"ЗАТВЕРДЖУЮ"

Директор ТОВ «Saints & Sinners (S&S)»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сулім М.В.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021р.

**ІНСТРУКЦІЯ**

**З ОРГАНІЗАЦІЇ ВИРОБНИЧИХ ПРОЦЕСІВ**

Версія: 1.0

Розробив: Сулім М.В.

Введено в дію з 28.10.2021р.

Київ, 2021

ЗМІСТ

[Вступ 3](#__RefHeading___Toc28_3516559530)

[1. Формування домену завдань 5](#__RefHeading___Toc75153585)

[1.1. Загальний процес 5](#__RefHeading___Toc75153586)

[1.2. Процес формування функціональних завдань 6](#__RefHeading___Toc75153587)

[1.2.1. Створення запиту на завдання 6](#__RefHeading___Toc75153588)

[1.2.2. Специфікування вимог до задачі 7](#__RefHeading___Toc75153589)

[1.2.3. Погодження та затвердження специфікацій 8](#__RefHeading___Toc75153590)

[1.3. Процес формування завдань на усунення дефектів 9](#__RefHeading___Toc75153591)

[1.3.1. Створення запиту на усунення дефекту 9](#__RefHeading___Toc75153592)

[1.3.2. Специфікування вимог щодо усунення дефекту 10](#__RefHeading___Toc75153593)

[1.3.3. Погодження та затвердження специфікацій на усунення дефекту 11](#__RefHeading___Toc75153594)

[1.4. Процес формування системних задач 11](#__RefHeading___Toc75153595)

[2. Планування робіт 11](#__RefHeading___Toc75153596)

[2.1. Загальний процес 11](#__RefHeading___Toc75153597)

[2.2. попереднє планування 12](#__RefHeading___Toc75153598)

[2.3. ескізне проектування 13](#__RefHeading___Toc75153599)

[2.4. детальне планування 14](#__RefHeading___Toc75153600)

[3. Виконання плану 14](#__RefHeading___Toc75153601)

[3.1. Загальний процес 14](#__RefHeading___Toc75153602)

[3.2. проектування 17](#__RefHeading___Toc75153603)

[3.3. Розробка 17](#__RefHeading___Toc75153604)

[3.4. тестування 18](#__RefHeading___Toc75153605)

[3.5. документування 19](#__RefHeading___Toc75153606)

[3.6. архівування 19](#__RefHeading___Toc75153607)

[3.7. передача замовнику 20](#__RefHeading___Toc75153608)

[3.8. Впровадження у нового замовника 20](#__RefHeading___Toc75153609)

[4. Аналіз виконання плану 21](#__RefHeading___Toc75153610)

# Вступ

Даний документ задає базову структуру (модель) виробничих процесів ТОВ «Saints & Sinners (S&S)», пов'язаних з розробкою, впровадженням і супроводом програмних систем. Ці процеси описані з різним ступенем деталізації і можуть уточнюватися в ході їх виконання.

В основу моделі виробничих процесів закладений акцент на артефактах, які розглядаються в якості прагматичного результату будь-якого процесу, тобто модель виробничих процесів, керована артефактами.

У загальному випадку будь-яка робота розглядається як процес, в результаті якого створюються артефакти:

артефакт

показники

призначення

критерії

зв'язки

стандарт

стани

процес

інструмент

цілі

завдання

зв'язки

Учасники

критерії

артефакти

Артефакт має атрибути:

* Призначення - описує його застосування;
* Процес його створення;
* Стандарт оформлення (структура, зміст, оформлення, інструмент)
* Критерії якості - задає вимоги до нього;
* Показники - метрики для аналізу (трудовитрати планові, фактичні і т.п.)
* Зв'язки з іншими артефактами;
* Стану, в яких він може перебувати.

Процес має атрибути:

* Мета - очікуваний результат;
* Завдання - складові процесу;
* Учасники процесу (в т.ч. ініціатор);
* Артефакти, які він створює;
* Інструмент автоматизації;
* Зв'язки з іншими процесами;
* Критерії якості - задає вимоги до нього.

Всі артефакти класифікуються на:

* Завдання - безліч робіт (вимоги, специфікації, виправлення помилок), які потрібно виконати;
* Плани - група впорядкованих робіт, обраних для реалізації у встановлені терміни (випуск версій, календарний);
* Результати - артефакти, отримані в результаті виконання плану (проект, код, документація).

сховище артефактів

Учасники процесу

опису процесів

завдання

плани

результати

Кожен клас артефактів має підкласи (види), версії і ЖЦ.

Мета-завданням будь-якого процесу є створення артефакту.

На мета-рівні необхідні опису (+ шаблони) артефактів і опису процесів.

Опис процесів задає штатний (за замовчуванням) режим їх виконання, яке доповнюється (коригується) процесом оперативного (ручного) керування.

