**Лекція 9. Процеси управління якістю програмного забезпечення**

**Різниця між забезпечення якості (**Quality Assurance **- QA) та контроль якості (**Quality Control **-** QC**).**

**Якість** відповідає вимогам, очікуванням і потребам замовника без дефектів, недоліків та істотних варіантів. Існують стандарти, яких потрібно дотримуватися, щоб задовольнити вимоги замовника.

**Гарантія якості** – це запевнення, яке надає керівництво організації, що забезпечує гарантію того, що продукт працюватиме без будь-яких збоїв відповідно до очікувань або запитів. **Забезпечення якості** відоме як QA і спрямоване на запобігання дефектам. Забезпечення якості забезпечує правильну реалізацію підходів, методів, методів та процесів, розроблених для проектів. Діяльність із забезпечення якості контролює та перевіряє, чи дотримувались і працюють процеси, що використовуються для управління та створення результатів. Забезпечення якості є ініціативним процесом і носить профілактичний характер. Він визнає вади в процесі. Забезпечення якості повинно пройти перед контролем якості.

**Контроль якості** полягає у тестуванні або перевірці фактичних результатів шляхом порівняння їх із визначеними стандартами.

Контроль якості відомий як QC і зосереджений на виявленні дефекту. Контроль якості гарантує, що підходи, методи, методи та процеси, розроблені в проекті, правильно дотримуються. Діяльність з контролю якості контролює та перевіряє, чи відповідають результати проекту визначеним стандартам якості.

Контроль якості є реактивним процесом, розпізнає дефекти. Контроль якості повинен бути завершений після забезпечення якості.

**Різниця в забезпеченні якості та контролі якості.**

Багато людей вважають, що QA та QC однакові та взаємозамінні, але це неправда. Обидва вони тісно пов’язані, і іноді виявити відмінності дуже важко. Вони за своїм походженням вони різні. QA та QC є частиною управління якістю, однак QA зосереджується на запобіганні дефектам, тоді як QC зосереджується на виявленні дефекту.

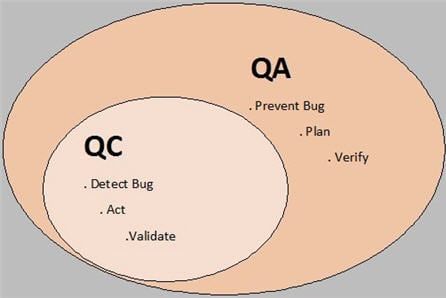
**Ось точна різниця між контролем якості та забезпеченням якості, яку потрібно знати:**

|  |  |
| --- | --- |
| **Гарантія якості** | **Контроль якості** |
| QA означає планування виконання процесу. | Контроль якості означає заходи для виконання запланованого процесу. |
| Це процес, який передбачає надання впевненості, що запит на якість буде досягнутий. | Контроль якості - це процес, який передбачає виконання запиту на якість. |
| Метою контролю якості є запобігання дефекту. | Метою контролю якості є виявлення та покращення дефектів. |
| QA - це техніка управління якістю. | Контроль якості - це метод перевірки якості. |
| QA не передбачає виконання програми. | Контроль якості завжди передбачає виконання програми. |
| Усі члени команди відповідають за забезпечення якості. | Команда тестування відповідає за контроль якості. |
| Приклад контролю якості: Перевірка | Приклад контролю якості: перевірка. |
| Статистична техніка, що використовується для контролю якості, відома як статистичний контроль процесів (SPC). | Статистична техніка, яка використовується для контролю якості, відома як статистичний контроль якості (SPC). |
| QA гарантує, що ви робите правильні речі. | Контроль якості гарантує, що результати того, що ви зробили, такі, як ви очікували. |
| QA Визначає стандарти та методології, яких слід дотримуватися для задоволення вимог замовника. | Контроль якості забезпечує дотримання стандартів під час роботи над продуктом. |
| QA - це процес створення результатів. | Контроль якості - це процес перевірки результатів. |
| QA відповідає за повний життєвий цикл розробки програмного забезпечення. | QC відповідає за життєвий цикл тестування програмного забезпечення. |

**Розглянемо чи гарантія якості усуває необхідність контролю якості, чому якщо QA (Забезпечення якості) зроблено, нам потрібно проводити QC (Контроль якості)?**

Якщо правильно і повністю дотримуватися усіх заздалегідь визначених процесів, політик та стандартів, нам потрібно проводити раунд контролю якості після того, як буде зроблена перевірка якості.

