**Тема. Робота з контрактами**

1. **Типи контрактів**
2. **Оцінка тривалості і вартості контракту**
3. **Захист прав на продукт**
4. **Типи контрактів**

У сучасній розробці є три типи контракту — Fixed Price (Фіксована ціна), Time and Material (Час та матеріали) та Dedicated team (Виділена команда).



Почнемо із найпростішого типу контракту — Fixed price.

## Fixed price

У цьому контракті написано, що ваша компанія має зробити X роботи за суму Y до дати Z. До нього завжди додається специфікація, де докладно описано, як має працювати кожна складова.

Здається, що не може бути нічого простішого, ніж робота з чіткими вимогами. Але на практиці потрапити у всі три параметри XYZ складно. Тому в такий контракт потрібно закладати додаткові грошові та часові ризики. Наприклад, проєкт чистою вартістю $60 000 та тривалістю 5 місяців продається за $100 000 та 8 місяців.

Скільки саме додається зверху, залежить від ризиків, які є у проєкті. Що менше вимог надіслав замовник, то більше ризиків закладе виконавець. Це робиться, щоб не піти в мінус і отримати прибуток із проєкту.

**Варіанти FP-проектів**

**Firm Fixed Price (FFP).**Це договір із жорсткою сумою, яка залишається постійною незалежно від фактичних витрат виконавця. Це найбільш загальний тип контракту для тих випадків, коли обсяг проекту та витрати досить прогнозовані.

**Fixed Price Incentive Fee (FPIF).**Фіксована ціна + винагорода = фіксована оплата плюс бонус за обумовлені показники продуктивності (швидше та краще).

**Fixed Price Award Fee (FPAF).**Фіксована ціна включає винагороду + премія (бонус). Можлива премія визначається заздалегідь та розподіляється на підставі Execution (виконання).

**Fixed Price Economic Price Adjustment (FPEPA).**Контракт із фіксованою ціною та застереженням про можливе коригування ціни, наприклад, через зміну курсу НБУ на момент виплати.

## Завдання

Як думаєш, з якими складнощами зіткнеться проджект, роблячи проєкт за Fixed price контрактом?

###### Відповідь

Замовнику невигідно витрачати на проєкт більше, ніж планувалося. Тому твої дії **будуть обмежені початковими вимогами контракту.**Це позбавляє команду гнучкості. У проєкті можуть виникнути непередбачувані зміни, але через Fixed price тобі все одно доведеться вкладатися в початковий бюджет та терміни — іноді коштом якості роботи.

Також з Fixed price команда **повинна працювати максимально злагоджено**, оскільки встановлені чіткі дедлайни. Якщо не налаштувати процеси заздалегідь, ви просто не встигнете виконати роботу на належному рівні та у потрібний час.

Зрештою, такий контракт **дозволяє працювати лише над зрозумілими та невеликими проєктами**. Чим більше буде невідомих, тим вищий ризик провалу.

Попри те, що Fixed price контракт — це певний скоуп робіт та чіткі дедлайни, завжди настає момент, коли замовник просить додати щось ще. І це нормально. Неможливо заздалегідь продумати та внести у контракт весь потенційний функціонал. Але це не означає, що замовник може змінювати будь-що і коли захоче. Зміни у проєкті мають обговорюватися. Підкреслюй, що за додаткову фічу потрібно доплатити. Після цього оцінюй час на роботу та подовжуй контракт на потрібний команді період.

Усі зміни у проєкті повинні додатково засвідчуватися: проджект складає та підписує додатковий контракт. Це довга рутинна робота, тому по Fixed price зазвичай ведуться передбачувані й зрозумілі в розробці проєкти. Ті, що команда зможе зібрати за інструкцією, як стіл з IKEA.

Проєкт зі зрозумілими вимогами часто розробляється за моделлю **Waterfall**.

**Отже.** Fixed price контракт слід укладати, якщо всі вимоги зрозумілі, дедлайн і бюджети встановлені, а сам проєкт можна повністю розпланувати ще до початку роботи.

Але іноді такої впевненості нема. Це буває зі складними комплексними проєктами, або коли запит замовника обмежується абстрактною ідеєю. Незрозуміло, скільки часу займе проєкт, скільки буде фіч і скільки вони коштуватимуть. Тоді краще укласти контракт Time and Material.

**2. Time and material (T&M)**

За контрактом Time and material замовник платить за час, витрачений на виконання завдання, а не за конкретний результат. Принцип роботи T&M схожий на Uber — що далі їдеш, то більше платиш. Якщо на дорозі пробки, вартість подорожі зміниться. Діставшись однієї точки, можна поставити нову і продовжити шлях.

На відміну від Fixed price, у такому контракті не прописані конкретні скоуп та вартість робіт. Команда сама оцінює, скільки часу та ресурсів займе кожне нове завдання. Іноді в T&M контрактах вказується орієнтовна оцінка термінів та вартості, але це не є обов'язковою умовою.

Щоб краще зрозуміти як це працює, розгляньмо T&M на прикладі. Замовник просить зробити функцію викривлення звуку в аудіоредакторі. Отримавши вимоги, проджект з командою оцінюють роботу в 40 годин і $1600.

Попри ці орієнтири, кількість годин і вартість можуть змінитися у процесі розробки. Наприклад, ви зрозумієте, що потрібно взяти більше часу на вивчення фічі або додаткові тести. Ти озвучуєш замовнику ці умови, після чого він може погодитись і дати добро, щось поміняти або взагалі прибрати фічу зі скоупу.

Так, узгоджуючи фічу за фічею, ви працюєте ітераціями. Раз на тиждень або з іншою періодичністю команда обговорює та оцінює завдання, які надіслав замовник. А наприкінці місяця ти підраховуєш витрачені години та надсилаєш клієнту інвойс на оплату.

Time and Material контракти зазвичай укладаються для проєктів по гнучких методологіях. Найчастіше це **Scrum**. Ця методологія використовується, коли замовник не має специфікації й незрозуміло, як робити проєкт. А ще якщо проєкт зовсім новий та не має аналогів.

Наприклад, зробити клон соцмережі Twitter можна і за Waterfall — у нього зрозуміла функціональність. Але написати двигун рекомендацій, як у Tiktok, — вже інше завдання. Відразу дуже складно сказати, що знаходиться «під капотом». У цій фічі багато невідомих змінних — краще вибрати T&M та Scrum.

Scrum дозволяє швидко реагувати на зміни та відстежувати проміжні результати. Тому більшість сучасних проєктів створюються саме за цією методологією.

По структурі Scrum схожий на стислий до двох тижнів Waterfall. Він відрізняється тим, що замовник отримує результат у вигляді фічі, що працює, вже в кінці першого спринту.

## Завдання

Як думаєш, чому контракт Time and Material та практика Scrum добре працюють разом?

###### Відповідь

Scrum — досить гнучка методологія. З нею **легко опрацьовувати ризики та змінювати напрямок роботи**, якщо так захоче замовник або вимагатимуть обставини. З контрактом Time and Material ти передбачаєш ці ризики й для команди, одразу домовляючись про **погодинну оплату**.

Уяви, що розробляєш програму з можливістю відстежувати об'єкти в кадрі та створювати їх 3D копії у реальному часі. У середині проєкту замовник дізнається, що хтось із конкурентів також впроваджує цю фічу. Потрібно терміново переглядати поточне рішення.

Зі Scrum ти швидко відреагуєш на зміни: зміниш пріоритети завданням в беклозі та внесеш корективи в новий спринт. Але зміняться ціна та час розробки. Саме тому ви укладаєте контракт Time and Material — щоб замовник міг змінювати хід проєкту, а ти — виставляти інвойси за фактом та отримувати оплату за результатами спринтів.

Ми розібралися з двома основними типами контрактів — Fixed Price та Time and Material. Їх обирають, коли замовник готовий повністю чи частково довірити проєкт аутсорс-компанії.

Але у клієнта може бути й своя команда, в якій просто не вистачає спеціалістів. У такому разі слід укладати контракт Dedicated team.

