МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КПІ»

ІМЕНІ ІГОРЯ СІКОРСЬКОГО

ІНСТИТУТ СПЕЦІАЛЬНОГО ЗВ`ЯЗКУ ТА ЗАХИСТУ ІНФОРМАЦІЇ

ЗВІТ

з дисципліни “Технологія розробки програмного забезпеченя”

Автоматична система управління медпунктом

Виконали: командир С-94 групи

сержант Панченко А.В.

курсант С-94 групи

солдат Кілочицький І.В.

Перевірив: доцет кафедри № 5

Соколов Володимир Володимирович

Київ 2021

ПЛАН

[Розробка графіку випуску версій 3](#_Toc89984762)

[Інкрементна розробка  4](#_Toc89984763)

[Інкрементна розробка поділяє систему на інкременти(порції). 5](#_Toc89984764)

[Розподіл задач по виконавцях 8](#_Toc89984765)

[Масштаб роботи 10](#_Toc89984766)

[Складність роботи 10](#_Toc89984767)

[Сприйняття роботи 11](#_Toc89984768)

# **Розробка графіку випуску версій**

Розробка плану стадій випуску програмного забезпечення відповідно до нашого ТЗ, полягає в поділі відповідних завдань на розділи виконання. В кінці отримали відповідний план:

* **Pre-alpha** початкова стадія розробки; період часу зі старту розробки ПЗ до виходу стадії «Alpha». На відміну від альфа і бета версій, пре-альфа може включати в себе не весь спектр функціональних можливостей програми. У цьому випадку, маються на увазі всі дії, що виконуються під час проектування і розробки програми аж до тестування. До таких дій ми віднесли — розробку дизайну, аналіз вимог, власне розробка програми, а також налагодження окремих модулів.
* **Alpha** до цієї стадії ми віднесли дії під час яких ПЗ підлягає внутрішньому тестуванню. Стадія початку тестування програми в цілому фахівцями-тестерами, зазвичай, не розробниками програмного продукту, а усередині організації або співтовариства, що розробляють цей продукт. Тобто в цю стадію ми віднесли тестування коду самостійно тестувальником.
* **Beta** ц**е**стадія, під час якої ПЗ підлягає публічному тестуванню; стадія активного бета-тестування і налагодження програми, що пройшла альфа-тестування. Програми на цій стадії розробки можуть бути використані іншими розробниками програмного забезпечення для випробування сумісності. Тим не менш, програми цього етапу можуть містити достатньо велику кількість помилок.
* **Release Candidate** - бета-версія, яка має потенціал до становлення готовим до випуску кінцевим продуктом, але якщо не виявляться будь-які значні помилки.
* **Public release -** стабільна версія програми, що пройшла всі попередні стадії розробки, в яких виправлені основні помилки, але існує ймовірність появи нових, раніше не помічених, помилок.

На даний час програма знаходиться у стані “Beta”, тому що є недоліки у графічному плані та реалізації заповнення бази даних але через декілька тижнів перейде у стан “ **Release Candidate**”. “**Public release**” планується через **3 тижні**.

**Серед усіх моделей життєвого циклу ПЗ, ми вибрали**  **Rational Unified Process (інкрементна розробка).**

**Інкрементна розробка** — це серцевина циклічного процесу розробки ПЗ, який був розроблений у відповідь на слабкі сторони водоспадної моделі. Процес починається з початкового планування і закінчується впровадженням (готового ПЗ) з циклічними взаємодіями між цими етапами.

Часто помилково вважають, що «ітеративний» це синонім до «інкрементний». Хоча, в розробці ПЗ та програмних систем вони часто йдуть пліч-о-пліч. Основна ідея полягає в тому, щоб розробити систему шляхом циклів, що повторюються (ітеративний) та в менші проміжки часу (інкрементний), даючи змогу розробнику скористатися перевагами того, що було вивчене під час розробки попередніх порцій або версій системи. Навчання приходить як з процесу розробки так і з процесу використання системи, де можливі ключові кроки процесу починаються з простої реалізації підмножини вимог до програмного продукту та ітеративного вдосконалення версій, які еволюціонують доки повна система не буде реалізована. На кожному кроці (ітерації) виконуються зміни дизайну та додаються нові функціональні можливості.

