

Менеджмент командної розробки програмного забезпечення

Практичне заняття 1. Розкриття змісту дисципліни “Менеджмент командної розробки програмного забезпечення”

Ієрархічний список проблемних питань:

1. Основний перелік
 - 1.1. Формування вимог. Як вирішити проблему нестачі сформованих вимог або невідповідність вимог до фактичних потреб ?
 - 1.2. Нестача координації в команді. Які чинники впливають на це явище та як його усунути ?
 - 1.3. Впровадження методологій розробки. Який перелік основних методологій, їх переваги та недоліки ?
 - 1.4. Нестача координації в команді. Які чинники впливають на це явище та як його усунути ?
 - 1.5. Відсутність прозорості в процесі прийняття рішень. Як приймати важливі рішення на різних рівнях ієрархії ?
 - 1.6. Доступність ресурсів. Як надати кожному учаснику команди потрібні ресурси для виконання задач ?
2. Другорядний перелік
 - 2.1. Недостаток тестування та контролю якості. Як оптимізувати процес тестування елементів продукту або продукту та покращити контроль якості ?
 - 2.2. Захист даних компанії та замовника. Як вирішити проблему витоку даних за межі команди або компанії ?
 - 2.3. Розширення команди. Які є шляхи розширення команди та які в кожного з них переваги ?
 - 2.4. Оптимізація кадрової політики. Яка саме кадрова політика може бути вигідна і в яких випадках ?
 - 2.5. Брак інструментів сумісної розробки. Як вирішити проблему вибору та надання інструментарію для команди ?

2.6. Географічне розташування команди. Як оптимізувати роботу команди, різні учасники якої знаходяться в різних точках світу ?

В ході дискусій на тему “чого ми очікуємо від курсу” кожен учасник нашої команди запропонував своє бачення результату. Погляди базуються на знаннях та попередньому досвіді кожного з учасників.

Мішак Максим

- **Доступність ресурсів. Як надати кожному учаснику команди потрібні ресурси для виконання задач ?**

Список кроків для забезпечення доступності ресурсів:

1. Аналіз потреб кожного учасника.
2. Створення індивідуальних планів роботи.
3. Забезпечення доступу до необхідних інструментів та ПЗ.
4. Надання доступу до навчальних матеріалів.

Забезпечення доступності ресурсів:

- Доступність ресурсів для кожного учасника команди є критично важливою у сумісній розробці ПЗ. Щоб вирішити цю проблему, необхідно вжити комплекс заходів:
- Аналіз потреб кожного учасника: Важливо провести спільну зустріч або опитування для визначення конкретних ресурсів, які кожен член команди потребує для ефективної роботи та навчання.
- Створення індивідуальних планів роботи: Розроблення персоналізованих робочих планів дозволить чітко визначити необхідні ресурси для кожного учасника та встановити конкретні завдання та терміни виконання.
- Забезпечення доступу до необхідних інструментів та програмного забезпечення: Необхідно впевнитися, що кожен учасник має доступ до необхідних програм, редакторів, бібліотек і інших інструментів, необхідних для ефективної роботи.
- Надання доступу до навчальних матеріалів: Забезпечення можливості навчання для кожного учасника шляхом надання доступу до необхідних навчальних матеріалів, курсів або документації.

Крім цього, важливо:

Забезпечити стабільне інтернет-з'єднання для всіх учасників, оскільки воно є ключовим для спільної роботи над проектом. Регулярно взаємодіяти з учасниками команди, щоб переконатися, що вони мають все необхідне для продуктивної роботи, і вирішити будь-які проблеми, які можуть виникнути. Забезпечити гнучкість у розгляді потреб учасників і адаптації планів, якщо змінюються обставини або вимоги проекту. Встановити спільний комунікаційний канал, де учасники можуть обмінюватися інформацією, вирішувати питання та ділитися ресурсами. Сприяти відкритій комунікації щодо проблем та питань, пов'язаних з доступністю ресурсів, і шукати разом з учасниками ефективні рішення.