**Dedicated team**

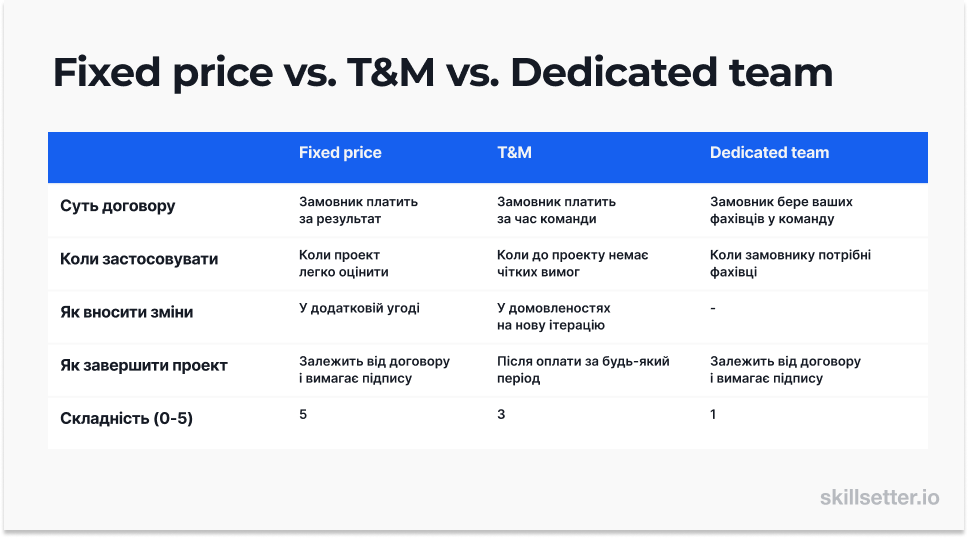
DT — найпростіший тип контракту, за яким клієнт отримує у розпорядження фахівців з вашого штату і платить за час їхньої роботи. Виділена команда повністю фокусується на розробці, а клієнт контролює сам процес.

Контракт Dedicated team характерний для тимчасових проєктів у великих компаніях. Він зручний, тому що замовнику не потрібно брати людину в штат, оплачувати її медстраховку та нести інші витрати. Тимчасовий проєкт можуть закрити будь-якої миті, а розірвати контракт зі спеціалістом з аутсорсу набагато легше, ніж зі штатним співробітником.

Уяви, що замовнику потрібно реалізувати стрічку з порадами з фотографій у його фоторедакторі. Завдання вирішується менше, ніж за місяць, але в його штаті немає бекендера. Замовнику буде легше укласти контракт на кілька тижнів з аутсорс-компанією, а не наймати та навчати нового спеціаліста.

Проджект-менеджери рідко працюють із Dedicated team контрактами. Все, що потрібно від проджекту — це проконтролювати формальності. Запитати замовника, чи все йде добре, дізнатися у співробітника, коли він збирається у відпустку, виставити інвойс. Тому контрактами Dedicated team часто займаються менеджери з продажу, ресурсні менеджери або акаунт-менеджери.

Підіб'ємо підсумок і порівняємо всі три типи контрактів: Dedicated team, Fixed price і Time and Material.



## Рев'ю контрактів

Перше, що потрібно зробити у новому проєкті — перевірити, які документи підписано. Різні люди можуть дати різну інформацію: клієнт скаже одне, менеджер з продажу друге, а в контракті буде прописано третє. Орієнтуватися завжди потрібно на контракт.

У кожної компанії є свій шаблон контракту, але умови в однотипних контрактах майже завжди ті самі. Перевіряй, наскільки чітко вони прописані, і чи вказані всі потрібні пункти. Таблиця з умовами у документах.

|  |  |  |  |
| --- | --- | --- | --- |
| **Contract clause** | **Fixed price** | **Time and Material** | **Dedicated team** |
| Invoicing and Payment Інвойси і оплата | **+** | **+** | **+** |
| Termination Розірвання договору | **+** | **+** | **+** |
| Overtime Понаднормові | **+** | **+** | **+** |
| Buy-out clause Примітки про викуп | **+** | **+** | **+** |
| Intellectual Property Інтелектуальна власність | **+** | **+** | **+** |
| List of services (design, produce, develop, test, launch, etc.) Перелік надаваємих послуг (дизайн, створення, розробка, тестуванння, запуск та ін) | **+** | **+** | **-** |
| Process of evaluation, development and reporting Процес оцінки, розробки та звітіів | **+** | **+** | **-** |
| People working on the project: specialization, rate,  full time or part time Люди, що працюють над проектом: спеціалізація, ставка, повна або часткова зайнятість | **-** | **+** | **+** |
| Replacement for employees Заміщення персоналу | **-** | **+** | **+** |
| Trial period Випробувальний термін | **-** | **+** | **+** |
| The term of a contract Строк контракту | **-** | **-** | **+** |
| Contract extension and a contract renewal Продовження контракту | **+** | **+** | **+** |
| Sick leaves, holidays and days off Лікарняні листи, свята та вихідні | **-** | **-** | **+** |
| Scope of work Обсяг робіт | **+** | **-** | **-** |
| Terms of work Умови роботт | **+** | **-** | **-** |
| Budget Бюджет | **+** | **-** | **-** |
| Documents supporting Scope of work, Terms of work  and Budget Документи, що підтверджують об’єм робіт, умови роботи і бюджет | **+** | **-** | **-** |
| Assumptions Пропозиції | **+** | **-** | **-** |
| How to work with changes Як працювати зі змінами | **+** | **+** | **-** |
| Delivery: how, where and when your company delivers  the result of work Постачання: як, куди і коли компанія постачає результати роботи | **+** | **+** | **-** |
| Acceptance: how, where and when a customer accepts  the work and how your company reacts Приймання: як, де і коли клієнт приймає результати роботи, і що компанія роботь у відповідь | **+** | **+** | **-** |
| Bugs: what is a minor and what is a major problem Баги (помилки): що є несуттєвою проблемою, а що — важливою | **+** | **+** | **-** |
| Warranty Гарантія | **+** | **+** | **-** |

Який тип це контракту для Filmy?

Для Filmy вибрано контракт Time and Material. В таких випадках зазвичай використовуються гнучкі методології, як Scrum.

Як сказав Вілл, Джон мало розуміється на розробці. Він **не висуває конкретних вимог** — тобі доведеться вигадувати їх разом із замовником. Тому представники CreatorTY домовилися, що замовник платить за час, який команда витрачає на виконання завдань.

Джон не буде чекати на точний обсяг робіт до певного дедлайну, як це буває у Fixed price. Він згоден міняти проєкт по ходу, додавати та прибирати фічі.

До того ж Джон не просить окремих фахівців собі в команду. Має запит на цілий проєкт, тому Dedicated team теж не підходить.

## 2. Оцінка тривалості і вартості контракту

В аутсорс-компаніях з усталеними процесами проджект супроводжує продаж проєкту разом з менеджером з продажу. Він з'ясовує у клієнта, що потрібно побудувати, а потім оцінює скоуп завдань разом із розробниками.

Етапи роботи:

Визначення переліку **задач, їх декомпозиція і побудова ієрархії для всього проекту**.

О**пис кожної задачі**. Можна сильно не заглиблюватися і писати щось на кшталт: «Логін та реєстрація у додатку проводяться за e-mail, профілем на Facebook та за номером телефону».

Визначення **стеку технологій**, якими користуватиметеся в проєкті. Мови програмування, фреймворки — все це. Зазвичай це робить тімлід.

Визначення **складу команди та рейти**. Вам потрібно чітко розуміти, які фахівці працюватимуть над проєктом і скільки коштує їхня робота.

Встановлення всіх необхідних **видів тестувань**. Вони займають багато часу та бюджету. У цьому пункті краще не помилятися та обміркувати все ретельно. Серед тестів основними є: UI тести, юніт тести, тести на безпеку, продуктивність тощо. А потім перерахувати всі пристрої, операційні системи, браузери, що підтримуються. Треба зазначати все, **що клієнт має закупити сам,** якщо якихось специфічних девайсів у вас немає. Наприклад, у компанії є 30 пристроїв, але замовнику потрібно протестувати на певному, якого у нас немає. Прописуємо це в оцінці.

Тривалість робіт бажано **візуалізувати** та показати замовнику у вигляді гант-чарта або розбивки по фазах. І обов'язково прикріпи **всі можливі документи**, на основі яких ви оцінювали.

Закласти **ризики та припущення**. Наприклад, ти знаєш, що технологію, яку замовляє клієнт, ніхто у компанії ще не розробляв. На її вивчення явно піде додатковий час. Розв'язуєш питання тим, що закладаєш більше ресурсів.