Процедура складається з кроку ініціалізації, ітеративного кроку та контрольного списку проєкту. Крок ініціалізації створює базову версію системи. Метою початкової реалізації є створення продукту на який можуть відгукнутися користувачі. Цей продукт повинен запропонувати набір зразків ключових аспектів проблеми і забезпечити розв'язання, яке буде достатньо простим для того, щоб легко зрозуміти і реалізувати його. Для керування ітеративним процесом створюється контрольний список проєкту, який вміщує записи усіх завдань, які потрібно виконати. Він включає такі елементи: нові можливості програми, які потрібно реалізувати та частини поточного рішення, які потрібно переробити. Контрольний список постійно змінюється, в результаті фази аналізу.

Ітерація включає реконструювання та реалізацію завдань з контрольного списку, а також аналіз поточної версії системи. Рівень деталізації проєктного плану(дизайну) не диктується ітеративним підходом. У невеликих ітеративних проєктах код може представляти джерело документації системи. Проте, у критичних до організації проєктах може використовуватися формальний документ-план програмного продукту. Аналіз ітерації базується на відгуках користувачів та на доступних засобах аналізу програми. Він включає аналіз структури, модульності, зручності у використанні, надійності, продуктивності та досягнення поставлених цілей. Контрольний список проєкту змінюють на основі результатів аналізу.

## **Інкрементна розробка поділяє систему на інкременти(порції).** У кожній порції доставляється певна частина функціональності. Уніфікований процес поділяє порції/ітерації на такі фази: початок, створення плану, побудова та перехід.

* Початок визначає масштаб проекту, ризики і вимоги (функціональні та нефункціональні) на високому рівні, але достатньому для того, щоб оцінити складність.
* Результатом створення плану стає виробнича архітектура, яка пом'якшує найзначніші ризики та заповнює нефункціональні вимоги.
* Побудова поступово(інкрементно) заповнює архітектуру готовим кодом, який створено завдяки аналізу, плануванню, реалізації та тестуванню функціональних вимог.
* Перехід переводить систему у виробниче середовище.

Кожна з фаз може бути розділена на 1 або більше ітерацій, які, зазвичай, обмежені у часі, а не у функціональності. Архітектори та аналітики працюють на одну ітерацію попереду програмістів та тестувальників, для того, щоб тримати їхній перелік завдань повним.