- **Відсутність прозорості в процесі прийняття рішень. Як приймати важливі рішення на різних рівнях ієрархії ?**

Способи забезпечення прозорості в процесі прийняття рішень на різних рівнях ієрархії:

1. Визначення чітких процедур прийняття рішень.
2. Забезпечення відкритого обговорення.
3. Впровадження системи зворотного зв'язку.
4. Проведення регулярних зустрічей та консультацій.
5. Використання технологій для відстеження рішень.
6. Створення реєстру прийнятих рішень.

Для забезпечення прозорості в процесі прийняття рішень на різних рівнях ієрархії слід вжити комплекс заходів. Важливо ретельно розробити та документувати процедури, які визначатимуть кроки прийняття рішень на різних рівнях ієрархії. Це дозволить всім учасникам процесу розуміти, які кроки потрібно виконати. Крім того, слід стимулювати відкрите обговорення пропозицій та альтернативних варіантів рішень. Комунікація між різними рівнями ієрархії допомагає уникнути упущень та недорозумінь. Також важливо створити механізми для отримання зворотного зв'язку від учасників на різних рівнях ієрархії щодо запропонованих рішень. Це дозволяє враховувати різні точки зору та вдосконалювати прийняття рішень.

Додатково, рекомендується організовувати регулярні зустрічі та консультації між представниками різних рівнів ієрархії для обговорення ключових питань та прийняття рішень. Важливим є

також впровадження спеціалізованих програмних засобів для відстеження процесу прийняття рішень та надання доступу до цієї інформації відповідним стейкхолдерам. Не менш важливо вести реєстр ухвалених рішень, в якому будуть вказані відповідальні особи, терміни виконання та результати. Це забезпечить відкритість та відповідальність в процесі прийняття рішень.

Застосування цих практик допомагає зменшити можливі конфлікти та непорозуміння, а також підвищує довіру та співпрацю в команді.

- **Оптимізація кадрової політики. Яка саме кадрова політика може бути вигідна і в яких випадках ?**

Оптимізація кадрової політики може бути вигідною в різних ситуаціях. Наприклад, при швидкому зростанні компанії важливо мати гнучку кадрову політику, що дозволить ефективно пристосовуватися до змін. Крім того, вибір конкретної кадрової політики пов'язаний з особливостями бізнесу та його цілями.

Наприклад, компанія, яка акцентується на інноваціях та креативності, може надавати пріоритет гнучким робочим графікам та стимулювати ініціативу співробітників. Додатково, оптимізація кадрової політики може включати в себе впровадження систем мотивації та навчання співробітників. Враховуючи всі ці аспекти, оптимізація кадрової політики може бути вигідною для забезпечення ефективного управління персоналом та досягнення стратегічних цілей компанії.

- **Захист даних компанії та замовника. Як вирішити проблему витоку даних за межі команди або компанії ?**

Захист даних компанії та замовника є однією з найважливіших завдань в сучасному бізнесі. Перш за все, важливо ретельно обирати та впроваджувати надійні системи криптозахисту та шифрування для зберігання конфіденційної інформації. Додатково, слід надавати доступ до цих даних тільки авторизованим користувачам, обмежуючи права доступу відповідно до ролей та обов'язків.

Крім того, важливо регулярно оновлювати програмне забезпечення та використовувати надійні антивірусні програми для захисту від вразливостей та потенційних загроз. Для попередження витоку даних через людський фактор, слід проводити навчання співробітників щодо правил безпеки та конфіденційності даних.

Безпека даних також включає в себе вживання заходів для моніторингу та виявлення неавторизованого доступу чи спроб несанкціонованого доступу до інформації. Важливим етапом є розробка та впровадження політик безпеки, які чітко визначають процедури дій у разі виявлення порушень.

Компанія повинна також мати план реагування на інциденти, що передбачає швидке та ефективне реагування в разі виявлення витоку даних чи іншого порушення безпеки. Остаточним кроком є регулярна аудиторія та перевірка систем безпеки для впевненості в їх ефективності та відповідності сучасним стандартам безпеки даних.

Гоша Давід

- **Налагодження взаємодії різних команд**

Успішне налагодження взаємодії між різними командами є ключовим елементом для досягнення ефективності організацій. Це вимагає від організацій створення сприятливого середовища, в якому можливе взаєморозуміння та співпраця між різними відділами. Діалог та обмін ідеями між командами сприяють інноваційності та продуктивності. Для вирішення можливих конфліктів та непорозумінь, слід впроваджувати культуру взаємоповаги та взаємопідтримки, а також розвивати навички комунікації та переговорів у всіх співробітників.