Ось чек лист проекту

Оцінка проекту

1. Декомпозиція на задачі (WBS)

2. Короткий опис задач

Наприклад, логін/реєстрація по e-mail, профіль у Facebook, номер телефону.

3. Стек технологій

Мови програмування, фреймворки, системи управління базами даних, та ін. д.

4. Рейти, узгоджені з сейлом

5. Всі необхідні види тестування.

UI тести, юніт-тести, тести на продуктивність і безпеку. А також перераховані всі підтримувані пристрої, операційні системи та браузери.

6. Закуплені за рахунок клієнта пристрої

Наприклад, у компанії є 30 пристроїв, але замовлення потрібно протестувати на якому-то визначеному, якого в компанії немає.

7. Ризики і пропозиції

Все, що забезпечує швидкість роботи команди.

8. Склад команди

Спеціалісти, їх рівень експертизи та кількість.

9. Тривалість проекту

Бажано з візуалізацією у вигляді гант-чарта або розбивки по фазам.

10. Посилання на документи, на основі яких ви провели оцінку.

Документи, що підтверджують усі попередні пункти.

Опрацювати контракт від Джона (додаток А в кінці документа).

Ця корисна інформація знадобиться, коли почнеш самостійно оцінювати проєкти ще до того, як вони потраплять у продакшн. Як правило, це довіряють лише досвідченим проджект-менеджерам.

Завдання. 1. Який тип це контракту для нашого додатку? Обгрунтувати.

Завдання. 2. Визначити тривалість контракту для виконання робіт по створенню клона додатку для замовлення таксі.

Таблиця 1

Вихідні параметри процесу розробки мобільного додатку для замовлення таксі

|  |  |
| --- | --- |
| І. Процес пасажира | ІІ. Процес таксиста |
| 1 | 2 |
| Ціль пасажира: попасти з точки А (адреса відправлення) в точку Б (адреса призначення). | Ціль таксиста: виконати замовлення вчасно та якісно та отримати оплату. |
| *Функції продукту (5 функцій та 18 задач):* | *Функції продукту:* |
| 1. Замовлення таксі;    1. вказати текстом в додатку адресу А;    2. вказати текстом в додатку адресу Б;    3. за допомогою GPS в телефоні автоматично визначити адрес А.    4. на карті вказати адрес В. | 1. Обрати та прийняти замовлення:    1. бачити замовлення списком;    2. сортувати замовлення за критерієм час-ціна    3. обрати замовлення;    4. відправити повідомлення про прийняття замовлення. |
| 1. Вихід та очікування таксі:    1. отримання повідомлення про прибуття таксі;    2. відзначити повідомлення як отримане;   2.3. перегляд на карті місце перебування таксі;  2.4. прийняти дзвінок від таксиста. | 1. Забрати клієнта з точки А:   2.1. бачити адресу відправника;  2.2. прокласти маршрут за допомогою навігації до точки А.  2.3. відправити повідомлення про прибуття.  2.4. відправити дзвінок про прибуття. |
| 1. Посадка в таксі;   3.1. в додатку побачити колір, марку та номер таксі;  3.2. подивитися на карті місце знаходження таксі;  3.3. подивитися на екрані вихідні дані замовлення (км, вартість). | 3) Посадка в таксі:  3.1. включити голосове повідомлення по початок виконання замовлення;  3.2. включити автовідповідач про час прибуття |

Продовження табл.1

|  |  |
| --- | --- |
| 1 | 2 |
| 1. Прибуття в місце призначення.   4.1. перегляд руху таксі на карті;  4.2. огляд заторів за маршрутом;  4.3. коригування маршруту;  4.4. перегляд часу прибуття за адресою Б. | 3) Довести клієнта до точки В:  3.1. бачити адресу призначення на карті;  3.2. прокласти маршрут за допомогою навігації до точки Б. |
| 1. Оплата за послугу (готівкою або карткою з додатку).   5.1. оплата готівкою з електронним чеком;  5.2. оплата карткою через додаток;  5.3. оплача через QR-код. | 4) Прийняти оплату:  4.1. прийняти готівку;  4.2. прийняти оплату на картку через додаток;  4.3. прийняти оплату через QR-код. |

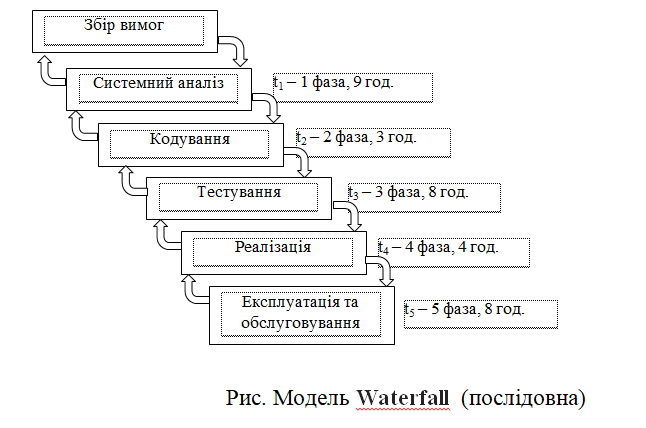


Рис.2. Модель Waterfall (послідовна)

Послідовна характеризується тим, що кожна наступна операція (задача) починається тільки після її повного завершення на попередній операції (стадії процесу або задачі). Тривалість технологічного циклу визначається за формулою:

|  |  |
| --- | --- |
|  | (1) |

де *n* – кількість задач у процесі, 18;

*ti***–** трудомісткість i-ї операції (задачі), нормо-год. (див. дані рис.2) ;

*сi*- кількість робочих місць на операції або членів проектної команди на даній операцій, по 1 особі.

За формулою 1 визначаємо тривалість технологічного циклу

*Тпос = 18·(9/1+3/1+8/1+4/1+8/1) = 576 год.*

З Waterfall легко контролювати розробку, тому що всі етапи йдуть строго послідовно. Легше ставити дедлайни, формувати таски та ін. Така модель знижує ризики, оскільки проєкт детально опрацьовується ще на початку реалізації.

Тпос = 576 год.

480

480

480

480

480

480



480

480

480

480

480

480

427

394

326

140

72

144

144

54=18\*3

162=18\*9

Номер

операції

1 (t1=9)

2 (t2=3)

3 (t3=8)

4 (t4=4)

5 (t5=8)

Час, год.

Рис.3. Послідовна модель планування тривалості проекту

Паралельно-послідовна модель характеризується тим, що задачі передаються на наступну стадію процесу (спринту) таким чином, щоб забезпечити безперервність всього процесу розробки програмного продукту. Тривалість технологічного циклу розраховується за формулою:

|  |  |
| --- | --- |
|  | (2) |

де, *n* – кількість задач у процесі, 18; кількість проміжних етапів (епік або спринтів), з яких складається розробка: 18/3 = 6 задач в епіку або спринт (етап);

*р* – кількість задач у етапі процесу або спринті, 3 (визначено командою експертним шляхом на основі оцінки сторі поінт) .

** - сума тривалості всіх суміжних операцій з виконання задач (етапів або спринтів) з мінімальними циклами.

За формулою 2 визначаємо тривалість технологічного циклу проекту при паралельно-послідовному підході:

*Тпп =* ***18·(9/1+3/1+8/1+4/1+8/1)****-(18-3)(3/1+3/1+4/1+4/1)=366 год.*

При запарелелюванні процесів, графік побудови технологічного процесу розробки проекту буде мати наступний вигляд (рис.3).

Тпп = 366 год.

Номер

операції

1 (t1=9)

2 (t2=3)

3 (t3=8)

4 (t4=4)

5 (t5=8)

Час, год.

27

27

27

27

27

162/6=27 год.

9

9

9

9

9

9

24

24

24

24

24

24

12

12

12

12

12

12

24

24

24

24

24

24

Рис.. Паралельно-послідовна модель планування тривалості проекту

В таблиці Додатку, лист 5 наведено сітьовий графік тривалості проекту.



На слайді видно, що паралельно із розробкою першої фічі дизайнер малює UI для другої. А поки розробники роблять другу фічу, команда QA тестує першу. Ні дизайн другої фічі, ні тестування першої в критичний шлях не увійдуть, тому що вони коротші за розробку і виконуватимуться паралельно.

Але зверни увагу на послідовність дизайну та розробки для першої фічі та розробки та тестування — для третьої. Нескладно здогадатися, що не можна розпочати розробку першої фічі без дизайну та завершити проєкт без тестування останньої.