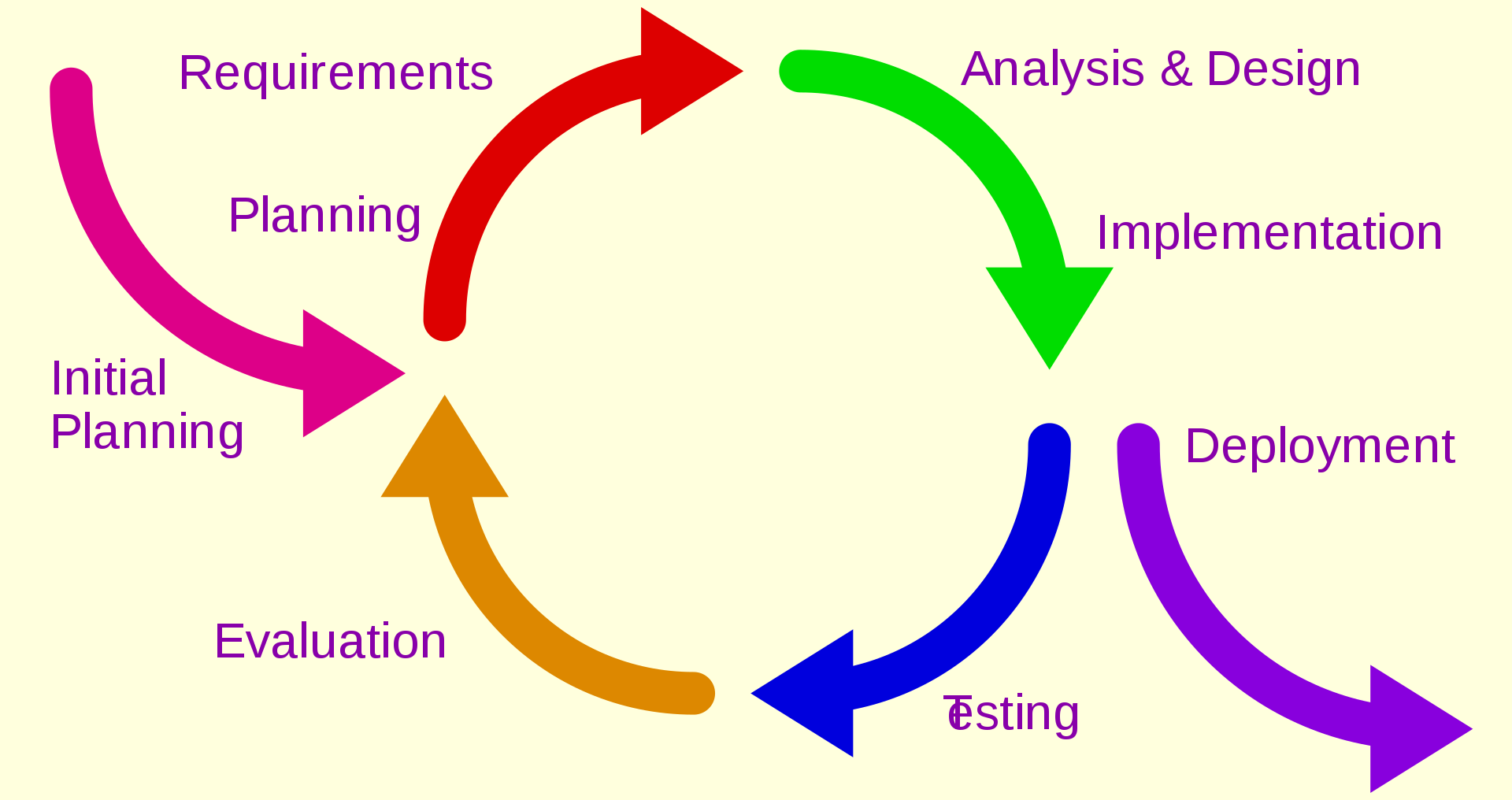


Рис. 1 – інкрементна розробка

**РОЗПОДІЛ ЗАТВЕРДЖЕННЯ**

|  |  |
| --- | --- |
| **Ім'я** | **Посада/вид роботи** |
| Сулім Михайло В’ячеславович | Керівник проекту, програміст |
| Фадєєв Ігор Андрійович | Головний програміст, база даних |
| Рудик Денис Валерійович | Дизайнер |
| Панченко Артур Віталійович | База даних, технічна документація |
| Кілочицький Ілля Володимирович | Технічна документація |

**Сулім Михайло В’ячеславович** – виступає у ролі керівника проекту та споміжним програмістом. Його роль створити правильний розподіл задач між іншими виконавцями та керувати їхніми діями під час робіт. Узгодження задач між компанією виконавців та компанією замовників.