Процес оперативного управління заснований на аналізі показників процесу і впливає на організацію процесів і параметри артефактів (пріоритети, варіанти реалізації) для корекції і контролю якості.

**завдання**

нові

дефекти

**результати**

нові

дефекти

**плани**

нові

дефекти

З точки зору теорії систем маємо систему масового обслуговування з чергами і пріоритетами.

Критерій ефективності - максимум (за важливістю) нових результатів при мінімумі повертаються дефектів.

В якості моделі життєвого циклу (ЖЦ) процесів створення системи використовується ітераційна модель - поступове нарощування функціональності при змінюються вимоги користувачів.

Цикл ітерації - 2-3 місяці. У цьому циклі реалізуються всі процеси, пов'язані з плануванням і випуском оновлення системи. При цьому деякі фази сусідніх ітерацій можуть накладатися, наприклад, фаза планування подальшої ітерації може проводитися в кінці попередньої ітерації для забезпечення безперервності процесів, в якому задіяні фахівці різної спеціалізації.

Білд (***build***) - поточне внутрішнє оновлення версії програмного продукту, викликане необхідністю виправлення критичних помилок, додавання важливих функцій і тестування.

Реліз (***release***) - планове оновлення продукту, що містить істотні доопрацювання функціональності і виправлення помилок, що виставляється замовнику.

Версія (***version***) - планове оновлення продукту, що містить фундаментальні функціональні або архітектурні зміни, що виставляється замовнику.

Даний документ може використовуватися як основа для розробки положення про компанії, посадових інструкцій, створення системи якості.

# Формування домену завдань

## Загальний процес

Мета - забезпечення завантаження персоналу проектів відповідно до загальної стратегії підприємства.

Завдання:

1. Формування завдань щодо усунення дефектів (усунення дефектів);
2. Формування завдань на проведення експериментів і досліджень по вибору нових технологій і архітектурних рішень (системні завдання);
3. Формування завдань зі створення нової і зміни існуючої функціональності (функціональні завдання).

Учасники процесу:

* + керівництво;
  + куратор,
  + замовники,
  + аналітики.

Артефакти:

* + вимоги;
  + постановки завдань;
  + специфікації.

Критерії якості:

* + наявність достатньої кількості завдань в домені для завантаження персоналу проектів на плановий період;
  + відповідність завдань стратегії підприємства (концепції системи);
  + виконання критеріїв якості створюваних артефактів.

Черги домену завдань

усунення дефектів

Системні завдання

функціональні завдання

Кожне завдання в черзі має 3 показника пріоритету:

* + пріоритет джерела:
    - керівництво - max;
    - куратор;
    - замовники;
    - аналітики - min;
  + пріоритет черги:
    - дефекти (критичні) - max;
    - системні (критичні);
    - функціональні - min;
  + пріоритет завдання:
    - критична;
    - важлива;
    - середня;
    - низька.

Ці пріоритети використовуються в процесі планування.

Оперативне управління процесом здійснюють уповноважені особи (керівництво, Куратор) шляхом контролю критеріїв якості процесу і артефактів, а також зміни пріоритетів окремих завдань.

Всі завдання проекту, які належить виконати, складають у загальний список - беклог. Беклог - це банк завдань проекту. Кожне завдання має бути актуальна. Якщо буде потрібно, їх можна додавати в беклог або видаляти з нього «на льоту».

Кожне завдання має вагу - зазвичай це час, який потрібно на рішення. Команда сама оцінює вагу всіх завдань, тому якщо проект не закінчений в строк, винна команда.

Також у кожного завдання є пріоритет. У Kanban пріоритети розставляє команда. Пріоритети можна і навіть потрібно переглядати по ходу проекту - це один із стовпів гнучких методологій.

## Процес формування функціональних завдань

Мета - формування постановок задач для подальшої їх реалізації персоналом проекту.

Завдання:

* + створення запиту на завдання;
  + створення специфікацій вимог до задачі;
  + погодження та затвердження специфікацій.

Учасники процесу:

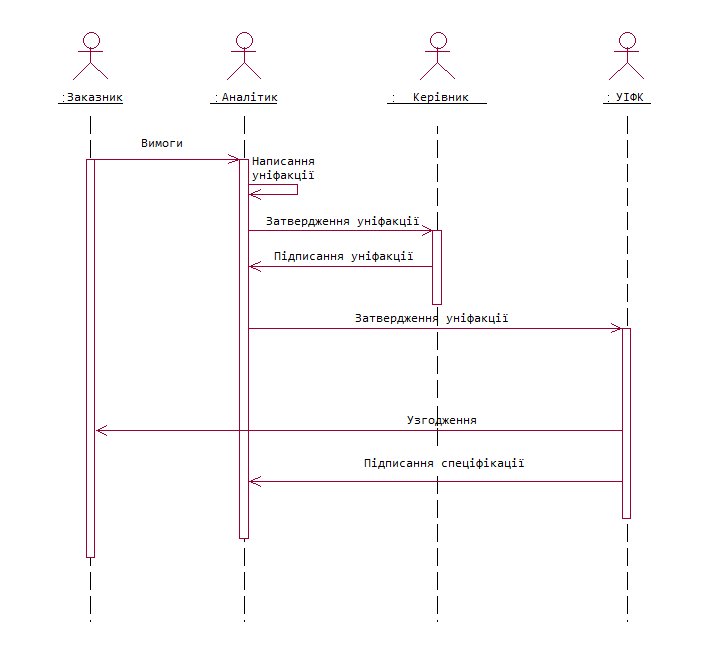
* + ініціатор;
  + відповідальний (аналітик);
  + керівник проекту;
  + сторони, що погоджують.