Отже, оптимізація процесів з розробки програмного забезпечення за рахунок зменшення часу технологічного процесу шляхом запаралелювання призведе до зменшення тривалості циклу розробки проекту на 209 год. (575-366) та усунення часових розривів – це виявлення тимчасових розривів у бізнес-процесах і усунення часу простою або перевантаження.

Тільки після процесу аналітичного планування можна переходити до вибору інструментів роботи по проекту.

Зміст та структура документації на етапі ініціалізації та планування проекту також має містити:

* 1. vision and scope document (документ про загальне бачення та зміст продукту).
  2. business requirements document (BRD) (документ, що описує бізнес-процеси, в тому числі за допомогою графіків та рисунків та деталізує бізнес-вимоги клієнта для внутрішнього подальшого користування).
  3. software requirements specification (SRS) (специфікація вимог до програмного забезпечення на основі потреб клієнта, споживачів та інших стейкхолдерів).
  4. functional requirements specification (FRS) (специфікація функціональних вимог (FRS), як правило для MVP. Більше технічний документ, до якого долучаються розробники.

В таблиці наведені основні характеристики документації та її зміст на етапі ініціалізації та планування проекту. Більшість таких документів розробляється для невеликих проектів і є базовими концептуальними документами, які в подальшому трансформуються в Jira та оформлюються в Confluence.

Таблиця 1.7

Формування та зміст документації на етапі ініціалізації та планування проекту

|  |  |  |  |
| --- | --- | --- | --- |
|  | Документ бізнес-вимог (BRD) | Специфікації вимог до програмного забезпечення (SRS) | Специфікації функціональних вимог (FRS) |
| Назва |  | Документ вимог до продукту (PRD) і специфікація системних вимог | Документ з функціональними специфікаціями (FSD), Документ зі специфікаціями продукту (PSD), функціональні характеристики (FS) |
| Створений | Бізнес-аналітик | Бізнес/системний аналітик | Бізнес/системний аналітик/керівники впровадження |
| Містить | Високий рівень вимог бізнесу та зацікавлених сторін | Детальні функціональні вимоги, нефункціональні вимоги та варіанти використання | Детальні функціональні вимоги, потік даних і діаграми UML |
| Використаний | Керівництво вищої та середньої ланки | Менеджери проектів, МСП (експерти по тематиці), технічний керівник та керівник впровадження | Технічні керівники, команди розробників та команди з тестування. |
| Підготовлено в | Фаза ініціації | Фаза планування | Фаза планування |
| Відповіді | «Чому?» вимоги виконуються | «Які вимоги мають бути виконані, щоб задовольнити потреби бізнесу | «Як саме» очікується, що система буде функціонувати |
| Приклад | Підвищити ефективність, відстежуючи роботу співробітників в офісі | Пропоноване програмне забезпечення міститиме такі модулі: Вхід, Адміністратор, Співробітник та Звітність | Модуль входу  Введіть usemame.  Введіть пароль  Кнопка «Надіслати» |

Наступний напрям робіт на цьому етапі - формування статуту проекту – невеликий документ на декілька сторінок (концепція описова або таблична), що описує про що проект, що буде створено цим проектом, тобто список очікуваних результатів проекту. Це узгодження між усіма зацікавленими сторонами проекту. Це керівництво всією діяльністю по проекту та роботи зі стейкхолдерами проекту.

Таблиця 1.8

Елементи Статуту

|  |
| --- |
| Статут |
| Мета або обґрунтування проекту |
| Вимірювані цілі проекту та відповідні критерії успіху |
| Вимоги високого рівня |
| Опис проекту на високому рівні, характеристики продукції |
| Зведений графік етапів |
| Ризики високого рівня |
| Зведений бюджет |
| Вимоги до затвердження проекту (що означає успіх, хто вирішує його, хто підписує) |
| Призначений керівник проекту, відповідальність та рівень повноважень |
| Ім'я та відповідальність особи (осіб), що уповноважує статут проекту |
| Список зацікавлених сторін |

В таблиці Додатку, лист 3 наведено приблизний розрахунок бюджету і тривалості проекту.

**3. Захист прав на продукт**

Права інтелектуальної власності на мобільні додатки від продуктової ІТ-компанії Генезіс:

<https://www.youtube.com/live/8N9V53kAl0c>

У випадку з мобільними додатками як ринковими продуктами в продуктовій ІТ-компанії, важливо вжити заходів для захисту таких аспектів:

**1. Назва мобільного додатку:**

* **Реєстрація торгової марки:** Рекомендується зареєструвати назву мобільного додатку як торгову марку в країнах, де планується його випуск. Це допоможе захистити унікальну назву вашого продукту від незаконного використання конкурентами.
* **Авторське право:** Назва мобільного додатку також може підлягати авторському праву, якщо вона є оригінальною та креативною.

**2. Дизайн мобільного додатку:**

* **Авторське право:** Захистіть дизайн інтерфейсу користувача (UI) та інші візуальні елементи мобільного додатку авторським правом. Це допоможе запобігти копіюванню вашого дизайну конкурентами.
* **Патенти:** Деякі аспекти дизайну мобільного додатку, наприклад, інноваційні елементи інтерфейсу чи функціональні можливості, можуть підлягати патентуванню.

**3. Код мобільного додатку:**

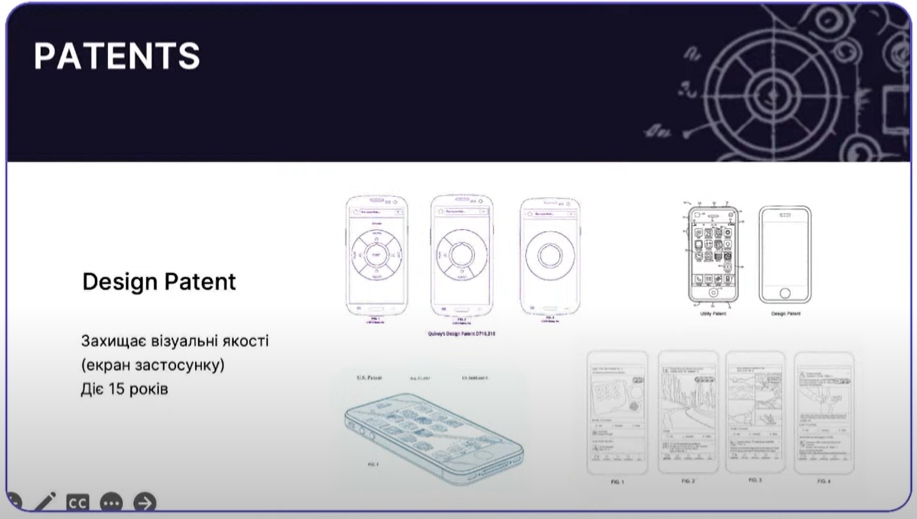
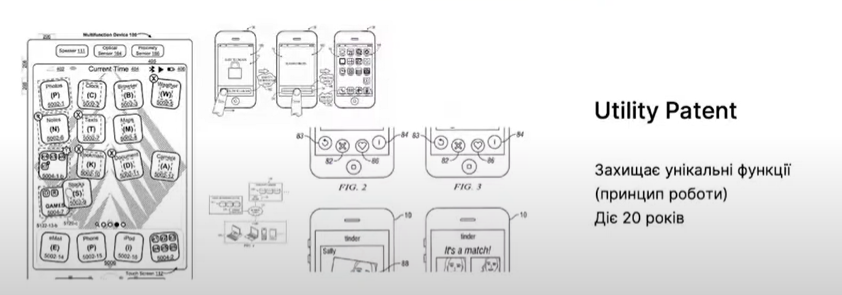
* **Авторське право:** Захистіть код мобільного додатку авторським правом. Це допоможе запобігти незаконному копіюванню та розповсюдженню вашого коду.
* **Ліцензування:** Ви можете ліцензувати код мобільного додатку іншим компаніям або розробникам за допомогою відкритих ліцензій або платних ліцензійних угод.

