МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет біоресурсів і природокористування України

Методичні вказівки до виконання лабораторних робіт з дисципліни

«Бізнес-аналіз в розробці програмного забезпечення (частина 1)»

для студентів ОС «Магістр»,
що навчаються за спеціальністю
«Інженерія програмного забезпечення»

Київ 2026

УДК 004.94
ББК 32.973-018.2

Укладач: к.е.н., Ніколаєнко Д.В.

Рецензенти: к.т.н., доцент Голуб Б.Л., д.е.н., доцент Кравченко В.М.

Затверджено навчально-методичною комісією факультету інформаційних технологій Національним університетом біоресурсів і природокористування України (протокол № від р.)

Методичні вказівки до виконання лабораторних робіт з дисципліни «Бізнес-аналіз в розробці програмного забезпечення (частина 1)»/ уклад.: к.е.н. Ніколаєнко Д.В. – К.:НУБіП, 2025. – 41 с.

Містить загальні рекомендації до виконання лабораторних робіт. Для студентів, що навчаються за ОПП «Програмне забезпечення інформаційних систем» ОС «Магістр»

© Ніколаєнко Д.В.

ЗМІСТ

ЗМІСТ 3
ЗАГАЛЬНІ МЕТОДИЧНІ ВКАЗІВКИ 4
ВСТУП 5
ЛАБОРАТОРНА РОБОТА №1 6
Теоретичні вказівки 6
Завдання 14
ЛАБОРАТОРНА РОБОТА №2 15
Теоретичні вказівки 15
Завдання 17
ЛАБОРАТОРНА РОБОТА №3 19
Теоретичні вказівки 19
Завдання 23
ЛАБОРАТОРНА РОБОТА №4 25
Завдання 25
ЛАБОРАТОРНА РОБОТА №5 26
Теоретичні вказівки 26
Завдання 31
ЛАБОРАТОРНА РОБОТА №6 33
Теоретичні вказівки 33
Завдання 35
ЛАБОРАТОРНА РОБОТА №7 36
Теоретичні вказівки 36
Завдання 40
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 41

ЗАГАЛЬНІ МЕТОДИЧНІ ВКАЗІВКИ

Для здачі лабораторної роботи студенту потрібно ознайомитися з теоретичним матеріалом, що представлений в лекції, методичними вказівками до відповідної роботи, підготувати звіт у форматі ".doc" або ".pdf" та завантажити його в ЕНК, захистити роботу під час проведення заняття за розкладом. Лабораторна робота вважається зданою , якщо за нею студент отримав не менше, ніж 60% зазначеної кількості балів у лабораторній роботі.

Звіт повинен відповідати такій структурі:

- титульна сторінка, де зазначена тема лабораторної роботи та хто її виконав;
- формулювання завдання;
- хід виконання роботи.

ВСТУП

Метою викладання дисципліни “Управління програмними проектами” є набуття студентами знань з підходів щодо управління життєвим циклом проектів в галузі інформаційних систем.

У результаті вивчення дисципліни “ Управління програмними проектами” студенти повинні мати знання з питань:

- основні поняття та термінологію курсу,
- визначення основних параметрів та характеристик проєкту,
- технології управління проєктами,
- основи мережевого планування та управління якістю, часом та ризиками проєкту

оволодіти практичними навичками:

- визначати основні характеристики проєкту,
- складати календарний план та бізнес-план проєкту,
- визначати параметри часу, якості та витрат проєкту,
- оптимізувати проєкт за різними параметрами

Матеріал, який викладається у цій дисципліні, використовується студентами для написання дипломної роботи магістра.

ЛАБОРАТОРНА РОБОТА №1

Тема

Мережева модель проекту

Мета роботи Придбання навичок побудови і розрахунку часових параметрів моделей мережевого планування і управління.

Теоретичні вказівки

Основними поняттями мережевих моделей є поняття події і роботи.

Робота - це певний процес, що приводить до досягнення певного

результату, що вимагає витрат яких-небудь ресурсів і має протяжність в часі.

По своїй фізичній природі роботи можна розглядати як:

• дія: розробка специфікації, розробка CRUD команд, розгортка енвайременту;

• процес: копіювання даних, травлення плат;

• очікування: очікування поставки комплектуючих, очикування тестування задачі .

За кількістю затраченого часу робота може бути:

• дійсної, тобто що вимагає витрат часу;

• фіктивної, тобто формально не вимагає витрат часу і представляє зв'язок між будь-якими роботами,

наприклад: передача змінених креслень від конструкторів до технологів; здача звіту.

Подія - це момент часу, коли завершуються одні роботи і починаються інші. Наприклад, фундамент залитий бетоном, старіння виливків завершено, комплектуючі поставлені, звіти здані і т.д. Подія являє собою результат проведених робіт і, на відміну від робіт, не має протяжності в часі.