Артефакти:

* + запит на завдання;
  + специфікації вимог.

Критерії якості процесу:

* + час обробки заявки, специфікування, погодження та затвердження;
  + якість створюваних артефактів.



передумови:

* + необхідна формальна («План управління вимогами») або неформальна домовленість із зацікавленими сторонами щодо організації процесу узгодження і затвердження вимог (специфікацій).

### Створення запиту на завдання

Мета - реєстрація вхідного запиту на розробку (зміну) функціональності та прийняття рішення.

Завдання:

* + заповнення атрибутів запиту;
  + збереження запиту;
  + рішення про прийняття або відхилення;
  + повідомлення учасників.

Учасники процесу:

* + ініціатор;
  + відповідальний (аналітик).

Артефакти:

* + копія вихідного документа (при наявності);
  + запит.

Інструмент автоматизації:

* + Trello

Критерії якості:

* + заповнення всіх необхідних полів бланка запиту;
  + ясність.

Запит має атрибути:

* автор (ініціатор);
* дата, №;
* підсистема;
* суть запиту (коротке формулювання);
* пріоритет;
* посилання на вихідний документ (ТЗ, лист, протокол).

Стан, в яких він може перебувати:

* зареєстрований;
* прийнятий;
* відхилений.

Примітка:

* можна залучати керівництво і Куратор для прийняття рішень за окремими заявками;
* відхилення заявки повинно обов'язково проходити через керівництво.

початок

заповнення атрибутів

Якість?

Збереження і реєстрація

Прийняття рішення

сповіщення

кінець

+

-

Безпосереднє заповнення запиту може робити або ініціатор, або відповідальний (аналітик) на підставі інформації (в будь-якій формі) від ініціатора.

### Специфікування вимог до задачі

Мета - створення повних документованих вимог до програмного забезпечення для реалізації завдання.

Завдання:

* + з'ясування (уточнення) вимог у ініціатора запиту;
  + документування специфікацій;
  + збереження в домені завдань.

Учасники процесу:

* + ініціатор;
  + відповідальний (кер.проекта);
  + виконавець (аналітик).

Артефакти:

* + документ-специфікація.

Інструмент автоматизації:

* + Trello

Критерії якості специфікацій:

* + функціональна повнота;
  + достатня деталізація;
  + відповідність загальної концепції побудови і розвитку системи;
  + відсутність вказівок на спосіб реалізації та планування;
  + несуперечливість;
  + достовірність, і тестований.

Зміст і форми подання специфікацій:

* титульний лист;
* лист погодження та затвердження;
* глосарій;
* текстовий опис вимог;
* діаграми Use case, станів, послідовності і т.п.
* вимоги до інтерфейсу (екранні і друковані форми);
* обмеження даних і передбачувані тести;
* основні і похідні вимоги;
* функціональні і нефункціональні;
* посилання на вихідний документ (ТЗ, лист, протокол);
* посилання на запит в Trello.

Стан:

* в розробці;
* на погодженні; на узгодженні;
* затверджені.

Безпосереднє формування специфікацій виробляє виконавець (аналітик) на підставі прийнятого запиту.

### Погодження та затвердження специфікацій

Мета - отримання документально затверджених вимог, які враховують інтереси всіх зацікавлених сторін і вимог нормативних документів.

Завдання:

* + прийняття рішення про необхідність узгодження;
  + вибір зацікавлених сторін і їх пріоритетів;
  + організація і контроль процесу узгодження (термінів);
  + вирішення спірних питань, внесення змін і узагальнення вимог (при необхідності);
  + передача документа специфікацій на затвердження;
  + збереження в домені завдань і в архіві документів.

Учасники процесу:

* + відповідальний (кер.проекта);
  + виконавець (аналітик);
  + керівництво;
  + сторони, що погоджують.

Артефакти:

* + узгоджений і затверджений документ-специфікація.

Інструмент автоматизації:

* + GitHub Projects,
  + Trello,
  + Discord.

Критерії якості:

* + повнота врахування вимог різних сторін;
  + несуперечливість нормативним документам;
  + відповідність загальної концепції побудови і розвитку системи;
  + достовірність, і тестований.