Роблячи «QA», ми визначаємо процеси, політики та стратегії, встановлюємо стандарти, розробляємо контрольні списки тощо, які потрібно використовувати та дотримуватися протягом життєвого циклу проекту. А, виконуючи перевірку якості, ми дотримуємось усіх тих визначених процесів, стандартів та політики, які ми заклали в QA, щоб переконатися, що проект підтримує високу якість, а кінцевий результат проекту принаймні відповідає очікуванням замовника.



Контроль якості спрямований на виявлення та виправлення проблем, тоді як забезпечення якості спрямоване на запобігання виникненню проблем.

**Забезпечення якості не забезпечує якість, навпаки, воно створює та забезпечує дотримання процесів для забезпечення якості. Контроль якості не контролює якість, швидше він вимірює якість.** Результати вимірювання контролю якості можуть бути використані для корекції / модифікації процесів контролю якості, які також можуть бути успішно впроваджені в нових проектах.

Діяльність з контролю якості зосереджена на самому продукті. Діяльність із забезпечення якості зосереджена на процесах, які дотримуються для створення результату.

**Контроль якості та контроль якості є частиною управління якістю, і це потужні методи, які можуть бути використані для забезпечення високої якості матеріалів, що відповідають очікуванням споживачів.**



Коли ми говоримо про тестування програмного забезпечення, воно потрапляє в сферу контролю якості, оскільки воно зосереджується на продукті чи додатку. Ми перевіряємо якість, щоб контролювати її. Крім того, гарантія якості гарантує, що ми проводимо тестування правильно.

**Приклад:** Припустимо, нам потрібно використовувати систему відстеження проблем, щоб реєструвати помилки під час тестування веб-програми.

Забезпечення якості QA включатиме визначення стандарту для додавання помилки та того, які всі деталі повинні бути там, наприклад, підсумок проблеми, де вона спостерігається, кроки для відтворення помилок, знімки екрана тощо. Це процес створення результату, який називається 'повідомлення про помилку'. Коли помилка насправді додається до системи відстеження проблем на основі цих стандартів, тоді цей звіт про помилку є нашим результатом. Ця діяльність є частиною процесу контролю якості.

Тепер, припустимо, деякий час на більш пізній стадії проекту, ми усвідомлюємо, що додавання "ймовірної першопричини'' до помилки на основі аналізу тестера надало б деяке розуміння команді розробників, тоді ми оновимо наш попередньо визначений процес і, нарешті, це також буде відображено в наших звітах про помилки.

Додавання цієї додаткової інформації до звіту про помилку для підтримки швидшого та кращого вирішення проблеми є частиною процесу контролю якості. Отже, таким чином контроль якості QC надає свої вклади в забезпечення якості QA для подальшого вдосконалення якості та остаточних результатів.

**Приклади реального життя для прикладів забезпечення якості та контролю якості**

Припустимо, нашій команді доведеться працювати над абсолютно новими технологіями для майбутнього проекту. Члени нашої команди новачки в технологіях. Отже, для цього нам потрібно скласти план підготовки членів команди до нової технології.

На основі наших знань нам потрібно зібрати необхідні умови, такі як документ про взаєморозуміння, проектний документ, документ про технічні вимоги, документ про функціональні вимоги тощо, і поділитися ними з командою. Це було б корисно під час роботи над новою технологією і навіть було б корисно для будь-якого новачка в команді. Цей збір та розповсюдження документації, а потім розпочата програма навчання є частиною процесу контролю якості.

Після завершення навчання, як ми можемо переконатися, що навчання було успішно проведено для всіх членів команди? Для цього нам доведеться збирати статистичні дані, наприклад кількість балів, які слухачі отримали з кожного предмету, і мінімальна кількість балів, що очікується після закінчення навчання. Крім того, ми можемо переконатись, що всі пройшли повне навчання, перевіривши показники відвідуваності кандидатів.