На етапі структурного планування взаємозв'язок робіт і подій, необхідних для досягнення кінцевої мети проекту, зображується за допомогою мережного графіка (мережевої моделі). На мережевому графіку роботи зображуються стрілками, які з'єднують вершини, що зображують події. Початок і закінчення будь-якої роботи описуються парою подій, які називаються початковим і кінцевим подіями. Тому для ідентифікації конкретної роботи використовують код роботи (i, j), що складається з номерів початкової (i-ої) і кінцевої (j-ої) подій (див. рис.1.1).

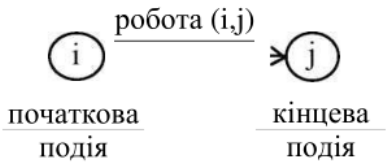


Рис.1. Кодування робіт

Будь-яка подія може вважатися такою що настала тільки тоді, коли закінчаться всі вхідні в неї роботи.

Тому, роботи, що виходять з деякої події не можуть розпочатися, поки не будуть завершені всі роботи, що входять в цю подію.

Подія, що не має попередніх до неї подій, тобто з якої починається проект, називають початковою. Подія, яка не має наступних подій і відображає кінцеву мету проекту, називається завершальною.

При побудові мережевого графіка необхідно дотримуватися наступних правил:

• довжина стрілки не залежить від часу виконання роботи;

• стрілка може не бути прямолінійним відрізком;

• для дійсних робіт використовуються суцільні, а для фіктивних - пунктирні стрілки;

• кожна операція повинна бути представлена тільки однією стрілкою;

• між одними і тими ж подіями не повинно бути паралельних робіт, тобто робіт з однаковими кодами;

• слід уникати перетину стрілок;

• не повинно бути стрілок, спрямованих справа наліво;

• номер початкового події повинен бути менше номера кінцевого події;

• не повинно бути висячих подій (тобто, які не мають попередніх подій), крім початкової;

• не повинно бути тупикових подій (тобто, які не мають наступних подій), крім завершальної;

• не повинно бути циклів (див. Рис.1.2).

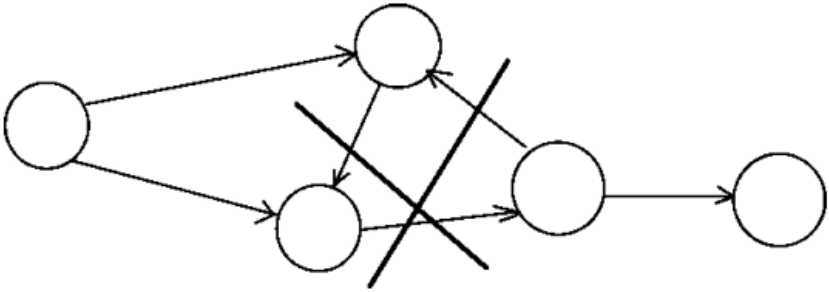


Рис.1.2. Неприпустимість циклів

Важливе значення для аналізу мережевих моделей має поняття шляху.

Шлях - це будь-яка послідовність робіт в мережевому графіку (в окремому випадку це одна робота), в якій кінцева подія однієї роботи збігається з початковою подією наступної за нею роботи. Розрізняють такі види шляхів.

Повний шлях - це шлях від початкової до завершальної події. Критичний шлях - максимальний за тривалістю повний шлях. Роботи, що лежать на критичному шляху, називають критичними. Підкритичній шлях – повний шлях, найближчий за тривалістю до критичного шляху.

Побудова мережі є лише першим кроком на шляху до побудови календарного плану. Другим кроком є

розрахунок мережевої моделі, який виконують прямо на мережевому графіку, користуючись простими правилами.

Часові параметри подій

До часових параметрів подій відносяться:

• Тр (і) - ранній термін настання події і. Це час, який необхідно для виконання всіх робіт, що передують даній події і. Воно дорівнює найбільшою з тривалості шляхів, що передують даній події.

• Тп (і) - пізній термін настання події і. Це такий час настання події і, перевищення якого викличе аналогічну затримку настання завершального події мережі. Пізній термін настання будь-якої події «і» дорівнює різниці між тривалістю критичного шляху і найбільшою з тривалістю шляхів, наступних за подією і.

• R (і) - резерв часу настання події і. Це такий проміжок часу, на який може бути відстрочено настання події «і» без порушення термінів завершення проекту в цілому. Початкові і кінцеві події критичних робіт мають нульові резерви подій.

Розраховані чисельні значення часових параметрів записуються прямо в вершини мережевого графіка (див. рис.1.3).

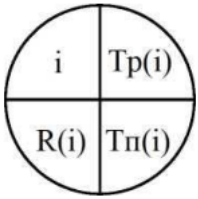


Рис. 3. Відображення часових параметрів подій в вершинах мережевого графіка

Розрахунок ранніх термінів звершення подій Тр (і) ведеться від початкової (П) до завершальної (З) події.

Примітка. Оскільки тривалість роботи може бути як нормальної Тн, так і прискореної Ту(п) (див. п. 3), то для спільності викладу будемо надалі позначати поточну тривалість роботи буквою t з відповідним кодом роботи, наприклад, t (i, j), t (k, j) і т.п.

1. Для початкової події (П) Тр(П)=0.