Безпосереднє узгодження специфікацій виробляє відповідальний (аналітик) за участю керівництва та Куратора для організації процесу.

На погодження та затвердження виносяться істотні (значущі) частини програми.

## Процес формування завдань на усунення дефектів

Мета:

* + оперативне усунення виявлених дефектів;
  + аналіз і усунення причин дефектів;
  + підвищення якості продукту і зниження кількості помилок у користувача;
  + зменшення числа помилок в процесі тестування.

Завдання:

* + створення запиту на усунення дефекту;
  + створення специфікацій щодо усунення дефекту;
  + погодження та затвердження специфікацій.

Учасники процесу:

* + ініціатор;
  + відповідальний (тестувальник);
  + керівник проекту;
  + погоджують сторони.

Артефакти:

* + запит на усунення дефекту;
  + специфікації вимог на усунення дефекту.

Критерії якості процесу:

* + час обробки заявки, специфікування, погодження та затвердження;
  + якість створюваних артефактів.

### Створення запиту на усунення дефекту

Мета - реєстрація вхідного запиту на усунення дефекту та прийняття рішення.

Завдання:

* + заповнення атрибутів запиту;
  + збереження запиту;
  + відтворення дефекту;
  + рішення про прийняття або відхилення;
  + повідомлення учасників.

Учасники процесу:

* + ініціатор;
  + відповідальний (тестувальник).

артефакти:

* + копія вихідного документа (при наявності);
  + запит.

Інструмент автоматизації:

* + Github Issues (в т.ч. експорт негативних результатів з інструменту тестування)

Критерії якості:

* + заповнення всіх необхідних полів бланка запиту;
  + ясність;
  + відтворюваність дефекту.

Запит має атрибути:

* автор (ініціатор);
* дата, №;
* версія, підсистема;
* опис дефекту;
* копії екранних форм;
* програмні та системні повідомлення про помилку;
* пріоритет.

Стани, в яких він може перебувати:

* зареєстрований;
* прийнятий;
* відхилений.

Примітка:

* можна залучати керівництво і Куратора для прийняття рішень по окремим запитам;
* Відхилення запрошення повинно обов'язково проходити через керівництво.

початок

заповнення атрибутів

Якість?

Збереження і реєстрація

Прийняття рішення

сповіщення

кінець

+

-

Відтворені-ня

Безпосереднє заповнення запиту може виробляти або ініціатор, або відповідальний (тестувальник) на підставі інформації (в будь-якій формі) від ініціатора.

### Специфікування вимог щодо усунення дефекту

Мета - створення повних специфікацій для усунення дефекту.

Завдання:

* + з'ясування (уточнення) дефекту у ініціатора запиту;
  + локалізація місця дефекту в програмі і відновлення вихідних специфікацій;
  + визначення можливих причини появи дефекту;
  + документування специфікацій;
  + збереження в домені завдань.

Учасники процесу:

* + ініціатор;
  + відповідальний (тестувальник);
  + аналітик;
  + виконавець (кер.проекта).

Артефакти:

* + документ-специфікація.

Інструмент автоматизації:

* + GitHub Issues;
  + GitHub Projects.

Критерії якості специфікацій:

* + функціональна повнота;
  + достатня деталізація;
  + несуперечливість;
  + достовірність, і тестований.

Зміст і форми подання специфікацій:

* передбачувана причина дефекту;
* опис того, як має бути з посиланням на специфікації вимог;
* локалізація на діаграмі Use case, станів, послідовності і т.п.
* локалізація на інтерфейсі (екранної і друкованій формі);
* основні і похідні дефекти;
* посилання на вихідний документ дефекту (ТЗ, лист, протокол);
* посилання на запит в GHI.

Стан:

* в розробці;
* на погодженні; на узгодженні;
* затверджені.

Безпосереднє формування специфікацій виробляє відповідальний (тестувальник) на підставі прийнятого запиту.

### Погодження та затвердження специфікацій на усунення дефекту

Виконується аналогічно п.1.2.3.

## Процес формування системних задач

Виконується аналогічно п.1.2.

# Планування робіт

## Загальний процес

Мета - формування узгодженого графіка якісного вирішення першочергових завдань у визначені терміни відповідно до загальної стратегії підприємства.

Завдання:

1. Попереднє планування - вибір пріоритетних завдань для включення в план і попередня оцінка масштабу (трудовитрат) і ризиків;
2. Проектування з уточненням масштабу і ризиків;
3. Створення детального плану, розподіл робіт, погодження та затвердження.

Учасники процесу:

* + керівництво;
  + керівники проектів.

Артефакти:

* + ескізний проект функціональних і системних завдань;
  + план робіт на період (випуску версії) із зазначенням пріоритетів, масштабів і ризиків.

Критерії якості:

* + повнота обліку пріоритетних завдань;
  + системність (включення пов'язаних завдань, їх групування, облік забезпечують завдань);
  + виправданий масштаб;
  + узгодженість між проектними групами.