Якщо оцінки, набрані кандидатами, відповідають очікуванням тренера / оцінювачів, тоді ми можемо сказати, що навчання пройшло успішно, інакше нам доведеться вдосконалити наш процес, щоб забезпечити якісне навчання.

Іншим способом вдосконалення навчального процесу був би збір відгуків від слухачів в кінці навчальної програми. Їх відгуки покажуть нам, що було добре у навчанні та які сфери ми можемо покращити якість навчання. Отже, така діяльність є частиною процесу контролю якості.

**Ключові моменти:**

* У системі контролю якості плануються процеси уникнення дефектів
* Контроль якості погоджується з виявленням дефектів та їх модифікацією під час виготовлення виробу
* Забезпечення якості QA виявляє слабкість
* Контроль якості QC виявляє дефекти
* Забезпечення якості QA орієнтоване на процес
* Контроль якості QC орієнтований на продукт
* Забезпечення якості QA - це система запобігання дефектам
* Контроль якості QC - це система виявлення дефектів.

QA & QC обидва відрізняються один від одного і вимагаються як частина управління якістю. Їх не слід розуміти як взаємозамінні терміни. Забезпечення якості QA орієнтована на процес, а Контроль якості QC - на кінцевий продукт.

Контроль якості - це перевірка товару чи послуги, щоб переконатися, що воно працює нормально. Якщо товар або послуга не працюють нормально, тоді проблему потрібно виправити або усунути, щоб відповідати стандартам відповідності. Отже, він спрямований на виявлення та виправлення проблем.

З іншого боку, забезпечення якості спрямоване на запобігання виникненню проблем у майбутньому шляхом вдосконалення процесу.

Підводячи підсумок, можна сказати, що забезпечення якості не усуває необхідність контролю якості, оскільки контроль якості лежить в основі управління якістю.

**Діяльність по забезпеченню якості**

***Перелік заходів з контролю якості:***

**1) Створення плану управління SQA** (Software quality assurance**)**

Першочергова діяльність включає складання належного плану щодо того, як буде здійснюватися SQA у вашому проекті. Поряд із тим, яким підходом SQA ви будете слідувати, які інженерні заходи будуть виконуватися, а також це включає забезпечення того, щоб у вашій команді була правильна суміш кваліфікацій.

**2) Встановлення контрольних точок**

Команда SQA встановлює різні контрольні точки, відповідно до яких вона оцінює якість проектної діяльності в кожній точці / етапі проекту. Це забезпечує регулярний контроль якості та роботу відповідно до графіка.

**3) Застосування технічних прийомів та програмного забезпечення:**

Застосування деяких методів програмної інженерії допомагає розробнику програмного забезпечення досягти високоякісних специфікацій. Для збору інформації дизайнер може використовувати такі методи, як інтерв’ю та Технологію системи функціонального аналізу FAST .

Пізніше, на основі зібраної інформації, розробник програмного забезпечення може підготувати оцінку проекту, використовуючи такі методи, як WBS (структура розподілу робіт), SLOC (вихідний текст коду) та оцінку функціональної точки FP.

**4) Виконання офіційних технічних оглядів**

Для оцінки якості та дизайну прототипу проводиться зустріч з технічним персоналом для обговорення фактичних вимог до якості програмного забезпечення та якості дизайну прототипу. Ця діяльність допомагає виявляти помилки на ранній фазі і зменшує зусилля з переробки на пізніх фазах.

**5) Наявність стратегії мультітестування**

Під стратегією мультітестування мається на увазі, що не слід покладатися на будь-який єдиний підхід тестування, натомість потрібно проводити кілька типів тестування, щоб програмний продукт можна було тестувати добре з усіх боків для забезпечення кращої якості.

**6) Примусове дотримання процесу**

Ця діяльність наполягає на необхідності дотримання процесу під час процесу розробки програмного забезпечення. Процес розробки також повинен дотримуватися визначених процедур. **Ця діяльність являє собою поєднання двох підзаходів, які детально пояснюються нижче:**

**(i) Оцінка продукції:**

Ця діяльність підтверджує, що програмний продукт відповідає вимогам, визначеним у плані управління проектом. Це забезпечує належне дотримання встановлених стандартів проекту.