Захист мобільної гри як продукту продуктової ІТ-компанії потребує комплексного підходу, який охоплює різні аспекти інтелектуальної власності. Ось деякі ключові моменти, які слід врахувати:

**1. Авторське право:**

* **Ігрові механіки:** Оригінальні ігрові механіки та концепції можуть бути захищені авторським правом. Це може включати правила гри, системи підрахунку очків, віртуальні світи та персонажі.
* **Дизайн інтерфейсу користувача (UI):** Візуальні елементи інтерфейсу користувача (UI), такі як іконки, меню та графічні елементи, можуть підлягати авторському праву, якщо вони є оригінальними та креативними.
* **Художні твори:** Музика, звукові ефекти, зображення та інші художні твори, що використовуються в мобільній грі, можуть бути захищені авторським правом.
* **Літературні твори:** Текст, що використовується в мобільній грі, наприклад, діалоги, описи та сюжетні лінії, може підлягати авторському праву.

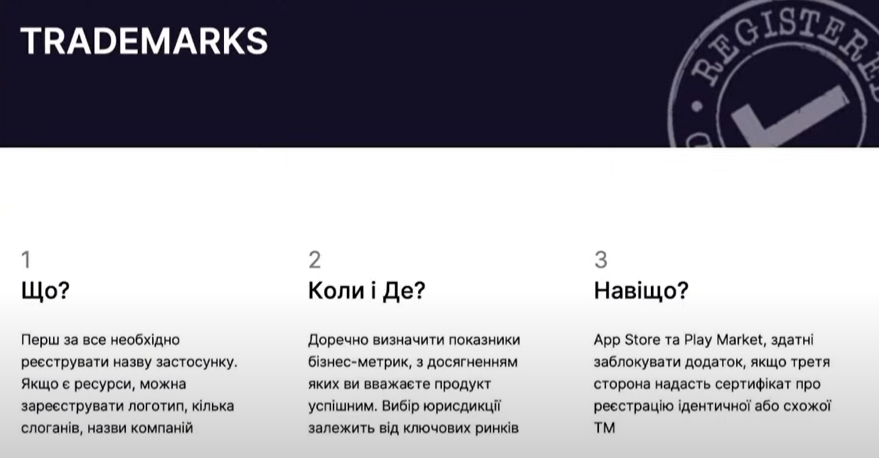
**3. Патенти: Не є поширеним явище в ІТ**

* **Інноваційні ігрові механіки:** Деякі аспекти ігрової механіки, наприклад, унікальні алгоритми або системи, можуть підлягати патентуванню.
* **Технологічні рішення:** Використовувані в мобільній грі технологічні рішення, наприклад, системи штучного інтелекту або алгоритми стиснення даних, можуть підлягати патентуванню.
* 
* По суті перший вид патенту захищає UI, другий – UX частину, наприклад Facebook та Tinder мають такий вид патенту на багато своїх механік та програм.
* 

**2. Торгова марка:**

* **Назва гри:** Унікальна назва мобільної гри може бути зареєстрована як торгова марка, щоб захистити її від незаконного використання конкурентами.
* **Логотип:** Логотип мобільної гри також може бути зареєстрований як торгова марка.

**Реєстрація однієї ТМ по одному класу (продукти розподілені по класам), наприклад назва гри приблизко коштує від 4000 грн. ДП "Укрпатент" реєструє ТМ і видає вам свідоцтво про реєстрацію. Весь процес реєстрації ТМ може зайняти від 16 до 20 місяців.**

****

**Авторське право на код мобільного додатку, як правило, належить тому, хто його написав.**

Це означає, що **якщо ІТ-компанія-виконавець розробила код для замовника, то авторське право на цей код зазвичай належить ІТ-компанії.**

**Однак, існують нюанси, які слід врахувати:**

* **Умови договору:** Найважливішим фактором, який визначає, кому належить авторське право, є **умови договору, укладеного між замовником та ІТ-компанією.**
  + У договорі повинно бути чітко прописано, хто володіє авторськими правами на код, а також інші права інтелектуальної власності, пов'язані з мобільним додатком.
  + Якщо в договорі не чітко визначено власника авторського права, то воно може належати ІТ-компанії, замовнику або їм обом спільно.
* **Хто написав код:** Якщо код був написаний **штатними співробітниками ІТ-компанії**, то авторське право зазвичай належить ІТ-компанії.
  + Якщо ж код був написаний **фрілансерами або сторонніми розробниками**, то авторське право може належати їм, якщо не було укладено спеціальної угоди про передачу авторських прав.
* **Внесок замовника:** Якщо замовник **активно брав участь у розробці коду**, наприклад, надавав ідеї, технічні специфікації або тестував код, то він може мати **часткове авторське право** на код.
* **Оплата:** Оплата за розробку коду **не гарантує автоматичного переходу авторських прав до замовника.**

**Чи може розробник, який володіє авторським правом на код, використовувати його для розробки інших продуктів?**

**Відповідь залежить від декількох факторів:**

**1. Умови ліцензії:**

* Якщо розробник **використовує відкриту ліцензію**, наприклад, GPL або MIT, то він може вільно використовувати код для розробки інших продуктів, але при цьому він **повинен дотримуватися умов ліцензії**.
  + Це може включати вимоги щодо надання вихідного коду, згадування авторства або надання інших ліцензій на свої похідні продукти.
* Якщо розробник **використовує закриту ліцензію**, то він може використовувати код для розробки інших продуктів **лише з дозволу власника ліцензії**.
  + Власник ліцензії може надати дозвіл безкоштовно або за плату, а також може накласти певні обмеження на використання коду.

**2. Мета використання:**

* Розробник може **вільно використовувати код для особистих цілей**, наприклад, для навчання або досліджень.
* **Комерційне використання коду** для розробки інших продуктів може бути обмежене умовами ліцензії або авторським правом на інші компоненти коду.
* Розробник **не може використовувати код для створення конкуруючого продукту**, якщо він не має на це дозволу від власника авторського права.

**3. Авторське право на інші компоненти:**

* Мобільний додаток може містити **компоненти, які не підпадають під авторське право розробника**, наприклад, бібліотеки сторонніх розробників або зображення.
  + Використання цих компонентів для розробки інших продуктів може бути обмежене їх власними ліцензіями або авторським правом.

**Ліцензія на код** - це юридичний документ, який визначає умови використання, розповсюдження та модифікації програмного забезпечення. Вона встановлює права та обов'язки як розробника, так і користувачів коду.

**Де можна отримати ліцензію на код:**

* **Відкриті ліцензії:**
  + Існують численні веб-сайти, де можна знайти безліч безкоштовних відкритих ліцензій.
  + Вибираючи відкриту ліцензію, важливо чітко розуміти її умови та те, як вона впливає на використання вашого коду.
* **Платні ліцензії:**
  + Деякі розробники пропонують платні ліцензії на свій код, які надають користувачам ширші можливості використання, наприклад, комерційне використання або модифікацію коду.
  + Умови платних ліцензій зазвичай детально описуються в ліцензійній угоді, яку необхідно прочитати та погодитися перед використанням коду.
* **Власні ліцензії:**
  + Ви можете створити власну ліцензію, якщо вам потрібні специфічні умови використання вашого коду.
  + Створення власної ліцензії може бути складним завданням, тому рекомендується проконсультуватися з юристом, який спеціалізується на інтелектуальній власності.

**Навіщо потрібна ліцензія на код:**

* **Захист авторських прав:**
  + Ліцензія на код допомагає захистити авторські права розробника на його програмне забезпечення.
  + Вона чітко визначає, хто володіє кодом і як він може використовуватися.
* **Регулювання використання:**
  + Ліцензія на код дозволяє розробнику контролювати те, як його програмне забезпечення використовується іншими людьми.
  + Вона може обмежувати комерційне використання, модифікацію коду або його поширення.
* **Сприяння співпраці:**
  + Відкриті ліцензії на код сприяють співпраці та розробці програмного забезпечення спільнотою.
  + Вони дозволяють іншим розробникам використовувати, модифікувати та розповсюджувати код, що може призвести до його вдосконалення та нових інновацій.