2. Для всіх інших подій «і» Тр (і) = max[Тр (k) + t(k,i)], де максимум береться по всіх роботах (k, i), що входять в подію і.

Пізні терміни звершення подій Тп (і) розраховуються від завершальної до початкової подій.

3. Для завершальної події (З) Тп(З)= Тр(З).

4. Для всіх інших подій Тп(і) = min[Тп(j) − t(i, j)], де мінімум береться по всіх роботах (i, j), які виходять з події і.

5. R(i) = Тп (і) − Тр (і).

Часові параметри робіт і шляхів

До найбільш важливим часовим параметрам робіт відносяться:

• Трп (i, j) - ранній термін початку роботи;

• Тпп(i,j) - пізній термін початку роботи;

• Трз(i,j) - ранній термін закінчення роботи;

• Тпз(i, j) - пізній термін закінчення роботи;

Для критичних робіт Трп(i, j) = Тпп(i, j) и Трз (i, j) = Тпз (i, j).

• Rп(i,j) - повний резерв роботи показує максимальний час, на який може бути збільшена тривалість роботи (i, j) або відстрочено її початок, щоб тривалість максимального шляху, що проходить через неї, не перевищила тривалості критичного шляху. Найважливіша властивість повного резерву роботи (i, j) полягає в тому, що його часткове або повне використання зменшує повний резерв у робіт, що лежать з роботою (i, j) на одному шляху.

Таким чином, повний резерв належить не одній даній роботі (i, j), а всім роботам, які лежать на шляхах, що проходять через цю роботу.

• Rc(i, j) - вільний резерв роботи показує максимальний час, на який можна збільшити тривалість роботи (i, j) або відстрочити її початок, не змінюючи ранніх термінів початку наступних робіт. Використання вільного резерву однієї з робіт не змінює величини вільних резервів решти робіт мережі.

Часові параметри робіт мережі визначаються на основі ранніх і пізніх термінів подій.

• 1) $Трп\ (i, j) = Тр\ (i)$;

• 2) $Трз(i, j) = Тр(i) + t(i, j)$ или $Трз(i, j) = Трп(i, j) + t(i, j)$;

• 3) $Тпз(i, j) = Тп(\ j)$;

• 4) $Тпп\ (i, j) = Тп\ (\ j) - t(i, j)$ или $Тпп(i, j) = Тпз(i, j) - t(i, j)$;

• 5) $Rп(i, j) = Тп(\ j) - Тр(i) - t(i, j)$;

• 6) $Rс(i, j) = Тр(\ j) - Тр(i) - t(i, j)$.

Часові параметри робіт вносяться в таблицю. При цьому коди робіт записують в певному порядку: спочатку записуються всі роботи, що виходять з вихідної, тобто першої, події, потім - що виходять з другого події, потім - з третьої і т.п.

Резерви часу, крім робіт і подій, мають повні шляхи мережевий моделі.

Різниця між тривалість критичного шляху $Т(Lкр)$ і тривалістю будь-якого іншого повного шляху $Т(Lп)$ називається повним резервом часу шляху Lп, тобто $R(Lп) = Т(Lкр) - Т(Lп)$. Цей резерв показує, на скільки в сумі може бути збільшена тривалість всіх робіт даного шляху L, щоб при цьому не змінився загальний термін закінчення всіх робіт.

Завдання

1. Відповідно до номеру свого варіанту отримайте наступні вихідні дані: Тн - час нормальної тривалості кожної роботи мережевої моделі і опис упорядкування цих робіт.
2. Відповідно до правил побудови мережевих графіків і на основі вихідних даних Вашого варіанту побудуйте мережеву модель, потім необхідно пронумерувати події отриманої мережі.
3. У відповідності з методиками, описаними в вище
   - розрахуйте і відобразіть на мережевому графіку часові параметри подій: ранній і пізній термін звершення події, резерв події;
   - розрахуйте і відобразіть в таблиці часові параметри робіт: час раннього і пізнього початку робіт; час раннього і пізнього закінчення робіт; повний і вільний резерви робіт.
4. Звіт з лабораторної роботи повинен містити:
   - номер варіанту;
   - вихідні дані варіанту;
   - мережевий графік з відображеними на ньому часових параметрів подій;
   - таблицю з кодами і часовими параметрами робіт.

ЛАБОРАТОРНА РОБОТА №2

Тема

Оптимізація виконаців

Мета роботи знайомство з методикою і придбання навичок проведення оптимізації мережевих моделей за критерієм "Мінімум виконавців".

Теоретичні вказівки

При оптимізації використання ресурсу робочої сили найчастіше мережеві

- роботи прагнуть організувати таким чином, щоб:
- кількість одночасно зайнятих виконавців було мінімальним;

вирівняти потребу в людських ресурсах протягом терміну виконання проекту.

Суть оптимізації завантаження мережевих моделей за критерієм "мінімум виконавців" полягає в наступному: необхідно таким чином організувати виконання мережевих робіт, щоб кількість одночасно працюючих виконавців було мінімальним. Для проведення подібних видів оптимізації необхідно побудувати і проаналізувати графік прив'язки і графік завантаження.