**(ii) Моніторинг процесу:**

Ця діяльність перевіряє, чи були вжиті правильні кроки під час розробки програмного забезпечення. Це робиться шляхом узгодження фактично вжитих кроків із задокументованими кроками.

**7) Контроль змін**

У цій діяльності використовується поєднання ручних процедур та автоматизованих інструментів, щоб мати механізм контролю змін. Перевіряючи запити на зміну, оцінюючи характер змін та контролюючи ефект зміни, забезпечується підтримка якості програмного забезпечення на етапах розробки та обслуговування.

**8) Вимірювання впливу зміни**

Якщо команда з контролю якості повідомляє про будь-який дефект, тоді відповідна команда усуває дефект. Після цього команда контролю якості повинна визначити вплив змін, спричинених цим виправленням дефекту. Їм потрібно протестувати не тільки, якщо зміна усунула дефект, а й чи сумісна ця зміна з усім проектом. Для цього використовуються показники якості програмного забезпечення, що дозволяє менеджерам та розробникам спостерігати за діяльністю та пропонованими змінами від початку до кінця життєвого циклу розробки та ініціювати коригувальні дії, де це потрібно.

**9) Виконання аудиту SQA**

Аудит забезпечення якості ПЗ (SQA) перевіряє весь фактичний процес життєвого циклу розробки ПЗ з подальшим порівнянням його із встановленим процесом. Він також перевіряє те, що виконувались дії про які повідомлялося у звітах, і чи відповідає про стан справ, визначений у звітах. Ця діяльність також виявляє будь-які проблеми з недотриманням вимог.

**10) Ведення записів та звітів**

Дуже важливо зберігати необхідну документацію, що стосується SQA, та ділитися необхідною інформацією з SQA із зацікавленими сторонами. Результати випробувань, результати аудиту, звіти про огляд, документацію із запитами на зміни тощо слід зберігати для подальшого використання.

**11) Управління хорошими стосунками**

Насправді дуже важливо підтримувати гармонію між контролем якості та командою розробників. Ми часто чуємо, що тестувальники та розробники часто почуваються вищими один від одного. Цього слід уникати, оскільки це може вплинути на загальну якість проекту.

**Методи визначення показників якості програмного забезпечення**

Методи визначення показників якості програмного забезпечення можна класифікувати виходячи з ряду факторів, які представлені в табл. 1.

Таблиця 1 Класифікація методів визначення показників якості програмного забезпечення

|  |  |
| --- | --- |
| Ознака | Метод |
| Спосіб отримання інформації | Вимірювальний |
| Реєстраційний |
| Органолептичний |
| Розрахунковий |
| Джерело отримання інформації | Експертний |
|  | Соціологічний |
|  | Традиційний |

***Вимірювальний метод*** визначення показників якості програмного забезпечення – це метод отримання інформації про його властивості та характеристики шляхом вимірювань за допомогою інструментальних засобів (наприклад, кількість операторів в програмі, кількість виконаних операторів, кількість операндів, час виконання програми за певних наборах вхідних даних тощо).

***Реєстраційний метод*** визначення показників якості програмного забезпечення – це метод отримання інформації про властивості та характеристики ПС під час його випробування або функціонування, коли реєструються певні події, наприклад, кількість збоїв, відмов.

***Органолептичний метод*** визначення показників якості програмного забезпечення – це метод отримання інформації про властивості та характеристики програмного забезпечення, заснований на сприйнятті таких органів почуттів людини, як зір і слух, наприклад, зручність використання.

***Розрахунковий метод*** визначення показників якості програмного забезпечення – це метод отримання інформації про властивості та характеристики програмного забезпечення, заснований на використанні емпіричних і теоретичних залежностей на ранніх етапах його розроблення, накопичених при випробуваннях, експлуатації та супроводі програмного забезпечення статистичних даних,. Таким чином може визначатися, наприклад, точність обчислень.