Регулярні зустрічі та сесії мозкового штурму між командами можуть виявити нові можливості для співпраці та інновацій. Крім того, спільні тренінги та розважальні заходи можуть допомогти командам ліпше пізнати одна одну та розбудовувати відносини між співробітниками. Важливо також визначати та артикулювати спільні цілі та цінності, що забезпечують напрямок та мотивацію для всіх учасників організації.

- **Впровадження методологій розробки**

Методології розробки є фундаментом, на якому будується успішний процес створення програмного забезпечення. Одна з найвідоміших методологій - це Agile. Agile допомагає командам вирішувати проблеми швидко та адаптуватися до змін, забезпечуючи при цьому високу продуктивність та якість роботи. Проте, Agile може бути викликом для великих організацій через його гнучкість та динамічність.

Ще однією важливою методологією є Waterfall, котра характеризується послідовністю та структурністю. Ця методологія може бути ефективною для проектів з чітко визначеними вимогами та обмеженими змінами. Але її недоліком є відсутність гнучкості, що може ускладнити адаптацію до несподіваних змін у процесі розробки.

Scrum, як частина Agile, дозволяє командам працювати ітераційно та інкрементально, що сприяє постійному вдосконаленню процесу та продукту. Однак, для ефективного впровадження Scrum, необхідно мати досвідченого Scrum Master та зрілу команду, здатну самоорганізовуватися та саморегулюватися.

Всі ці методології вимагають від організацій визначення та артикуляції

- **Розширення команди**

Розширення команди — це необхідність, яка виникає, коли організації потрібно підвищити продуктивність та розширити обсяги роботи. Один із шляхів розширення команди — наймання нових співробітників. Це дає можливість привести свіжі ідеї та нові навички в команду, але також вимагає значних витрат часу та ресурсів на пошук, відбір та адаптацію новачків.

Інший варіант — аутсорсинг або віддалена робота. Це може бути корисним, коли потрібні спеціалізовані навички або тимчасова допомога, і може зменшити витрати та забезпечити гнучкість, але може виникнути проблема з узгодженістю та комунікацією в команді.

- **Брак інструментів сумісної розробки**

Щоб вирішити проблему вибору та надання інструментарію для команди, потрібно розробити чіткі критерії вибору інструментів, які будуть відповідати потребам команди та організації. Розглядайте інструменти, які вже використовуються в індустрії, та ті, які мають хороші відгуки та рекомендації.

При виборі інструментарію для сумісної розробки важливо зосередитися на технологіях, що підтримують колаборацію, таких як системи

управління версіями, платформи для спільної роботи, та інструменти для комунікації та управління проектами.

Після вибору інструментів важливо забезпечити належне їх впровадження, навчання та підтримку. Забезпечення доступу до ресурсів для навчання та документації допоможе співробітникам ефективно використовувати нові інструменти, а введення процесів для вирішення проблем та підтримки допоможе усунути труднощі та перешкоди на шляху ефективного використання інструментів.

Ось так, підходящі стратегії розширення команд та відповідні інструменти можуть не лише вдосконалити процес розробки, але й сприяти культурі співпраці та інноваційності в організації.

Лебедєв Євгеній

- **Нестача координації в команді. Які чинники впливають на це явище та як його усунути ?**

Нестача координації в команді може призвести до затримок у виконанні проекту та втрати часу та ресурсів на недоречні дублювання роботи між членами команди. Також відсутність координації може призвести до невірної розподілу завдань і ресурсів, що може призвести до несправедливого навантаження окремих учасників команди та втрати мотивації для спільної роботи.

Нестача координації може виникнути з різних причин. Основні чинники, що спричиняють цю проблему:

Недостатня комунікація

Відсутність зв'язку або недостатність обміну інформацією між членами команди може спричинити неузгодженість і розходження в розумінні завдань і цілей.

Відсутність чітких ролей і відповідальності

Якщо члени команди не розуміють своїх обов'язків і відповідальностей, це може вести до перекриття завдань або їх пропуску.

Неспільна методологія розробки

Використання різних методологій розробки (наприклад, Scrum і Waterfall) може призвести до конфліктів і незгоди у плануванні і виконанні завдань.