**Фадєєв Ігор Андрійович** – виступає у ролі головного програмісту та провідного виконавця по створенню бази даних. Його задача створити програмне забезпечення у вигляді автономної системи управління для медпункту а також створити базу даних у якій будуть такі пункти:

* Лікарі:
  + П.І.Б.
  + Стать (М/Ж)
  + Посада
  + Спеціалізація
  + Стаж
  + Паспортні дані
  + Дату прийому на роботу
  + Графік роботи
  + Записи пацієнтів на прийом до лікаря
* Пацієнти:
  + П.І.Б.
  + Стать (М/Ж)
  + Дата народження
  + Група крові та резус-фактор (група O, A, B, C; + або -)
  + Зріст
  + Вага
  + Стан лікування (здоровий, призначено ліки, ліжковий режим, госпіталізований)
* Лікувальні процедури пацієнтам
  + Який лікар, коли і якому пацієнту призначив
  + Дата та час запланованого відвідування
  + Статус виконання процедур (відмінено, виконано, пропущено)
* Матеріальне забезпечення медпункту (саме відстеження переміщення того чи іншого медпрепарату або іншого матеріального забезпечення):
  + Коли замовлено
  + Коли доставлено на склад
  + Коли, ким і якому лікарю передано зі складу.
  + Коли, яким лікарем і якому пацієнту передано на лікування/користування
  + Коли, по якій причині списано з обліку на складі

Підтримка працездатності основних функціоналів програмного забезпечення та оновлення її декілька разів на квартал. Робити оновлення програми з усуненням попередніх помилок та багів.

**Рудик Денис Валерійович** – виступає у ролі дизайнера програми. Його робота полягає у створені зручного інтерфейсу для користувачів та адміністраторів. Використання простого але зручного дизайну – це його спеціалізація.

Підтримка працездатності основних функціоналів інтерфейсу. Робити оновлення інтерфейсу з усуненням попередніх помилок та багів, один раз на квартал (під час дрібних оновлень робити виправлення багів).

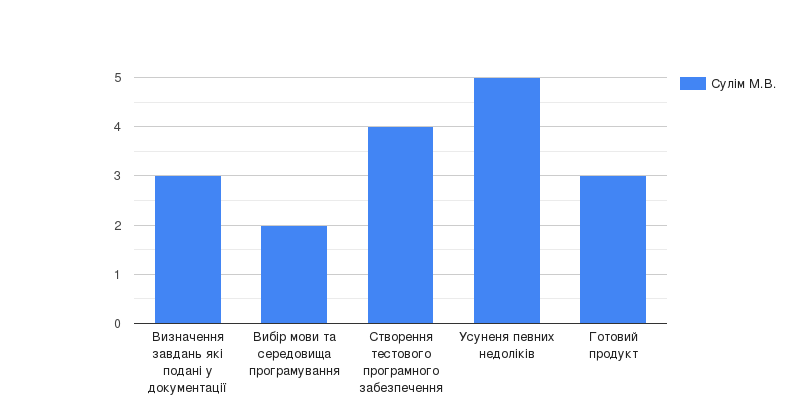
**Панченко Артур Віталійович** – виступає у ролі суміжного програміста зі створення бази даних та виступає у створені технічної документації. Його задача полягає у допомозі при створені бази даних та у заповнені відповідних документів до усього шляху під час створені програмного забезпечення.

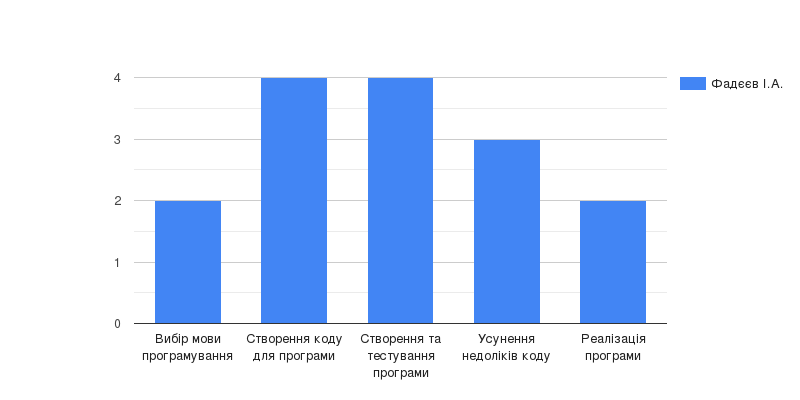
Підтримка праце здібності основних функціоналів програмного забезпечення та оновлення її декілька разів на квартал. Робити оновлення програми з усуненням попередніх помилок та багів. За вимоги оновляти документи які стосуються автономної системи управління.