**Авторське право на мобільний додаток: замовник vs. ІТ-компанія**

**Звичайно, це залежить від умов договору, укладеного між замовником та ІТ-компанією.**

**Загалом, авторське право на мобільний додаток зазвичай належить замовнику, якщо він чітко про це зазначив у договорі.**

Це означає, що замовник має виключне право на:

* Використовувати мобільний додаток для своїх цілей
* Розповсюджувати мобільний додаток
* Модифікувати мобільний додаток
* Створити похідні твори на основі мобільного додатку

**ІТ-компанія, як правило, володіє авторським правом на код мобільного додатку, який вона розробила.**

Це означає, що ІТ-компанія може:

* Використовувати код для розробки інших продуктів (за умови дотримання умов ліцензії)
* Надати ліцензію на код іншим сторонам (за умови згоди замовника)
* Модифікувати код
* Створити похідні твори на основі коду

**Однак, ІТ-компанія не може без згоди замовника:**

* Продавати код мобільного додатку третім сторонам
* Отримати патент на код мобільного додатку
* Використовувати код мобільного додатку для створення конкуруючого продукту

**Існує кілька способів, якими замовник може дізнатися, чи отримала ІТ-компанія ліцензію на код мобільного додатку:**

**1. Перевірити договір:**

* Найважливішим документом, який визначає права на код мобільного додатку, є **договір, укладений між замовником та ІТ-компанією.**
* У договорі повинно бути чітко прописано, чи отримала ІТ-компанія ліцензію на код, а також тип ліцензії (відкрита, закрита, ексклюзивна, неексклюзивна) та умови її використання.
* Якщо в договорі не чітко визначено, чи отримала ІТ-компанія ліцензію на код, то замовник може вимагати від ІТ-компанії надати відповідні документи.

**2. Запитати ІТ-компанію:**

* Замовник може **звернутися до ІТ-компанії з письмовим запитом**, щоб отримати інформацію про те, чи отримала вона ліцензію на код мобільного додатку.
* ІТ-компанія зобов'язана надати замовнику цю інформацію.

**3. Провести аудит:**

* Замовник може **провести аудит ІТ-компанії**, щоб перевірити, чи дотримується вона умов договору, включаючи умови використання коду мобільного додатку.

**4. Звернутися до юриста:**

* Якщо замовник має сумніви щодо того, чи отримала ІТ-компанія ліцензію на код мобільного додатку, або якщо він не задоволений відповідями ІТ-компанії, він може **звернутися до юриста, який спеціалізується на інтелектуальній власності.**
* Юрист допоможе замовнику зрозуміти його права та захистити його інтереси.

**отримання ліцензії на код не є обов'язковою умовою договору про розробку мобільного додатку.**

**Однак, це важлива умова, яку рекомендується чітко прописати в договорі.**

**Ось кілька причин, чому:**

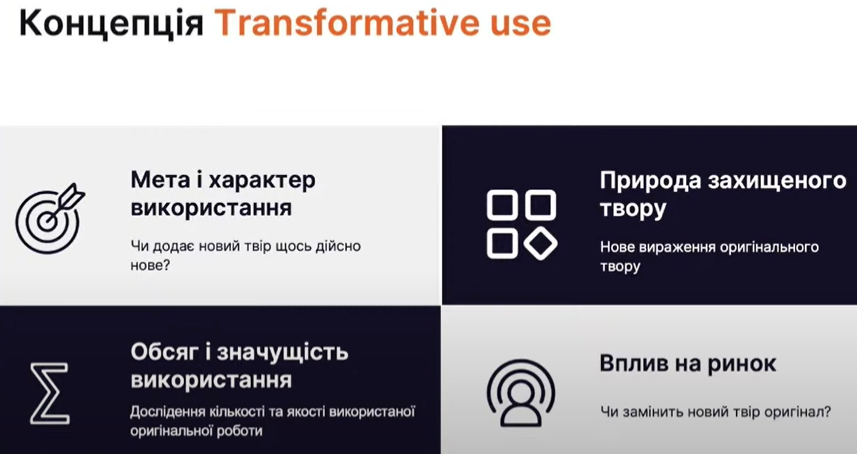
* **Захист авторських прав:**
  + Ліцензія на код допомагає захистити авторські права розробника на його програмне забезпечення.
  + Вона чітко визначає, хто володіє кодом і як він може використовуватися.
* **Регулювання використання:**
  + Ліцензія на код дозволяє розробнику контролювати те, як його програмне забезпечення використовується іншими людьми.
  + Вона може обмежувати комерційне використання, модифікацію коду або його поширення.
* **Сприяння співпраці:**
  + Відкриті ліцензії на код сприяють співпраці та розробці програмного забезпечення спільнотою.
  + Вони дозволяють іншим розробникам використовувати, модифікувати та розповсюджувати код, що може призвести до його вдосконалення та нових інновацій.

**Якщо в договорі не чітко визначено, чи отримала ІТ-компанія ліцензію на код, то це може призвести до суперечок і непорозумінь.**

**Тому рекомендується замовнику та ІТ-компанії чітко обговорити це питання та прописати в договорі:**

* **Чи отримує ІТ-компанія ліцензію на код?**
* **Який тип ліцензії (відкрита, закрита, ексклюзивна, неексклюзивна)?**
* **Умови використання коду.**

**В авторському праві існує відповідна концепція, на основі якої варто будувати власну політику створення контенту:**

****

## Концепція трансформаційного використання

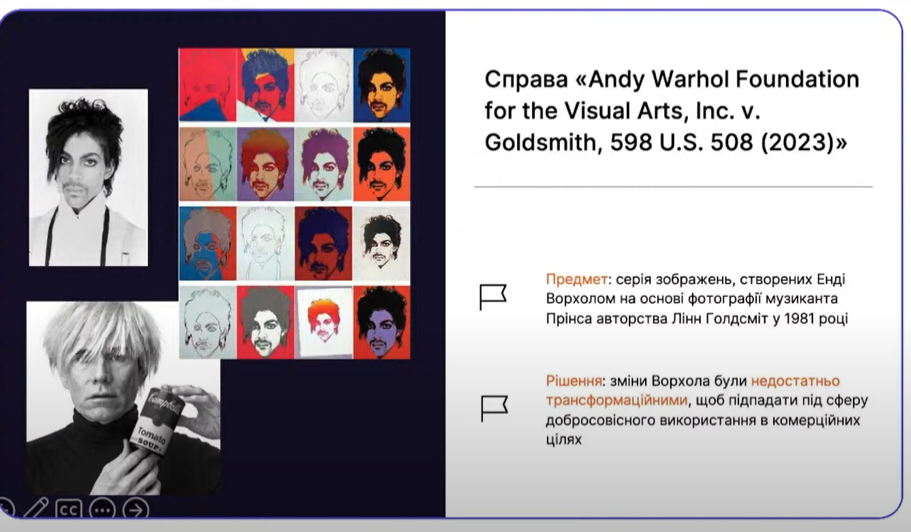
**Трансформаційне використання** - це концепція в авторському праві, яка дозволяє використовувати матеріал, захищений авторським правом, новим і відмінним способом, не порушуючи при цьому прав власника авторського права.

**Трансформаційне використання** зазвичай вважається **добросовісним використанням** захищеного авторським правом матеріалу, навіть якщо таке використання не авторизовано власником авторського права. Це пояснюється тим, що трансформаційне використання додає щось нове до оригінального твору, наприклад, нове значення, вираз або мету.

**Приклади трансформаційного використання:**

* Створення пародії на твір, захищений авторським правом
* Використання твору, захищеного авторським правом, для створення нового твору мистецтва, наприклад, картини чи скульптури
* Використання захищеної авторським правом пісні для створення нового реміксу чи mashup
* Використання уривка з захищеного авторським правом фільму для створення нового відеокоментаря
* **Трансформаційне використання є важливою концепцією в авторському праві,** адже воно дозволяє створювати нові та інноваційні твори мистецтва, літератури та музики. Воно також дозволяє використовувати матеріал, захищений авторським правом, для критики, коментарів або пародіювання оригінального твору.
* **У контексті розробки програмного забезпечення** трансформаційне використання може бути використано для виправдання використання захищеного авторським правом коду новим і відмінним способом. Наприклад, розробник може використовувати трансформаційне використання, щоб виправдати використання захищеного авторським правом коду для створення нового проекту відкритого програмного забезпечення.

**Приклад невдалого використання концепції**

****

* **Замовник справи:** Andy Warhol Foundation for the Visual Arts, Inc. (Фонд візуальних мистецтв Енді Ворхола)
* **Відповідач:** Lynn Goldsmith (Лінн Голдсміт)

**Справа:** Andy Warhol Foundation for the Visual Arts, Inc. v. Goldsmith, 598 U.S. 508 (2023)

**Суть справи:**

Фонд візуальних мистецтв Енді Ворхола оскаржував твердження Лінн Голдсміт про те, що використання нею зображення Принца, створеного Енді Ворхолом, є порушенням авторського права.