**Черги домену завдань**

усунення дефектів

Системні завдання

функціональні завдання

**План на період**

**Загальний план компанії**

План по п / с 1

План по п / с 2

План по системі

ескізні проекти

Інструмент автоматизації:

* + GitHub Project (Синхронізація GHI/GHP)

Оціночна частка корисного часу (без урахування відряджень, хвороб, відпусток) на типи завдань повинна становити:

* + усунення дефектів - 25% (в перспективі 15%);
  + функціональні та системні завдання - 65% (в перспективі 75%);
  + резерв - 10%.

План верстається за допомогою нарад: інсталяційний, попереднього і остаточного.

На установчому нараді керівництвом формулюються основні цілі та головні завдання (можливо, відсутні в домені завдань) для попереднього планування.

На нараді за попередньою планування розглядаються плани проектних груп, представлені їх керівниками.

На остаточному нараді розглядаються, узгоджуються і затверджуються остаточні плани проектних груп, представлені керівниками, а також загальний план компанії.

Можливі деякі зміни плану в ході його реалізації (версії плану).

Можливо ознайомлення з планом зацікавлених сторін і їх участь в плануванні.

Терміни випуску нових версій підсистем можуть не збігатися, але до цього треба прагнути.

Номер планової версії збільшує як мінімум десятковий знак на 1 від попереднього номера версії.

Команда розбиває час роботи над проектом на рівні відрізки - спринти. Спринт може тривати і день, і місяць.

Оскільки все спринти однакові за тривалістю, в роботі команди з'являється ритм. Ритм - важливий аспект методології.

Спринт складається з чотирьох послідовних етапів.

- Планування. Команда перевіряє завдання в беклоге і вибирає найбільш пріоритетні. На спринт беруть стільки завдань, скільки встигнуть зробити.

- Виконання. В ідеальній команді фахівці працюють паралельно: поки програміст створює код, тестувальник пише до нього тести, а технічний письменник - документацію.

- Реліз. Команда представляє результати своєї роботи світу. До моменту кожного релізу продукт повинен бути працездатним, корисним для користувача і більш досконалим, ніж до спринту. Більше про ці вимоги я написав в статті про Agile.

- Ретроспектива. Команда обговорює спринт і виникли проблеми. Всі разом думають, як поліпшити роботу і зробити в наступному спринті більше.

В кінці спринту недороблені завдання йдуть назад в беклог. Чи потрібно їх доробляти і коли, визначають на етапі планування наступного спринту.

Проект ділять на ітерації, але вони можуть бути будь-якої довжини. Ритмічність Kanban не наказує.

Етапи спринту:

-Планування.

-Виконання.

-Реліз.

-Ретроспектіва.

У Kanban один етапи не зав'язані на інші. Настають вони, коли вирішить команда. Наприклад, реліз - по вівторках, планування - коли закінчилися завдання, ретроспективи - кожен останній четвер місяця, а на тлі всього цього безперервно йде розробка.

Оскільки виражених спринтів немає, з'являються особливості:

- нові завдання додають в будь-який час. Якщо потрібно терміново щось зробити, команда не чекає наступного спринту;

- завдання залишається в роботі скільки завгодно довго, поки команда не закінчить її або не скасує.

## Попереднє планування

Мета - відбір завдань-претендентів для включення в план на період.

Завдання:

* + створення нових завдань відповідно до цілей і завдань, поставлених керівництвом;
  + вибір завдань з черги за пріоритетом, цілям і політичних мотивів;
  + визначення пов'язаних з ними завдань (які потребують попереднього рішення, що піддаються впливу, близьких, які забезпечують);
  + груба оцінка масштабу і ризиків;
  + розподіл обраних завдань за виконавцями для ескізного проектування;
  + узгодження між групами;
  + збереження в домені планів.

Учасники процесу:

* + керівництво;
  + керівники проектів;
  + аналітики.

Артефакти:

* + нові завдання (при наявності);
  + попередній план.

Інструмент автоматизації:

* + Github Project;
  + Trello;

Критерії якості:

* + відповідність поставленим цілям і завданням;
  + облік пріоритетних завдань;
  + загальний масштаб (для всіх груп) не перевищує задані терміни;
  + узгодженість між групами.

Зміст плану (Github Project):

* назва і атрибути завдань.
* пріоритет, масштаб і ризик по кожній задачі і загальний;
* послідовність рішення.

Стан:

* в розробці;
* на погодженні; на узгодженні;
* затверджений.

Безпосереднє формування планів груп виробляють керівники груп. Контроль і координацію робіт здійснює тех.діректор.

## Ескізне проектування

Мета - вибір варіантів рішень і уточнення масштабу і ризиків завдань з попереднього плану.

Завдання:

* + розробка варіантів вирішення завдань;
  + оцінка масштабу і ризиків за варіантами;
  + узгодження між групами;
  + узгодження із зацікавленими сторонами (при необхідності);
  + збереження в домені планів.