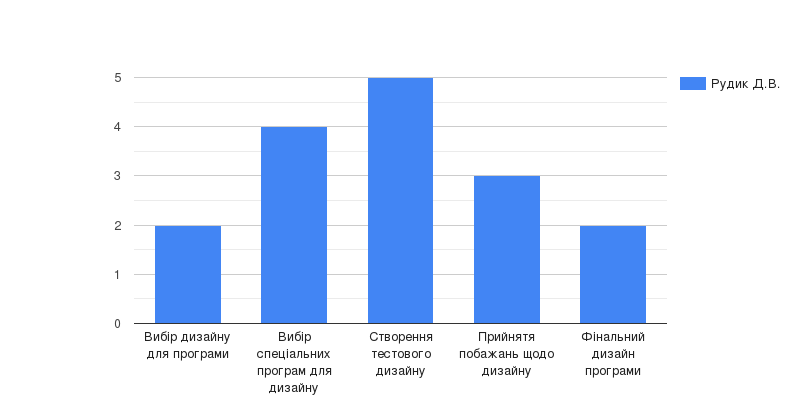
**Кілочицький Ілля Володимирович** – виступає у ролі виконавця по заповненню технічної документації. Його задача полягає у заповнені відповідних документів до усього шляху під час створені програмного забезпечення.

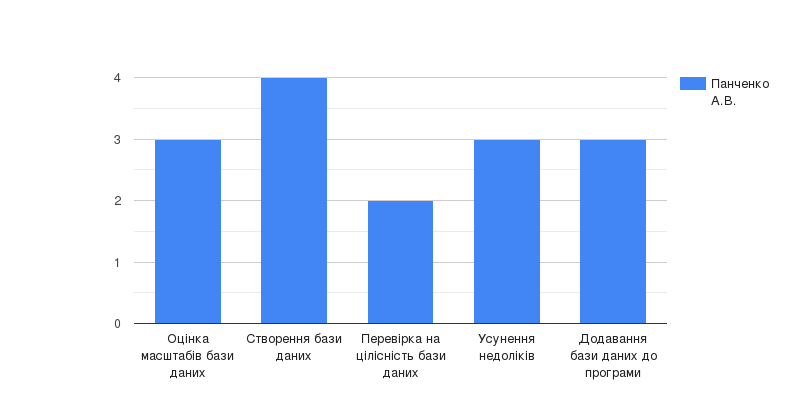
Оновляти документи які стосуються проекту та створювати нові, які під час створення додаються.

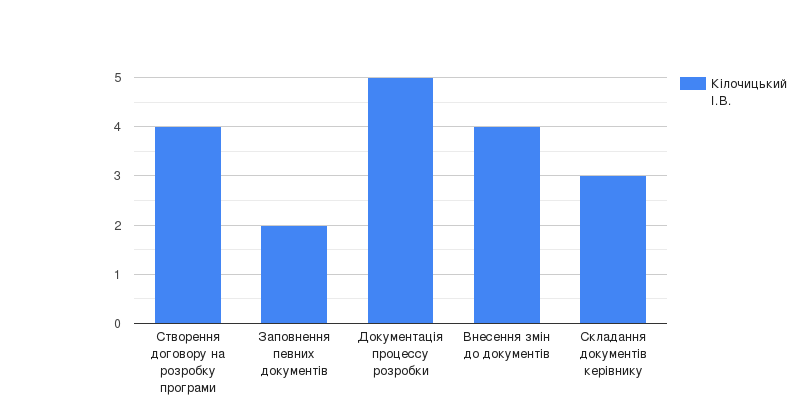
# **Розподіл задач по виконавцях**

. 