Розрізнені часові зони і віддалена робота

1. Якщо члени команди розташовані в різних географічних регіонах або працюють в різний час, це може створювати труднощі у спільній роботі та зв'язку.

Для усунення цих проблем можна:

2. Встановити регулярні зустрічі і спільні платформи для обговорення завдань і планів.
3. Встановити, хто відповідає за які аспекти проекту
4. Вибрати методологію розробки, яка найкраще відповідає потребам проекту і команди, і навчити всіх її правилам і принципам.
5. Використовувати інструменти для спільної роботи і управління проектами, щоб полегшити спільну роботу, навіть якщо члени команди працюють віддалено або в різних часових зонах.
6. Підтримувати навчання та розвиток членів команди, щоб вони були в курсі нових методів і технологій.

- **Формування вимог. Як вирішити проблему нестачі сформованих вимог або невідповідності вимог до фактичних потреб ?**

Від правильно сформованих вимог залежить успішність проекту. Нестача сформованих вимог або їх невідповідність фактичним потребам користувачів може призвести до затримок, бюджетних перевищень і незадоволеності клієнтів.

Проблема нестачі сформованих вимог або невідповідності вимог до фактичних потреб користувачів виникає через декілька факторів:

Недостатня взаємодія зі замовником

Якщо розробники не спілкуються належним чином з замовником або не розуміють його потреб, вимоги можуть бути неповними або неправильно зрозумілими.

Зміни вимог

У процесі розробки програмного забезпечення можуть змінюватися потреби і вимоги користувачів, що може призвести до невідповідності початковим вимогам.

Недостатня експертиза

Відсутність фахівців, які розуміють технічні і функціональні аспекти програми, може призвести до невірного формулювання вимог.

Для вирішення цих проблем необхідно:

Підтримувати активну комунікацію з замовником

Постійна взаємодія зі замовником допоможе зрозуміти його потреби і вимоги.

Застосовувати гнучкі методології розробки

Гнучкі методи розробки, такі як Scrum або Agile, дозволяють більш гнучко адаптуватися до змін вимог.

Проводити аналіз вимог

Детальний аналіз і верифікація вимог визначають недоліки і невідповідності, які можуть бути виправлені на ранніх етапах розробки.

Вирішення проблеми нестачі сформованих вимог або невідповідності вимог до фактичних потреб вимагає системного підходу та уважної роботи зі замовником та командою розробників.

- **Географічне розташування команди. Як оптимізувати роботу команди, різні учасники якої знаходяться в різних точках світу ?**

Географічне розташування команди з учасниками, що знаходяться в різних точках світу, може стати викликом для спільної роботи. Оптимізувати цю ситуацію допомагають використання спеціалізованих онлайн-інструментів для спільної комунікації, планування та управління проектами, а також встановлення чітких правил взаємодії та гнучкі методології розробки, сприяючи координації та забезпечуючи ефективну командну роботу незалежно від географічного розташування учасників.

Для оптимізації роботи команди, що розташована у різних частинах світу, також важливо встановити чіткі засоби звітування і визначити години роботи, які враховують часові пояси всіх учасників. Крім цього, постійна комунікація та розуміння особистих особливостей робочого процесу кожного члена команди допомагають зберегти сплоченість та досягнути спільних цілей, незважаючи на відстань між ними.

- **Недостаток тестування та контролю якості. Як оптимізувати процес тестування елементів продукту або продукту та покращити контроль якості ?**

Недостаток тестування та контролю якості може призвести до виходу на ринок продукту з численними помилками та недоліками. Для оптимізації цього процесу і покращення контролю якості, важливо впровадити автоматизоване тестування, яке дозволить ефективно виявляти та виправляти помилки на ранніх етапах розробки. Потрібно створити детальну стратегію тестування, яка враховує всі аспекти продукту і забезпечує високу якість на кожному етапі розробки.

Додатково, для оптимізації процесу тестування і контролю якості рекомендується надавати пріоритет тестуванню функцій та компонентів, які найбільше впливають на користувацький досвід та безпеку програмного продукту. Також потрібно забезпечити взаємодію між розробниками і тестувальниками для швидкого виправлення виявлених проблем і вдосконалення якості продукту під час розробки.