**Верховний суд США постановив, що використання Фондом Ворхола зображення Принца не було трансформаційним і тому порушує авторське право Голдсміт.**

**Деталі:**

* У 1981 році Голдсміт сфотографувала музиканта Принца для журналу Newsweek.
* У 1984 році Фонд Ворхола створив серію портретів Принца на основі фотографії Голдсміт.
* Голдсміт не дала згоди на використання її фотографії.
* У 2016 році Фонд Ворхола продав один із портретів за 10,5 мільйона доларів.
* Голдсміт подала до суду на Фонд Ворхола, стверджуючи, що використання нею її фотографії було порушенням авторського права.
* Окружний суд постановив на користь Фонду Ворхола, заявивши, що використання було трансформаційним.
* Апеляційний суд скасував це рішення, заявивши, що використання не було трансформаційним.
* Верховний суд підтвердив рішення Апеляційного суду.

**Виняткове право Закон України про авторське право (ЗУ Про авторське право і суміжні права) надає неоригінальним об’єктам, створеним комп’ютерними програмами Sui generis авторських прав.**

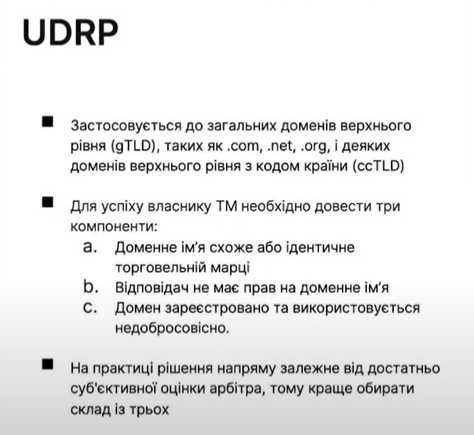
Прикладом **sui generis права** може бути захист баз даних. В Україні, як і в багатьох інших країнах, бази даних, які не відповідають критеріям оригінальності, все ж можуть отримати захист завдяки sui generis праву. [Це право надає власнику бази даних виключне право на використання, копіювання та розповсюдження бази даних1](https://cedem.org.ua/analytics/shtuchnyi-intelekt-pravo/).

Інший приклад — це захист неоригінальних об’єктів, створених за допомогою штучного інтелекту. [Наприклад, якщо комп’ютерна програма генерує зображення або текст, який не є оригінальним твором, власник програми може отримати sui generis право на цей об’єкт2](https://ukrpatent.org/uk/news/main/pravo-sui-generis-dyskusiia-02052024)

**Перші кроки (досудове вирішення) по шляху вирішення порушень прав інтелектуальної власності:**

****

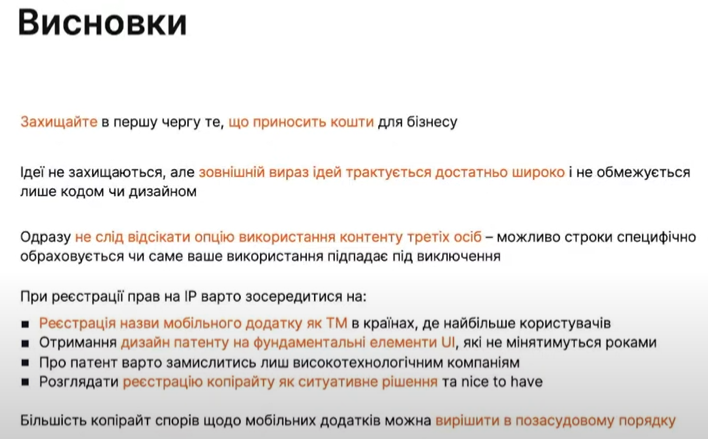
**А також скарги на порушників використання доменів:**

****

****

****

**Обмеження щодо збору даних в США**

****

**Додаток А**

**Договір**

Цей Договір ("Угода") укладається між:

Джон Клайд CreatorTY

803 Eastern Pkwy, Бруклін, 136 Grand St, Нью-Йорк, NY 10013, США 11213, США

ПРАВИЛА ТА УМОВИ

**Ця Угода набирає чинності з понеділка, 20 квітня 2024 року, а проект планується завершити в середу, 1 вересня 2024 року.** Підрядник не зобов’язується вписатися в ці терміни, хоча має розглядати це як керівництво.

CreatorTY має розробити, створити, розробити, протестувати та запустити програму для редагування відео для iOS, яка має такі функції:

Додаапти відео з галереї

Створення нового проекту

Увійти за допомогою облікових записів Facebook, Google, Apple

Обрізка відео

Додавати кольорові фільтри до відео

Додавати музику до відео (ITunes API)

Додавати текст до відео

Часові шкали: можливість переглядати та змінювати всі треки ефектів на одному екрані

Збереження відео в хмарному сервісі програми

Експортувати відео з якістю 360p, 480p і 720p

Вибір різних відеокадрів (4:3, 16:9)

Публікація безпосередньо в соціальних мережах

Картинка в картинці

**Орієнтовна оцінка цього набору функцій становить 110 000 доларів США**. Підрядник не зобов’язується вкластися в цей бюджет, хоча повідомить про те, як він використовується.

ВІДПОВІДАЛЬНІСТЬ ПІДРЯДНИКА

CreatorTY несе відповідальність за забезпечення необхідного стандарту якості своєї продукції. У разі недоліків у результатах роботи CreatorTY виправляє та/або виправляє помилки.

**Виправлення помилок забезпечується CreatorTY за рахунок додаткового бюджету.**

Будь-які передоплати, здійснені CreatorTY, не повинні тлумачитися як відмова Джона Клайда. У разі пошкодження або травм CreatorTY несе відповідальність за збитки.

ПІДПРИЄДНАННЯ

CreatorTY має отримати згоду з Джоном Клайдом на будь-яку роботу, яка буде виконана протягом життя проекту в частині субпідряду, заміни або спільного контракту з іншою фірмою.

ПРОЕКТНА КОМАНДА

Команда проекту означає Джона Клайда та будь-яких співробітників CreatorTY, найманих Джоном Клайдом з метою проектування, виробництва, розробки, тестування та запуску Проекту. Члени команди проекту будуть призначені в письмовій формі Джоном Клайдом і можуть час від часу змінюватися в письмовій формі Джоном Клайдом.

CreatorTY не видаляє або замінює жодних членів команди проекту, за винятком письмового дозволу Джона Клайда. Крім того, якщо будь-який член команди проекту припиняє роботу в Проекті з будь-якої причини, CreatorTY негайно замінить такого члена команди кваліфікованою особою, затвердженою Джоном Клайдом, у письмовій формі.

РОЗРАХУНОК ТА ОПЛАТА

**Рахунки Джону Клайду виставлятимуться щомісяця** **до надання результатів однієї ітерації**, а сплата таких зборів — протягом тридцяти (30) днів з дати кожного рахунку CreatorTY. Усі платежі будуть здійснюватися в доларах США. Прострочені платежі за цим договором нараховуватимуться у розмірі півтора відсотка (1 1/2%) на місяць або найвищої ставки, дозволеної чинним законодавством, залежно від того, яка з них нижча. Якщо за своїм рішенням CreatorTY визначить, що Джон Клайд не є кредитоспроможним або не є фінансово захищеним, CreatorTY може, після письмового повідомлення Джона Клайда, змінити умови оплати, щоб вимагати повної оплати до надання результатів однієї ітерації або інших гарантій для забезпечити виконання платіжних зобов'язань Клієнта за цим Договором.

**Розробники:**

Senior Старший розробник програмного забезпечення, призначений для роботи над Проектом, отримує оплату в розмірі 45 $ на годину.

Middle Середній розробник програмного забезпечення, призначений для роботи над Проектом, отримує оплату в розмірі 40 $ на годину.

Junior Молодший розробник програмного забезпечення, призначений для роботи над Проектом, отримує оплату в розмірі 36 $ на годину.

QA-інженери:

Senior Старший QA-інженер, призначений для роботи над Проектом, отримує оплату в розмірі 40 $ на годину.

Middle Середній QA-інженер, призначений для роботи над Проектом, отримує оплату в розмірі 35 $ на годину.

Junior Молодший QA-інженер, призначений для роботи над Проектом, отримує оплату в розмірі 31 $ на годину.

UI-дизайнер, призначений для роботи над Проектом, отримує оплату 30$ за годину.

Керівник проекту, призначений для роботи над Проектом, отримує оплату в розмірі 45 $ на годину.