Параметри роботи ми визначили на основі результатів її аналізу. До параметрів роботи ми віднесли її масштаб і складність, а також відносини роботи і по роботі, в які її виконавець вступає з іншими працівниками.

**Масштаб роботи** являє собою кількість завдань або операцій, які виконавці повинені виконати за певний термін. Зазвичай, якщо у виконавця більше задач або операцій, це займе в нього більше часу ніж планувалось.

**Складність роботи,** як параметр відображає ступінь самостійності в прийнятті рішень і ступінь володіння процесом. На практиці складність роботи залежить від самого виконавця, його характеру та даних йому прав щодо проекту. В управлінні можна зустріти людей, однакові за метою посади, але виконують різну за складністю роботу. У цьому проявляється неформальна сторона в управлінні організацією..

До вузькоспеціалізованим робіт відносяться ті, які мають низький масштаб роботи, реалізація яких здійснюється за допомогою запропонованих засобів (низька складність роботи). Широко спеціалізовані роботи мають протилежні характеристики розглянутих параметрів. У розвинутих великих організаціях розрив між цими двома видами робіт буває дуже великий. Правильно визначивши масштаб і складність роботи, керівникам допомагають правильно і чітко сформульовані задачі, мети і завдання організації.

Під відносинами по роботі при її проектуванні розуміється встановлення міжособистісних зв'язків між виконавцем роботи та іншими працівниками як з приводу самої роботи, так і в зв'язку з іншими видами робіт в організації.

Розгляд відносин як параметра, що використовується при проектуванні роботи, є містком до формування структури організації. Це пов'язано із завершенням організаційного відокремлення функцій у формі робочих відносин.

Ухвалення керівництвом рішень про природу і характер відносин по роботі тісно пов'язане з такими елементами структури, як масштаб керованості і створення ділянок роботи, служб і підрозділів в організації. Це також має зв'язок зі згуртованістю групи і організаційної культурою.

Так, невиправдано великі розміри групи, що робить загальну роботу, ускладнюють встановлення дружніх і зацікавлених відносин між її членами, що негативно впливає на ефективність виконуваної роботи. Причиною цього є рідкісна можливість для великого числа людей комунікувати один з одним, як по роботі, так і в особистому плані. В результаті страждає згуртованість групи і зникає важливе джерело задоволеності роботою для тих, хто відчуває потребу в соціалізації і самоствердженні шляхом встановлення відповідних відносин зі своїми колегами.

Створюючи ділянки роботи, служби і підрозділи, керівництво організації багато в чому виходить з практичного використання тих відносин по роботі, які вдалося отримати в ході її проектування. Функціональна спільність робіт дозволяє об'єднати їх в рамках однієї групи працівників, а всередині групи розташувати роботи за рівнями ієрархії в залежності від особистісних якостей і делегованих прав. Спільність інтересів по «горизонталі» і по «вертикалі» дозволяє учасникам процесу роботи встановити задовольняють їх соціальні відносини.

**Сприйняття роботи**. Сприйняття змісту роботи характеризує її з точки зору розуміння її природи окремим виконавцем. Існує відмінність між об'єктивними і суб'єктивними властивостями роботи, відображаються у сприйняттях людей. Не можна зрозуміти результат роботи, абстрагуючись від відмінностей в особистісних якостях, потребах і інтересах виконавців. Також не можна зробити це без урахування соціального середовища, в якій ця робота виконується. Щоб поліпшити результат виконання роботи, необхідно змінити сприйняття змісту роботи. Зміни при цьому можуть стосуватися проектування роботи, особистісних якостей або соціального середовища, тобто всього того, що впливає на сприйняте зміст роботи.