Учасники процесу:

* + керівники проектів;
  + аналітики;
  + керівництво;
  + зацікавлені сторони (при необхідності).

Артефакти:

* + ескізні проекти варіантів рішень з оцінками масштабу і ризиків;
  + уточнені оцінки попереднього плану.

Інструмент автоматизації:

* + GitHub Projects;
  + Trello;

Критерії якості:

* + зрозуміле уявлення ескізів для всіх учасників;
  + реалізація та тестової;
  + прийнятна точність оцінки масштабу і ризиків.

Стан:

* в розробці;
* на погодженні; на узгодженні;
* затверджений.

Відповідальні за ескізне проектування - керівники груп. Контроль і координацію робіт здійснює тех.діректор.

## Детальне планування

Мета - отримання зведеного плану робіт із зазначенням кінцевої дати, деталізацією по днях і виконавцям.

Завдання:

* + остаточне визначення переліку планових завдань (варіантів рішень);
  + розподіл завдань по групах і виконавцям;
  + уточнення кінцевих термінів з урахуванням масштабу, ризиків і паралельної роботи виконавців і груп, визначення віх (контрольних точок) плану;
  + узгодження між групами;
  + збереження в домені планів.

Учасники процесу:

* + керівництво;
  + керівники проектів;
  + аналітики.

артефакти:

* + остаточний детальний план.

Інструмент автоматизації:

* + Trello;
  + GitHub Projects;

Критерії якості:

* + відповідність часткою витрат часу типів завдань прийнятим нормативам;
  + рівномірність завантаження виконавців;
  + прийнятний масштаб і кінцевий термін;
  + облік ризиків;
  + узгодженість між групами.

Зміст плану (GitHub):

* назва і атрибути завдань (посилання .md)
* пріоритет, трудовитрати і ризик по кожній задачі і загальний;
* посилання на ескізні проекти;
* виконавці;
* послідовність рішення.

Стан:

* в розробці;
* на погодженні; на узгодженні;
* затверджений.

Безпосереднє формування планів груп виробляють керівники груп. Контроль і координацію робіт здійснює тех.діректор. Затверджується директором на нараді, формально - після нього.

# Виконання плану

## Загальний процес

Мета - рішення планових завдань в задані терміни і з заданою якістю.

Завдання:

1. проектування;
2. Розробка;
3. тестування;
4. документування;
5. архівування;
6. Передача замовнику;
7. Впровадження у нового замовника.

Учасники процесу:

* + Керівництво - контроль;
  + керівники проектів - організація виконання, виконання, прийом і відмітка про виконання;
  + персонал проектів - виконання, здавання, звіт.

Артефакти:

* + вихідний код;
  + програмний продукт (інсталяція);
  + проектна (програмна) документація;
  + експлуатаційна документація;
  + документація стадії впровадження.

Критерії якості:

* + повнота виконання;
  + своєчасність;
  + якість артефактів.

**План на період**

**Загальний план компанії**

План по п / с 1

План по п / с 2

План по системі

ескізні проекти

**домен результатів**

Вихідний код

Програмний продукт

документація

Інструмент автоматизації:

* + Trello
  + GitHub Project

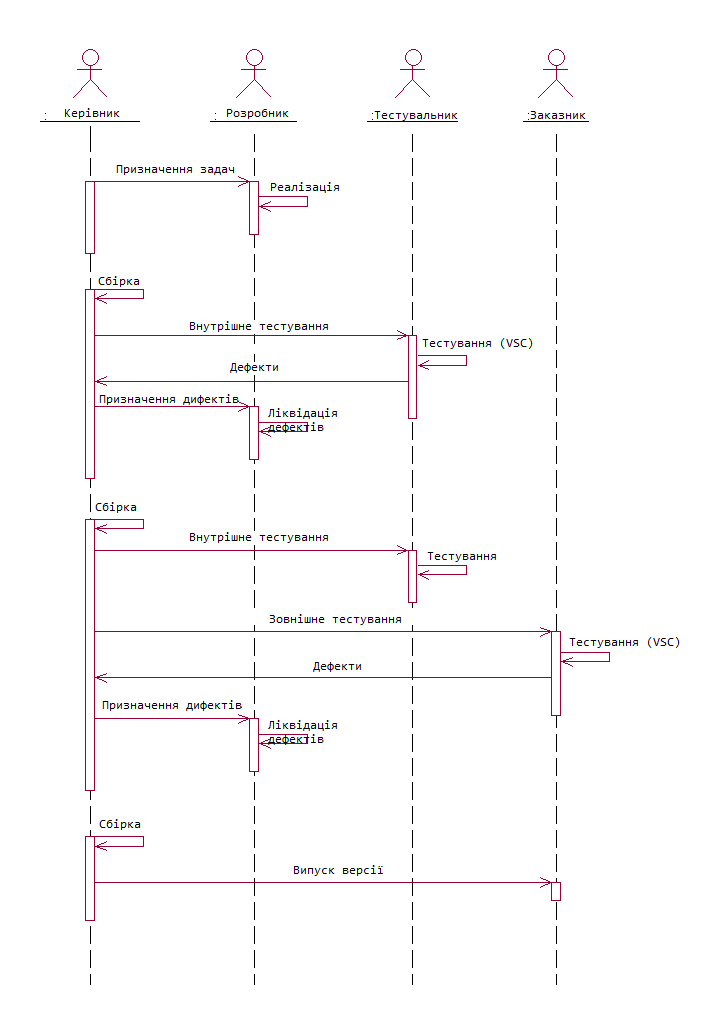
Оціночна частка корисного часу на створення програмного продукту повинна становити:

* + проектування - 25%;
  + розробка - 50%;
  + тестування - 25%.

Виконання плану (% за завданнями) відзначається щодня керівниками проектів за своїми підпроектах. Виконання загальних питань відзначає тех.діректор.

Відмітка про усунення дефектів проводиться в GitHub (з посиланням на версію).

Виконавці інформують безпосередніх керівників щодня. Після закінчення завдання виконавець здає її керівнику, який після приймання зазначає реально витрачений час на цю задачу (для юстування оцінок при майбутньому плануванні та аналізу причин відхилення від тимчасових рамок).



## Проектування

Мета - отримання погодженого з усіма зацікавленими сторонами детального опису рішення щодо майбутньої реалізації завдання для уникнення помилок до початку реалізації і комплексного вирішення проблеми.

Завдання:

* + розробка проекту та його документування;
  + погодження та затвердження;
  + збереження в домені результатів.

Учасники процесу:

* + виконавці;
  + керівники проектів;
  + аналітики;
  + керівництво;
  + зовнішні.

Артефакти:

* + документ-проект.

Інструмент автоматизації:

* + MO Word;
  + Github.

Критерії якості:

* + комплексність, узгодженість з іншими завданнями, повнота;
  + уявлення, зрозумілі для всіх учасників процесу;
  + універсальність, повторне використання коду;
  + реалізація та тестової;
  + надійність і простота;
  + трассируемого по специфікаціям вимог.

Зміст проекту:

* вихідні специфікації (посилання);
* мета і перелік вирішуваних завдань;
* планові терміни, витрати, ризики;
* ескізні проекти (посилання);
* опис рішень (моделі, діаграми, інтерфейс користувача, архітектура);
* похідні завдання (на що вплине);
* обмеження і варіанти тестування.

Стан:

* в розробці;
* на погодженні; на узгодженні;
* затверджений.

За організацію проектування і результат відповідають керівники груп. Контроль і координацію робіт здійснює тех.діректор. Затверджується директором.

## Розробка

Мета - створення (зміна) програми і БД відповідно до проекту і плану.

Завдання:

* + розробка програм і БД;
  + розробка тестів і локальне тестування;
  + документування;
  + здача результатів;
  + збереження в домені результатів.

Учасники процесу:

* + виконавці;
  + керівники проектів.

Артефакти:

* + вихідний і здійсненний код;
  + тести;
  + програмна документація (зміни і деталізація проекту, інструкції).

Інструмент автоматизації:

* + VSC;
  + .NET Fiddle;

Критерії якості:

* + відповідність специфікаціям (в т.ч. системним і нефункціональним вимогам, призначеного для користувача і програмного інтерфейсу);
  + достатню кількість тестів;
  + універсальність, повторне використання коду;
  + достатня повнота документації;
  + достатню кількість коментарів;
  + надійність і простота;
  + мінімум помилок.

стану:

* у черзі;
* на реалізації;
* прийнятий.

За виконання відповідають виконавці. Контроль і координацію робіт здійснюють керівники груп.

## Тестування

Мета - забезпечення якості програмного продукту до передачі замовнику.

завдання:

* + планування, розробка та узгодження тестів з оцінкою тестового покриття;
  + виконання тестів і формування запитів на усунення дефектів;
  + збереження тестів і результатів в домені результатів;
  + прийняття рішення про достатній рівень якості або відправлення на доопрацювання (керівнику проекту).

Учасники процесу:

* + тестувальники;
  + аналітики;
  + керівники проектів.

артефакти:

* + тести;
  + результати (протокол) тестування.

Інструмент автоматизації:

* + VSC;
  + .NET Fiddle;

Критерії якості:

* + мінімум помилок, виявлених замовником;
  + максимальне тестове покриття (за специфікаціями, гілках);
  + повторюваність тестів (регресійні тестування);
  + достовірність результатів тестування.

Види тестування:

* + функціональне;
  + навантажувальний.

## Документування

Мета - створення експлуатаційної документації на програмний продукт і файлів допомоги (тестові приклади, навчальні програми).