Графік прив'язки відображає взаємозв'язок виконуваних робіт в часі і будується на основі даних

або про тривалості робіт (в даній лабораторній це Тн), або про ранні терміни початку і закінчення робіт. При першому способі побудови необхідно пам'ятати, що робота (i, j) може почати виконуватися тільки після того як будуть виконані всі попередні їй роботи (k, j). За вертикальної осі графіка прив'язки відкладаються коди робіт, по горизонтальній осі – тривалість робіт (ранній початок і раннє закінчення робіт).

На графіку завантаження по горизонтальній осі відкладається час, наприклад в днях, по вертикальної - кількість осіб, зайнятих роботою в кожен конкретний день. Для побудови графіка завантаження необхідно:

• на графіку прив'язки над кожною роботою написати кількість її виконавців;

• підрахувати кількість працюючих в кожен день виконавців і відкласти на графіку завантаження.

Для зручності побудови і аналізу графіки завантаження і прив'язки слід розташовувати один над іншим.

Описані види оптимізації завантаження виконуються за рахунок зсуву в часі некритичних робіт, тобто робіт, що мають повний і / або вільний резерви часу.

Повний і вільний резерви будь-якої роботи можна визначити без спеціальних розрахунків, аналізуючи тільки графік прив'язки. Зрушення роботи означає, що

вона буде виконуватися вже в інші дні (тобто зміниться час її початку і закінчення), що в свою чергу призведе до зміни кількості виконавців, які працюють одночасно (тобто рівня щоденного завантаження мережі).

Завдання

1. Відповідно до номеру свого варіанту отримуйте дані про кількість виконавців, зайнятих на кожній роботі в мережевій моделі, і обмеження за чисельністю N одночасно зайнятих в роботі виконавців.
2. Побудуйте в звіті графіки прив'язки і завантаження, використовуючи нормальні тривалості робіт мережі - Тн (див. П.2.3.1), і покажіть їх викладачеві.
3. Перевірте правильність побудови графіків прив'язки і завантаження за допомогою комп'ютера, в разі необхідності виявите і усуньте помилки.
4. Проведіть зменшення чисельності виконавців, одночасно зайнятих на роботах мережі, до необхідного рівня N.
5. Звіт з лабораторної роботи повинен містити:
   - номер варіанта;

- вихідні дані варіанту;
- графіки прив'язки і завантаження до проведення оптимізації завантаження;
- графіки прив'язки і завантаження після проведення оптимізації завантаження (можливе використання пунктирних ліній на спочатку побудованих графіках для відображення змін в прив'язці робіт і завантаженні мережі, викликаних зсувами робіт);
- коди робіт, зсунутих в процесі оптимізації, і час їх зсуву.

ЛАБОРАТОРНА РОБОТА №3

Тема

Оптимізація витрат

Мета роботи знайомство з методикою і придбання навичок проведення оптимізації мережевих моделей за критерієм "Час-витрати".

Теоретичні вказівки

Метою оптимізації за критерієм "Час - витрати" є скорочення часу виконання проекту в цілому. Ця оптимізація має сенс тільки в тому випадку, коли час виконання робіт може бути зменшено за рахунок задіяння додаткових ресурсів, що призводить до підвищення витрат на виконання робіт (див. рис.3.1).

Для оцінки величини додаткових витрат, пов'язаних з прискоренням виконання тієї чи іншої роботи, використовуються або нормативи, або дані про виконання аналогічних робіт в минулому. Під параметрами робіт Сн (i, j) і Сп (i, j) розуміються так звані прямі витрати, безпосередньо пов'язані з виконанням конкретної роботи. Таким чином, непрямі витрати типу адміністративно-управлінських в процесі скорочення тривалості проекту до уваги не приймаються, однак їх вплив враховується при виборі остаточного календарного плану проекту.

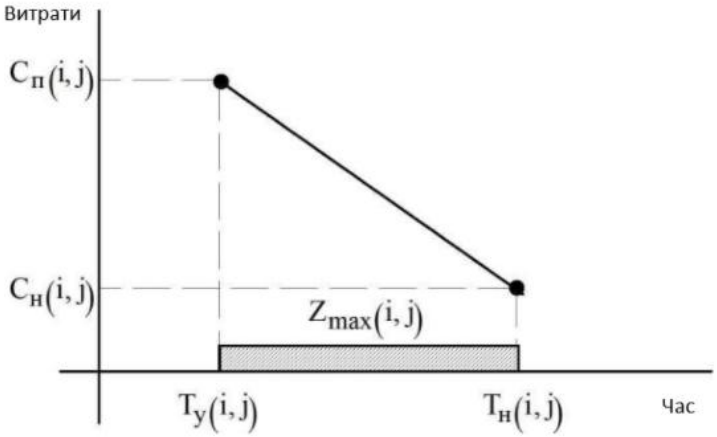


Рис.3.1. Залежність прямих витрат на роботу від часу її виконання

Важливими параметрами роботи (i, j) при проведенні даного виду оптимізації є:

• коефіцієнт наростання витрат

$$k(i,j)=\frac{C_П(i,j)-C_Н(i,j)}{T_Н(i,j)-T_У(i,j)},$$

який показує витрати коштів, необхідні для скорочення тривалості роботи (i, j) на один день;

• запас часу для скорочення тривалості роботи в поточний момент часу

$$Z_Т(i,j)=t_Т(i,j)-T_У(i,j),$$

де $t_т$ (i, j) - тривалість роботи (i, j) на поточний момент часу, максимально можливе значення запасу часу роботи дорівнює

$$Z_{max}(i,j)=T_Н(i,j)-T_У(i,j).$$

Ця ситуація має місце, коли тривалість роботи (i, j) ще жодного разу не скорочували, тобто $t_т(i,j) = T_н(i,j)$.

Загальна схема проведення оптимізації "час-витрати"

1. Виходячи з нормальних тривалостей робіт $T_н$ (i,j), визначаються критичні $L_{кр}$ і підкритичні $L_п$ шляхи мережевої моделі і їх тривалості $T_{кр}$ і $T_п$.

2. Визначається сума прямих витрат на виконання всього проекту при нормальній тривалості робіт.

3. Розглядається можливість скорочення тривалості проекту, для чого аналізуються параметри критичних робіт проекту.

3.1. Для скорочення вибирається критична робота з min коефіцієнтом

наростання витрат k (i, j), що має ненульовий запас часу скорочення $Z_т$ (i, j).

3.2. Час Δt (i, j), на який необхідно зжати тривалість роботи (i, j), визначається як

$$\Delta t(i, j) = \min\left[Z_T(i, j), \Delta T\right],$$

де $\Delta T = T_{кр} - T_п$ - різниця між тривалістю критичного і підкритичного шляхів в мережевій моделі. Необхідність врахування параметра ΔT викликана недоцільністю скорочення критичного шляху більш, ніж на ΔT одиниць часу. В цьому

випадку критичний шлях перестане бути таким, а підкритичний шлях навпаки стане критичним, тобто тривалість проекту в цілому принципово не може бути скорочена більше, ніж на ΔТ.

4. В результаті стиснення критичної роботи часові параметри мережевої моделі змінюються, що може привести до появи інших критичних і підкритичних шляхів. Внаслідок подорожчання прискореної роботи загальна вартість проекту збільшується на величину

$$\Delta C_{пр} = k(i, j)\Delta t(i, j).$$

5. Для зміненої мережевої моделі визначаються нові критичні і підкритичні шляху і їх тривалості, після чого необхідно продовжити оптимізацію з кроку 3.

При наявності обмеження в коштах, їх вичерпання є причиною закінчення оптимізації. Якщо не враховувати подібне обмеження, то оптимізацію можна продовжувати до тих пір поки у робіт, які могли б бути обрані для скорочення, не буде вичерпаний запас часу скорочення.

Примітка. Розглянута загальна схема оптимізації передбачає наявність одного критичного шляху в мережевий моделі. Якщо є кілька критичних шляхів необхідно або скорочувати загальну для них

усіх роботу, або одночасно скорочувати кілька різних робіт, що належать різним критичним шляхах.

Можлива комбінація цих двох варіантів. У кожному разі критерієм вибору роботи або робіт для скорочення повинен служити мінімум витрат на їх загальне скорочення.

Завдання

1. Відповідно до номеру свого варіанту отримайте наступні вихідні дані: $C_н$ (i,j) - вартість виконання роботи (i, j), яка має нормальну тривалість $T_н$ (i,j); $T_у$ (i,j) - час прискореного виконання роботи (i,j); $C_п$ (i, j) - підвищена вартість виконання роботи (i, j), яка має прискорену тривалість; $C_к$ - щоденні непрямі витрати організації, яка виконує проект; C_0 - обмеження по коштам, виділеним на проведення оптимізації.
2. Проведіть максимально можливе скорочення часу виконання проекту без урахування заданого обмеження на грошові кошти C_0.
3. Побудуйте графік прямих, непрямих і загальних витрат для проведеної оптимізації.
4. Визначте мінімально можливу тривалість виконання проекту з урахуванням заданого

обмеження на грошові кошти C_0, відобразіть прийняте рішення на графіку витрат.

5. Звіт по лабораторній роботі повинен містити:
   - номер варіанту;
   - початкові дані варіанти;
   - коефіцієнти наростання витрат робіт мережі;
   - опис кожного кроку оптимізації, а саме: критичні шляхи і їх тривалість; код скороченою роботи (робіт);
   - графік витрат.

ЛАБОРАТОРНА РОБОТА №4

Тема

Побудова проекту в MS Project

Мета роботи знайомство з програмним продуктом MS Project.