***Експертний метод*** визначення показників якості програмного забезпечення – це метод отримання інформації про властивості та характеристики програмного забезпечення на підставі думок групи експертів-фахівців, компетентних у вирішенні даного завдання на основі їх досвіду і інтуїції. Експертний метод застосовується, коли виконати оцінку показників якості за допомогою інших методів неможливо або потребує значних трудовитрат. За допомогою даного методу рекомендується визначати, наприклад, показники зрозумілості і опанування програмного забезпечення.

***Соціологічний метод*** визначення показників якості програмного забезпечення – це метод отримання інформації про властивості та характеристики програмного забезпечення на підставі обробки спеціальних анкет-опитувальників. Цим методом, наприклад, можуть визначатися окремі показники зручності використання.

***Традиційний метод*** визначення показників якості програмного забезпечення - це метод отримання інформації про властивості та характеристики програмного забезпечення на підставі безпосереднього спостереження за їх функціонуванням в процесі роботи. Цим методом, наприклад, можуть визначатися деякі з показників функціональності і зручності використання.

***Вдосконалення роботи QA команд і команд тестувальників***

Потреба в команді з забезпечення якості ПЗ зростає з розміром організації та рівнем її політики в області якості. Там, де потрібно така команда, дуже важливо, щоб функція QA залишалася незалежною від менеджменту проектом і операційної діяльності. Але зв'язки між QA команди з командою проекту повинні надавати їм потужну підтримку. Деякі організації мають функцію QA, вбудовану в офіс управління проектами. Така модель також відповідає критеріям незалежності. Однак, за такої організації треба бути впевненим, що QA команда складається з кваліфікованих аналітиків з забезпечення якості.

З огляду на відмінності розглянутих понять тестування програмного забезпечення, контролю якості і забезпечення якості, є також відмінності між обов'язками QA команд і команд тестувальників.

Обов'язки команд. В обов'язки команд тестувальників входить:

* планування тестування,
* написання тестових сценаріїв і тест кейсів, перевірку тестів,
* виконання тестів,
* аналіз результатів тестування,
* створення і аналіз звітності про результати тестування для різних рівнів випробувань.

В рамках своєї ролі з контролю якості тестувальники можуть висувати вимоги щодо:

* перевірки зразків документів проекту,
* діяльності з управління конфігураціями програмного забезпечення, дизайну, коду тощо.

В той же час QA команди виконують такі функції:

* формування організаційної політики щодо якості, стандартів і процесів розроблення;
* надання допомоги з підготовкою в області забезпечення якості та планів забезпечення якості проекту;
* перевірка відповідності між процесами проекту і планами якості;
* проведення регулярних перевірок проектних продуктів і процесів, а також регулярне представлення результатів аналізу оцінок якості для вищого керівництва;
* загострення ситуації з відхиленням від керівних принципів або стандартів.

В рамках своєї ролі щодо забезпечення якості QA-команда виконує контроль:

* незалежних оглядів;
* наявності процедур управління змінами проектів;
* наявності процедур управління конфігураціями проектів;
* наявності ретроспективи планування і виконання процесів життєвого циклу розробки;
* забезпечення якості на основі розвитку системи життєвого циклу;
* проведення безперервного поліпшення в процесі контролю якості і
* втілення рекомендацій, заснованих на отриманому раніше досвіді.

Виконування обов'язків QA команди ще означає їх розвиток командою, а тільки забезпечення їх реалізації.

***Планування випробувань і документація***. При плануванні випробувань тестувальники готують тестові стратегії і плани, засновані на базових тестових документах, таких як вимоги до програмного продукту і проектні рішення. Ці документи планування тестування є основою випробування процесів на різних запланованих випробувальних рівнях. Для кожного рівня випробувань складаються тести, набори вхідних даних і очікуваних результатів, докладні графіки випробувань, екологічні вимоги, управління дефектами, управління тестуванням і ведення звітності. На відміну від цього, документація з забезпечення якості програмного продукту або плани якості включають більш широкий набір дій протягом усіх етапів розроблення. Це відіграє свою роль при розробленні методології управління проектами.

