In Proceedings of the 3rd International Conference on Quality Software (QSIC'03) (pp. 211-214). Dallas, Texas.
14. Calvo-Manzano, J.A., et al. (2002). Experiences in the application of software process improvement in SMES. Software Quality Journal, 10, 261-273.
15. Шконда В.В., Кальнянов А.В., Давыдов П.Г. Феномен синергетики: наука – общество – образование: Монографія / Ред. Шконда В.В. – Донецк: Норд-Пресс, 2009. – 156 с.
16. Жидецький, В. Ц., Джигирей, В. С., & Мельников, О. В. (2000). Основи охорони праці. Львів: Афіша, 350, 132-136.
17. Навакатіян О.О., Кальниш В.В., Стрюков С.М. Охорона праці користувачів комп'ютерних відеодисплейних терміналів. - К.:1997. - 400с.