Завдання

1. Відповідно до номеру свого варіанту визначте назви роботи, їх тривалість, послідовність та залежність (відповідно до Лабораторної роботи 1).
2. Використовуючи комп'ютерну програму MS Project (або будь-який аналог), створіть відповідні роботи у вигляді окремих задач
3. Налаштуйте тривалість задач, послідовність та залежність відповідно до завдання (відповідно до Лабораторної роботи 1).
4. Створіть декілька людей в проєкті та додайте їх до створених задач
5. Збережіть отриману діаграму ганта та створіть новий проект
6. Експортуйте налаштування з першого проекту в другий
7. Налаштуйте початки робіт відповідно до результатів оптимізації з Лабораторної роботи 2

ЛАБОРАТОРНА РОБОТА №5

Тема
Критерії вибору методології розробки програмних продуктів

Мета роботи Опанувати принципи вибору методології розробки програмного забезпечення, навчитись визначати сфери застосування гнучких методології розробки програмного забезпечення, принципи організації гнучкої розробки

Теоретичні вказівки

Каскадна модель Waterfall йде строго поступово: спочатку збирається аналітика, описуються вимоги, далі розробка, тестування та запуск проекту з послідуючою підтримкою.

Аналіз втілюється в моделях, схемах і бізнес-правилах.

Дизайн - розробляється внутрішня архітектура програмного забезпечення, способи реалізації вимог. Це не тільки про інтерфейс і зовнішній вигляд ПО, а й про його внутрішню структурну логіку.

Розробка - безпосередньо пишеться код програми, йде інтеграція програмного забезпечення.

Тестування - баг-тестери (тестувальники) перевіряють фінальний продукт, заносячи в трекери відомості про дефекти коду програми або

функціоналу. У разі помилок і наявності часу/фінансів відбувається виправлення багів.

Впровадження та підтримка - продукт адаптується під різні операційні системи, регулярно оновлюється для виправлення виявлених користувачами багів і додавання функціоналу. В рамках стадії також здійснюється технічна підтримка клієнтів .

Переваги Waterfall

- зрозуміла і проста структура процесу розробки: це знижує поріг входження для команд
- зручна звітність: можна легко відстежити ресурси, ризики, витрачений час і фінанси завдяки суворій етапності процесу розробки та детальної документації проекту
- стабільність задач: задачі, які стоять перед продуктом, ясні команді з самого початку розробки, і залишаються незмінними протягом усього процесу
- оцінка вартості та термінів здачі проекту: терміни випуску готового продукту, як і його підсумкова вартість можуть бути прораховані до моменту запуску розробки.

Недоліки

- позбавлений гнучкості процес: так, якщо проект вимагає більше тимчасових і фінансових ресурсів, чим можливо, то під ніж піде фаза тестування.
- «стійкість» до змін: жорсткий каркас з етапів розробки і умова надання тільки готового продукту визначають неможливість вносити зміни під час розробки
- інерційність: на перших стадіях прогноз тимчасових і фінансових витрат може змінитися в бік збільшення, але змінити проект в сторону оптимізації витрат, зміни функціоналу або концепції до випуску готового продукту неможливо
- підвищений ризик: класична система тестування має на увазі окреме тестування кожного з компонентів проекту, в тому числі, у взаємодії з іншими. При використанні Waterfall відбувається тестування готового продукту.

Для яких проектів рекомендується використовувати Waterfall.

Середні та великі проекти.

- З декількома командами: дозволяє розділити задачі всередині проекту по декільком командам — ми маємо чітко розписані етапи, які не передбачають зміни, тож команди можуть виконувати задачі

паралельно одна до одної. Таке розподілення роботи по Waterfall дозволяє втілювати дуже великі проекти.

- Коли цілі проекту зрозумілі на початку та не передбачається суттєвих змін.
- Коли впроваджується тиражуєме рішення: системи, які відносяться до тиражуємих це - SAP, Axapta, Битрикс тощо. Тобто системи функціонування та запуск яких зрозумілий можно успішно впроваджувати за допомогою Waterfall
- Проекти в яких вимагається “fixed cost“.

Agile

Agile в управлінні проектами - це застосування принципів гнучкої розробки в проектному менеджменті з високим ступенем невизначеності і ризиків.

Цінності Agile:

- Люди та взаємодія важливіші за процеси та інструменти;
- Працюючий продукт важливіший за вичерпну документацію;
- Співпраця із замовником важливіша за погодження умов контракту;
- Готовність до змін важливіша за проходження початкового плану.

Принципи Agile Manifesto

Найвищим пріоритетом для нас є задоволення потреб замовника завдяки регулярній та ранній поставці цінного програмного забезпечення.

Зміна вимог вітається навіть у пізніх стадіях розробки. Agile-процеси дозволяють використовувати зміни для забезпечення конкурентної переваги замовника.

Працюючий продукт слід випускати якнайчастіше, з періодичністю від двох тижнів до двох місяців.

Протягом усього проекту розробники та представники бізнесу мають щодня працювати разом.

Над проектом мають працювати вмотивовані професіонали. Щоб роботу було зроблено, створіть їм умови, забезпечте підтримку - і повністю їм довіртеся.

Безпосереднє спілкування - найбільш практичний та ефективний спосіб обміну інформацією як із самою командою, так і всередині команди.

Працюючий продукт - основний показник прогресу.