Типовий проект плану якості включає в себе очікування клієнтів, критерії приймання, плановий контроль якості і аудиту процесів, плани управління конфігураціями і процедури управління змінами. При складанні планів щодо якості виходять з політики самої організації в цій сфері, стандартів або керівних принципів, які формують основи забезпечення якості. В ході створення проекту моніторинг плану забезпечення якості проекту здійснюється безперервно і на його основі оновлюються заплановані показники якості діяльності. Існують різні точки перетину між управлінням ризиками і якістю, і тому реєстр ризиків може зробити вагомий внесок в підготовку планів якості.

***Рекомендації щодо поліпшення роботи команд***.

Перед QA командами встає цілий ряд проблем щодо організації їх роботи, а саме: незалежність. Для успіху QA команди повинні бути незалежними від команди проекту. Це забезпечує QA команді можливість проведення об'єктивної оцінки проектів. В невеликих організаціях тестувальники і QA фахівці можуть перебувати в одній команді. Однак з'являється можливість створення конфлікту інтересів при моніторингу діяльності тестування.

Рішенням проблеми в залежності від політики самої організації в сфері якості може бути створення окремої команди для формування звітності; відносини всередині команди. Якщо аналітики забезпечення якості занадто процесно-орієнтовані і наполягають на процесах або документації, які можуть не мати особливого значення для проекту, це може погіршувати відносини з керівниками проекту. QA команді буде набагато легше працювати з проектними групами, якщо вони будуть працювати за принципом врахування цілей проекту.

Крім того, надання допомоги і сприяння з боку проектних команд формує основу для підтримки хороших відносин, що є важливим аспектом успішного тестування; залучення потрібних фахівців.

Для успішної діяльності QA команд провідну роль відіграє якісна кадрова політика. Люди з досвідом в області розробки ЖЦ програмного забезпечення є хорошими кандидатами для QA команди. Деякі знання в рамках стандартів ISO і принципів CMMI є додатковими корисними знаннями; контрольні списки вимог.

Стандартні контрольні списки є корисним механізмом для проведення аудиту проектів, особливо якщо вони розроблені відповідно до фазами ЖЦ. Для плідної співпраці з менеджерами проекту важливим є забезпечення участі в проекті зацікавлених сторін. Це дає можливість отримати від них зворотній зв'язок у відповідь на пропозиції щодо внесення змін до списків; зв'язок і звітність. Хоча регулярна звітність для вищого керівництва дуже важлива, розроблення правильних шаблонів і показників для забезпечення топ-менеджерів необхідною інформацією гарантує, що цими звітами приділяється належна увага. Це найкраще досягається шляхом проведення зустрічей з відповідними представниками вищого керівництва, надання їм звітів і отримання від них відгуків і коментарів. Крім того, QA командам необхідно постійно отримувати схвалення на внесення змін до процесів контролю якості та стандартів і забезпечувати ефективну взаємодію з зацікавленими сторонами; постійне вдосконалення.

Врахування попереднього досвіду забезпечує QA-команду основою для оцінювання процесів і рекомендаціями щодо забезпечення якості, включаючи постійні поліпшення. QA-команда повинна проявляти гнучкість, підтримувані хороші відносини із зацікавленими сторонами при внесенні поліпшення в управлінську звітність. Безперервні удосконалення можуть також вимагати внесення змін до методології розроблення системи, тому QA командам рекомендується зберігати методологію розробки ІТ-відділу.

**Система управління якістю (СУЯ)**

За допомогою професійної СУЯ визначаються рамки, які забезпечують систематичне управління якістю. Завдяки точному визначенню організаційної структури та процесів, а також виконання всіх завдань з якості, гарантує, що ресурси та заходи, пов’язані з якістю, можна краще скоординувати, спланувати та виміряти.

**ISO 9001:2015-11:** Системи управління якістю - Вимоги

«Базовий» стандарт управління якістю ISO 9001:2015 для систем контролю якості забезпечує цілеспрямований процесно-орієнтований дизайн усіх процесів компанії. Беручи до уваги очікування клієнтів, впроваджуючи ISO 9001, можна уникнути марної трати ресурсів будь-якого роду і піддавати свої процеси постійному вдосконаленню з точки зору якості.