ПОНАДУРОЧНИЙ ЧАС

Надурочна робота (понад 8 годин за робочий день) оплачується у подвійному місячному розмірі. Усі понаднормові роботи повинні бути схвалені Джоном Клайдом або представником першої сторони до їх фактичного настання. CreatorTY має право відмовитися від запитів на понаднормові години.

ВІДПУСКА, ЛІКУВАННЯ ТА ВІДПУСКА

Співробітники CreatorTY мають право на звичайну відпустку, згідно з урядовими законами США. Якщо співробітники CreatorTY, які на даний момент працюють у проекті Джона Клайда, беруть відпустку більше двох (2) тижнів, CreatorTY повинен повідомити Джона Клайда принаймні за один (1) місяць до цього.

Дні відпустки та лікарняні виключаються з рахунку, запланованого на наступний місяць.

Розрахунок для виключення має бути таким:

Сума для виключення з Місячної ставки = 1 місячна ставка / кількість робочих днів у місяці, коли береться відпустка та/або лікарняні \* кількість робочих днів у місяці, коли береться відпустка та/або лікарняні.

Усі державні свята вважаються вихідними. Усі місцеві національні свята CreatorTY вважаються вихідними, тому їм має передувати письмове сповіщення від CreatorTY.

ІНТЕЛЕКТУАЛЬНА ВЛАСНІСТЬ

Усі зареєстровані або незареєстровані, (i) патенти, патентоване винаходи та інші патентні права (включаючи будь-які розділи, продовження, часткове продовження, перевидання, повторну експертизу та їх втручання); (ii) товарні знаки, знаки обслуговування, торговельні назви, торгові назви, слогани, фірмові назви, логотипи та корпоративні назви та весь гудвіл, пов'язаний з ними; (ii) авторські права, роботи та дизайни масок; (iv) комерційні таємниці, ноу-хау, винаходи, процеси, процедури, бази даних, конфіденційну ділову інформацію та іншу інформацію та права власності; (v) комп'ютерні програми, включаючи весь вихідний код, об'єктний код, специфікації, конструкції та пов'язану з ними документацію; і (vi) доменні імена, Інтернет-адреси та інші ідентифікатори комп’ютерів, у кожному разі, які є суттєвими та використовуються в роботі Проекту CreatorTY, а Джон Клайд, який в даний час ведеться (“Інтелектуальна власність”) є (a) власністю John Clyde або (b) використовується Джоном Клайдом для дійсної ліцензії.

Наскільки CreatorTY та Джон Клайд знають, жодних порушень чи інших порушень будь-якої інтелектуальної власності, що належить Джону Клайду, будь-яка третя сторона, за винятком випадків, коли таке порушення чи порушення не мали б суттєвого негативного впливу.

CreatorTY та Джон Клайд вжили всіх необхідних заходів для підтримки та захисту кожного об’єкта інтелектуальної власності. Проведення проекту CreatorTY і Джона Клайда не порушує або іншим чином не порушує будь-яку інтелектуальну власність або інші права власності будь-якої іншої особи в суттєвому відношенні, і немає жодних дій, які очікують на розгляд, або, наскільки відомо CreatorTY і Джон Клайд погрожував, стверджуючи про будь-яке таке порушення чи порушення чи оскарження прав Джона Клайда на або на будь-яку інтелектуальну власність, за винятком будь-яких Позовів, які, у разі негативного для Джона Клайда вирішення, не мали б суттєвих негативних наслідків.

ПРИЙНЯТТЯ

Прийняття поставки. Вважається, що CreatorTY виконав свої зобов’язання щодо доставки, якщо, на думку Джона Клайда, результат відповідає критеріям прийнятності та

Джон Клайд письмово повідомляє CreatorTY про те, що він приймає доставку.

Перевірка та повідомлення. Джон Клайд матиме 7 робочих днів, щоб перевірити та оцінити продукт на дату доставки, перш ніж сповістити CreatorTY про те, що він приймає або відхиляє результат.

ПОСТАВКИ

CreatorTY погоджується розробляти результати, які відповідають Специфікаціям. «Технічні характеристики» — це функції, сумісність, функціональність, продуктивність, описи, вимоги та інші специфікації, встановлені CreatorTY та Джоном Клайдом. Специфікації можуть бути змінені Джоном Клайдом та CreatorTY.

Розробка та графік постачання. Розробка та надання Результатів та надання Послуг відбуватиметься відповідно до Графіку часу, який обговорюється та встановлюється двома сторонами у письмовій формі.

Звіти про хід виконання. За запитом Джона Клайда CreatorTY надає Джону Клайду письмові звіти про прогрес із описом статусу та прогресу Послуг і Результатів.

Доставка, тестування та виправлення. Коли CreatorTY завершить робочу версію Результатів, результати мають бути протестовані та впроваджені. Якщо виявлено проблеми, їх оцінюють. Якщо вони варті виправлення та схвалені для виправлення Джоном Клайдом, помилки виправляються. Після цього починається новий цикл роботи.

Джон Клайд не повинен безпідставно відмовлятися від прийняття.

Вихідний код і середовище розробки. CreatorTY передає Джону Клайду будь-який вихідний код, об’єктний код, виконуваний код, псевдокод, проекти, програмну документацію, блок-схеми, логічні схеми, специфікації та інші авторські роботи, які можуть бути написані або створені як частина або в зв’язок із Послугами чи Результатами або їх розробка, тестування чи виправлення та все вищезазначене вважається частиною «Результатів» для цілей цієї Угоди. CreatorTY має включати коментарі до вихідного коду. Коментарі та організація вихідного коду, а також документація з програмування, надана Джону Клайду, повинні відповідати професійним стандартам комп’ютерного програмування та бути достатніми, щоб дозволити програмістам, найманим Джоном Клайдом, підтримувати та покращувати результати.

CreatorTY також надасть Джону Клайду «Середовище розвитку» для Результатів. «Середовище розробки» означає програмні інструменти, утиліти, програмне забезпечення для автоматизації розробки та інші матеріали та елементи коду, які використовуються програмістами CreatorTY для проектування, розробки, компіляції, створення, тестування, підтримки та покращення Результатів. Все, що необхідно для компіляції або створення Результатів (крім комерційно доступних операційних систем, компіляторів, наборів інструментів і продуктів), повинно бути включено в Середовище розробки та разом із ним. Якщо компонент Середовища розробки комерційно доступний для громадськості, CreatorTY потрібно лише ідентифікувати компонент у письмовому документі, включеному в Середовище розробки. Середовище розробки не містить жодних результатів (тобто воно є на додаток до результатів).

ЗМІНИ АБО МОДИФІКАЦІЇ

Сторони погоджуються, що будь-які зміни або модифікації в домовленості, які можуть бути погоджені ними, будуть внесені в письмовій формі до початку робіт, які мають бути виконані. Будь-які дії, вчинені CreatorTY, які не зазначені в плані, будуть видалені CreatorTY за власний рахунок.

ВІДМОВИ

Жодна відмова будь-якої сторони не буде вважатися зробленою без будь-якого письмового повідомлення та підписаної стороною, яка виконує відмову. Відмова від будь-яких майбутніх дій є недійсною.

ПРАВОМ, ЩО РЕГУЛЮЄ

Ця Угода підпорядковується та тлумачиться відповідно до законодавства Нью-Йорка.

ЧАС – СУТНІСТЬ

Дія цієї Угоди починається з зазначеного дня і діє до настання або раніше встановленого терміну. Будь-яка затримка, яку зазнає боржник і без якої буде визнано необґрунтованою, є порушенням, і потерпіла сторона може вимагати відшкодування збитків.

РОЗДІЛЬНІСТЬ

Якщо будь-який з пактів, положень, положень, зазначених у цьому документі, буде визнано будь-яким компетентним судом незаконним або недійсним, така незаконність вплине лише на таке положення, а інші положення залишаються дійсними та підлягають виконанню.

НА ПІДТВЕРДЖЕННЯ ЧОГО сторони підтвердили, що ці подарунки мають бути виконані в день і рік, уперше зазначені вище.

ГАРАНТІЯ ПРОЕКТУ

Грантоотримувач заявляє, гарантує та зобов’язується, що він повністю обізнаний з положеннями, умовами та вимогами цієї Угоди та будь-яких застосовних законів, що регулюють цю Угоду або Проект. Проект повинен відповідати таким умовам або вимогам у всіх суттєвих аспектах. Грантоотримувач також представляє, гарантує та співпрацює