Agile допомагає налагодити сталий процес розробки. Інвестори, розробники та користувачі повинні мати можливість нескінченно підтримувати постійний ритм.

Постійна увага до технічної досконалості та якості проектування підвищує гнучкість проекту.

Простота як мистецтво скоротити до мінімуму зайву роботу вкрай необхідна.

Найкращі вимоги, архітектурні та технічні рішення народжуються у команд, що самоорганізуються.

Команда повинна систематично аналізувати можливі способи покращення ефективності та відповідно коригувати стиль своєї роботи.

Завдання

1. Описати мінімум два варіанти реалізації інформаційної системи щодо обраної теми для розробки за каскадною моделлю та за гнучкими методологіями.
2. Обрати тему із індивідуальних завдань або придумати власну і узгодити з викладачем
3. Створити порівняльну таблицю в якій описати два варіанти вимог до інформаційної системи за такими критеріями: час розробки, складність проекту, архітектура системи, масштабованість, можливі зміни, користувачі і навантаження на систему, повнота вимог і завдань на розробку, особливості системи.

4. У висновку описати для якого варіанта підходить розробка за гнучкими методологіями і чому.

ЛАБОРАТОРНА РОБОТА №6

Тема

Управління проектом за методологією Скрам

Мета роботи Опанувати особливості організації процесу розробки ПЗ за методологією Скрам

Теоретичні вказівки

У перекладі з англійської Scrum - це бійка або сутичка навколо м 'яча. Термін прийшов з регбі і означає специфічну ігрову ситуацію, в якій учасники команд змикаються в три лінії з кожного боку, коли в гру вводиться м 'яч після порушення правил; задача гравців - виграти за рахунок спільних зусиль команди.

Scrum - це одна з методологій agile для управління проектами. В її основі лежать серії ітерацій фіксованою тривалості, які називають спринтами. Структура спринту складається з чотирьох складових.

Принципи Scrum

Прозорість. Робочий процес та результати роботи мають бути видимими, як для виконавців цієї роботи, так і для тих, хто отримує її результати. Прозорість надає можливість для легкої перевірки та зменшення ризиків.

Перевірка. Аби виявити потенційні проблеми, потрібно часто і ретельно перевіряти артефакти та прогрес на досягнення узгоджених цілей.

Адаптація. Будь які корегування необхідно робити якомога швидше, щоб мінімізувати подальші відхилення від цілей. Корегуванню підлягає як сам продукт так і процес розробки.

Ролі Scrum

Скрам команда складається з:

- Розробників,
- Власника продукту,
- Скрам мастера.

Артефакти Scrum представляють собою роботу або цінність роботи. Кожен артефакт включає в себе зобов'язання забезпечити надання інформації, що підвищує прозорість та фокус, і на основі яких можна виміряти прогрес:

- для Беклогу продукту це Ціль продукту;
- для Беклогу спринта це Ціль спринта;
- для Інкременту це Визначення "виконаної" роботи.

Події Scrum

- Спринт (Sprint)
- Планування спринту (Sprint Planning)
- Дейлі скрам (Daily Scrum)
- Огляд спринту (Sprint Review)
- Ретроспектива спринту (Sprint Retrospective)

Основні правила Scrum Скрам

- Команда складається тільки з власника продукту, скрам-майстра і команди розробки;
- Усі спринти повинні мати однакову тривалість; По завершенню одного спринту відразу ж починається інший;
- Кожен спринт починається з планування спринту. Команда розробки щоранку проводить дейлі скрам;
- У кожному спринті проводиться огляд спринту, щоб стейкхолдери могли надавати зворотний зв ’язок;
- Доповнювати беклог спринту під час спринту не найкраща практика.

Завдання

1. Описати основні властивості компонентів Скраму: основні задачі для Ролей, цілі для Артефактів, призначення і учасники для Подій.
2. Для індивідуального завдання з лабораторної роботи №5 скласти беклог продукту.
3. Задачі в беклозі розташувати за пріорітетністю для проєкта (замовника), та розбити на спринти. Визначити самостійно довжину спринта та склад команди.

ЛАБОРАТОРНА РОБОТА №7

Тема

Управління проектом за методологією Канбан

Мета роботи Опанувати організацією роботи команди над розробкою програмного забезпечення за методологією Канбан

Теоретичні вказівки

Kanban - система організації виробництва і постачання, що дозволяє реалізувати принцип «точно в термін». Система Канбан була розроблена та вперше у світі реалізована в компанії «Toyota».

Ключові властивості Kanban

Картки: Для кожного завдання створюється індивідуальна картка, в яку заноситься вся необхідна інформація про завдання.

Обмеження на кількість завдань на етапі: Кількість карток на одному етапі суворо регламентована. Завдяки цьому відразу стає видно, коли в потоці операцій виникає «затор», який швидко усувається.

Безперервний потік: Завдання з беклога потрапляють в потік в порядку пріоритету. Таким чином, робота ніколи не припиняється.

Постійне поліпшення («кайзен» (kaizen)): Її суть в постійному аналізі виробничого процесу та пошуку шляхів підвищення продуктивності.