Професійна система управління якістю складається з кількох підаспектів:

1. Усі **дії** та **рішення**, пов'язані з цілями якості,
2. Усі **процеси** та **ресурси** для досягнення цілей, і
3. Усі елементи та особи для **контролю** та **управління** якістю роботи.

Стандарт встановлює універсальні вимоги до системи управління якістю:

1. Підприємницький успіх базується на принципі безперервного вдосконалення відповідно до циклу PDCA (Plan Do Check Act), який відповідає моделі Демінга - Шухерта.
2. Організації потрібні чіткі цілі та стратегії якості.
3. Процеси мають бути прозорими та ефективними.
4. Результати мають бути виміряними та задокументованими.
5. За якість відповідає рівень керівництва.
6. У центрі уваги вся компанія та її оточення.

З моменту запровадження стандарт постійно розвивався. Такий розвиток подій дає зрозуміти, наскільки змінилося поняття якості. Якщо раніше у центрі уваги були продукти та результати, то сьогодні – клієнти та процеси.

**I*SO 9004:2018-08***: Управління якістю - Якість організації - Керівництво для досягнення стійкого успіху.

Посібник не є окремим стандартом для сертифікації, але допомагає досягти сталого успіху та визначити рівень зрілості системи управління якістю.

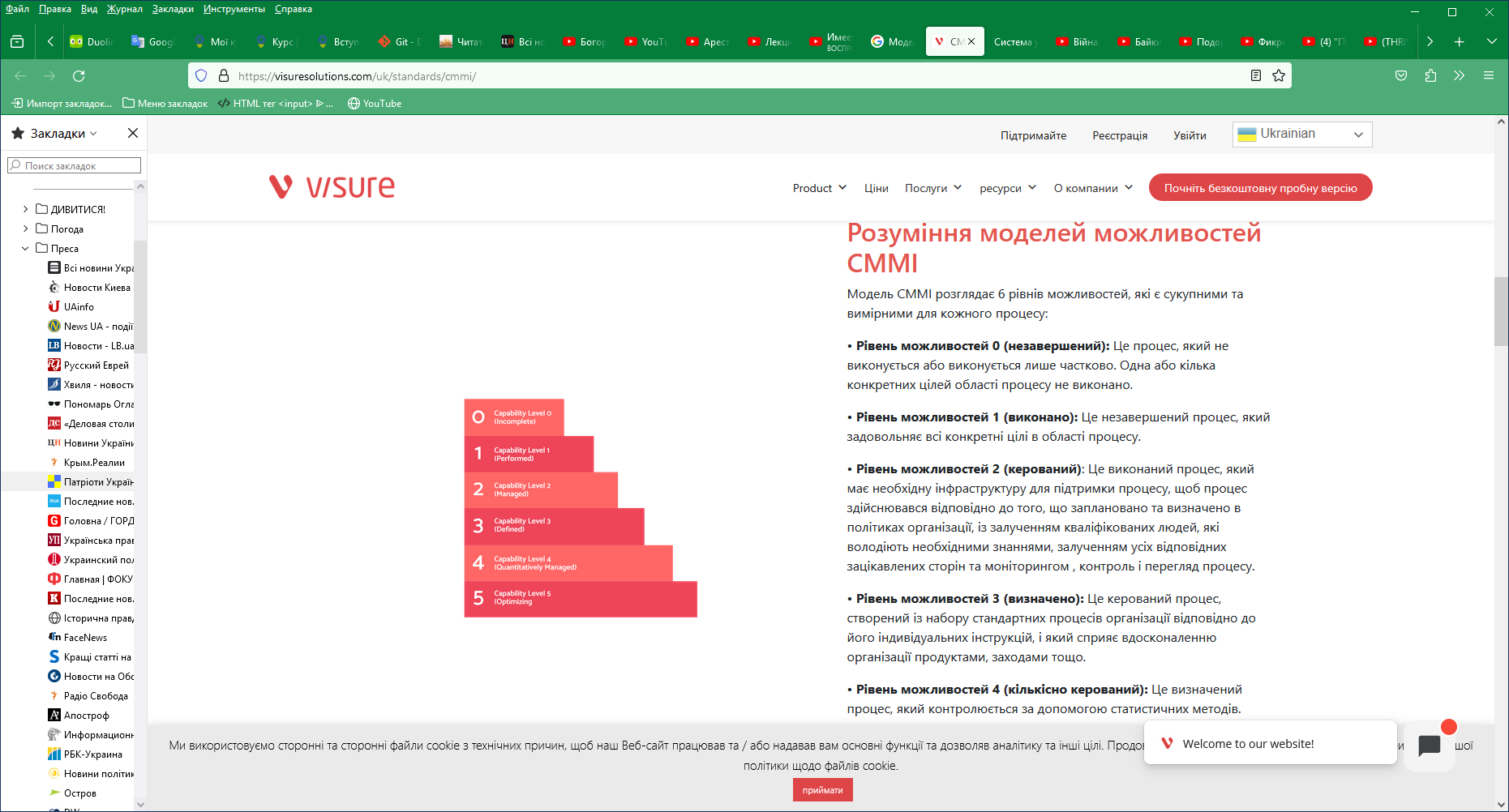
Таким чином компанії, які поєднують позитивні ефекти обох стандартів, можуть не тільки зробити свої процеси ефективними і дієвими, але й полегшити щоденну роботу своїм співробітникам. Робочі процеси просто стають ефективнішими.