Завдання:

* + розробка (зміна) документації;
  + погодження та затвердження;
  + збереження в домені результатів.

Учасники процесу:

* + технічний письменник;
  + аналітики;
  + керівники проектів.

Артефакти:

* + пакет актуальною експлуатаційної документації в складі:
    - загальний опис системи;
    - керівництво користувача;
    - керівництво адміністратора;
    - посібник зі створення звітів користувача;
    - технологічна інструкція;
    - методика впровадження і використання.
  + Help;
  + БД тестових прикладів.

Інструмент автоматизації:

* + HTML;
  + SASS(SCSS);

Критерії якості:

* + актуальність;
  + повнота;
  + ясність;
  + грамотність;
  + відповідність ГОСТам.

За загальне оформлення відповідає технічний письменник, він же пише загальні документи за допомогою аналітиків. За керівництво адміністратора і керівництво по створенню звітів відповідають керівники груп.

## Архівування

Мета - підвищення надійності зберігання та зниження ризиків втрати артефактів за рахунок створення їх резервних копій.

Завдання:

* + створення 3-х резервних копій після кожної ітерації;
  + перевірка копій;
  + фізичне рознесення копій.

Учасники процесу:

* + керівники проектів.

Артефакти:

* + копії бази завдань;
  + копії бази планів;
  + копії бази результатів.

Інструмент автоматизації:

* + VSC, утиліти.

Критерії якості:

* + надійність зберігання;
  + реєстрація в журналі.

За виконання відповідають керівники груп. Контроль і координацію робіт здійснює керівництво.

## Передача замовнику

Мета - передача замовнику результатів роботи (нової версії системи).

Завдання:

* + виставлення нової версії на сайті;
  + повідомлення замовників;
  + відмітка про успішну установки версії у замовника;
  + підписання актів прийому-передачі.

Учасники процесу:

* + керівники проектів;
  + керівництво.

артефакти:

* + виконані код;
  + скрипти для БД;
  + експлуатаційна документація;
  + протоколи внутрішніх випробувань;
  + акт прийому-здачі.

Інструмент автоматизації:

* + GitHub Releases;
  + FTP.

Критерії якості:

* + надійність передачі;
  + швидкість.

За виконання відповідають керівники груп. Контроль і координацію робіт здійснює керівництво.

## Впровадження у нового замовника

Мета - введення системи в експлуатацію в новій організації.

Завдання:

* + попередня установка системи у замовника для ознайомлення;
  + конвертація БД замовника в нашу систему;
  + навчання користувачів і групи впровадження замовника;
  + передача замовнику методики впровадження системи;
  + розробка документації стадії впровадження;
  + проведення попередніх приймально-здавальних випробувань;
  + усунення зауважень по протоколам випробувань;
  + повторні випробування по введенню в дослідну (промислову) експлуатацію;
  + допомога в організації процесу експлуатації, закриття місяця;
  + організація техпідтримки.

Учасники процесу:

* + замовник;
  + група впровадження;
  + керівники проектів;
  + аналітики;
  + керівництво.

Артефакти:

* + план робіт з впровадження системи (замовника);
  + програма і методики випробувань;
  + протоколи випробувань;
  + акти випробувань і введення в експлуатацію;
  + журнал (дослідної) експлуатації;
  + акт прийому-здачі.

Критерії якості:

* + мінімум зауважень під час випробувань;
  + успішна робота користувачів за розрахунковий період.

За виконання відповідають керівники груп. Контроль і координацію робіт здійснює керівництво.

Процес впровадження вимагає більш детального опису.

# Аналіз виконання плану

Мета - аналіз результатів виконання плану попереднього періоду для оцінки якості планування і виконання робіт, обліку результатів для нового планування.

Завдання:

* + загальна оцінка результатів виконання плану (виконаний, не виконано);
  + оцінка фактичних показників масштабу, ризиків, термінів і якості рішення задач;
  + аналіз проблем і виявлення їх причин;
  + прийняття рішень з проблемних питань і завдань (закінчити, відкласти, перенести в новий план, виключити).
  + заохочення успіху і вказівки на недоробки.

Учасники процесу:

* + виконавці - підсумкові результати (звіти) роботи за період;
  + керівники проектів - аналіз виконання планів своїх груп, персоналу;
  + керівництво - загальний аналіз, по групах, персоналу.

Артефакти:

* + фактичні показники виконання планових і позапланових завдань (трудовитрати, терміни, ризик, якість, кількість дефектів);
  + список невирішених завдань із зазначенням причин, пропозиції по ним;
  + список співробітників, що мають бути відмічені (в обидві сторони).

Критерії якості:

* + облік кожного завдання і кожного співробітника;
  + точність оцінок;
  + правильне визначення причин проблем.

Інструмент автоматизації:

* + GitHub Projects (Insights)

Аналіз виконання плану готується напередодні наради по ієрархії підпорядкованості. Результати аналізу обговорюються на спільній нараді.