Канбан базується на дуже простій ідеї: робота що виконується повинна бути обмежена і нову слід розпочинати лише тоді, коли наявна робота буде завершена Відповідні обмеження гарантують, що система не може бути перевантажена, і зможе підтримувати стійкі темпи розвитку.

З іншого боку, обмеження виявляють проблеми, що погіршують продуктивність. Коли робота не може рухатися вперед оскільки обмеження кількості роботи було досягнуто змушуючи команду не брати більше роботи до виправлення проблеми.

3 правила Kanban

Візуалізація виробничого процесу: розділення роботи на завдання, завдання на картки, картки на дошку зі стовпцями, що показують положення завдання у виробництві.

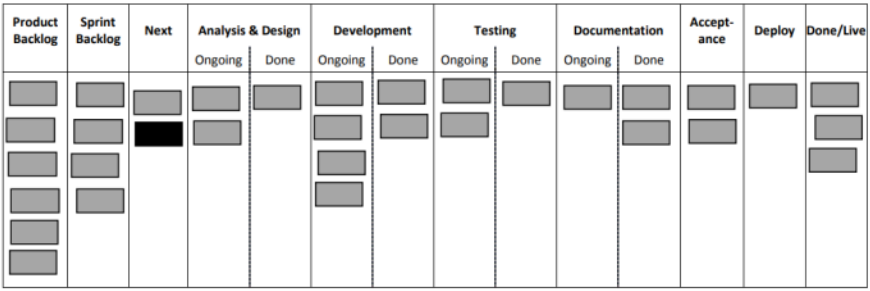
Обмеження на роботу, що виконується одночасно на кожному етапі виробництва.

Вимірювання часу циклу (середній час на виконання одного завдання) і постійна оптимізація процесу, щоб зменшити цей час.

Kanban-board може включати наступні етапи:

- Беклог продукту (Product Backlog)

- Беклог спринта (Sprint Backlog)
- Далі (Next)
- Аналіз і Дизайн (Analysis & Design)
  - В процесі | Виконано (Ongoing | Done)
- Розробка (Development)
  - В процесі | Виконано
- Тестування (Testing)
  - В процесі | Виконано
- Документування (Documentation)
- Прийняття (Acceptance)
- Впровадження (Deploy)
- Готово /Працює (Done/Live)



Проте частіше всього використовуються наступні:

- Беклог
- Аналіз (В процесі | Виконано)
- Розробка (В процесі | Виконано)
- Тестування (В процесі | Виконано)
- Прийняття

- Готово

Сильні сторони Kanban

Гнучкість планування. Команда концентрується тільки на поточній роботі, пріоритет задачі виставляється менеджером.

Висока включеність команди в процес розробки. Завдяки постійним зборам, прозорості процесів і можливостям самоорганізації працівники гуртуються і проявляють щирий інтерес.

Менша тривалість циклу. Якщо потрібний навичок має кілька людей, - тривалість скорочується, якщо ж тільки одна людина - з'являється вузьке місце. Тому співробітники повинні ділитися знаннями і тим самим оптимізувати тривалість циклу. Тоді вся команда зможе взятися за роботу, яку забуксувала, і відновити плавний потік.

Наочність. Коли всі виконавці мають доступ до даних, то вузькі місця легше помітити. Kanban-команди, крім самих карток, зазвичай використовують два загальних звіти: графіки управління і сукупного потоку.

Слабкі сторони Kanban

Система погано працює з командами чисельністю понад 5 осіб

Kanban найкраще підходить для команд, навички членів яких перетинаються один з одним.

Також Kanban краще підходить в тих випадках, коли немає жорстких дедлайнів.

Kanban не призначений для довгострокового планування.

Завдання

1. Для індивідуального завдання з лабораторної роботи №5 розробити Канбан-дошку проєкту.
2. Визначити кількість і назви стовпчиків, описати для кожного стовпця:
   - Назва
   - Що відбувається на цьому етапі
   - Ким виконується задача
   - Хто переносить задачу в наступний етап (стовпець) і за яких умов
   - Обмеження на кількість робіт на цьому етапі
3. Змоделювати процес виконання задач з першого Спринта індивідуального проєкта в розробленій Канбан-дошці для одної команди розробників.
4. Дати відповідь на питання: «На яких етапах в розробленій моделі можуть утворюватись скупчення робіт і простій із-за обмежень задач у роботі? Чому?»

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Waterfall Methodology: A Comprehensive Guide https://www.atlassian.com/agile/project-management/waterfall-methodology
2. What is scrum and how to get started https://www.atlassian.com/agile/scrum
3. What is kanban https://www.atlassian.com/agile/kanban
4. The New Methodology https://martinfowler.com/articles/newMethodology.html
5. Principles behind the Agile Manifesto https://agilemanifesto.org/iso/en/principles.html
6. Development managers vs. scrum masters https://www.atlassian.com/agile/software-development/dev-managers-vs-scrum-masters
7. Моделі мережевого планування і управління https://stud.com.ua/80815/ekonomika/modeli_merezhevogo_planuvannya_upravlinnya#927