**Модель управління якістю розробкою програмного забезпечення CMMI**

CMMI (Capability Maturity Model Integration) описує найкращі практики, які вже застосовуються в галузі, для розробки, підтримки та придбання продуктів і послуг. Він забезпечує структуру, яка дозволяє оцінити рівень зрілості організації або її спроможність щодо процесів, які вона виконує, встановити пріоритети, щоб застосувати на практиці вдосконалення, які мають бути здійснені, і реалізувати ці вдосконалення.

Існує 3 моделі CMMI, усі вони розроблені Інститутом розробки програмного забезпечення (SEI). Один із них, будучи CMMI для розвитку, застосовується до розробки та підтримки продуктів і послуг, незалежно від сфери чи сфери інтересів.

Основними елементами в моделі CMMI для розробки є області процесу; у кожній області процесу CMMI визначає набір конкретних і загальних цілей, а також набір практик як для управління, так і для проектування, які мають бути реалізовані для досягнення цих цілей і охоплюють кожну з сфер процесу.



Модель CMMI розглядає 6 рівнів можливостей, які є сукупними та вимірними для кожного процесу:

**• Рівень можливостей 0 (незавершений):** Це процес, який не виконується або виконується лише частково. Одна або кілька конкретних цілей області процесу не виконано.

**• Рівень можливостей 1 (виконано):** Це незавершений процес, який задовольняє всі конкретні цілі в області процесу.

**• Рівень можливостей 2 (керований)**: Це виконаний процес, який має необхідну інфраструктуру для підтримки процесу, щоб процес здійснювався відповідно до того, що заплановано та визначено в політиках організації, із залученням кваліфікованих людей, які володіють необхідними знаннями, залученням усіх відповідних зацікавлених сторін та моніторингом , контроль і перегляд процесу.

**• Рівень можливостей 3 (визначено):** Це керований процес, створений із набору стандартних процесів організації відповідно до його індивідуальних інструкцій, і який сприяє вдосконаленню організації продуктами, заходами тощо.

**• Рівень можливостей 4 (кількісно керований):** Це визначений процес, який контролюється за допомогою статистичних методів.

**• Рівень можливостей 5 (оптимізація):** Це кількісно керований процес, який покращується через кількісне розуміння причин варіації, спільних для процесу.

Література

1. Ушакова І.О. Проектування інформаційних систем : практикум / І. О.

Ушакова. – Х. : ХНЕУ ім. С. Кузнеця, 2015. – 236 с. (Укр. мов.)

2. Ушакова І.О. Методика управління вимогами в гнучких методологіях /

І.О. Ушакова. // Збірник наукових праць ХНУПС. – 2018.– Вип. 2(56). – С. 93 –98.