

DOI: 10.18372/2310-5461.40.13267

УДК 004.7.052(045)

О. О. Писарчук, д-р. техн. наук, проф.
Національний авіаційний університет
orcid.org/0000-0001-5271-0248
e-mail: PlatinumPA2212@gmail.com;

Д. В. Котенко
Національний авіаційний університет
orcid.org/0000-0003-2955-1319
e-mail: dkote@softserveinc.com;

Ткачук Д. О., старш. викладач
Державний університет інфраструктури та технологій
orcid.org/0000-0002-6235-8659
e-mail: bog2603@ukr.net;

Ганношина І. М., старш. викладач
Державний університет інфраструктури та технологій
orcid.org/0000-0001-5810-2462
e-mail: gannoshina im@softserveinc.com

ТЕХНОЛОГІЯ КОЛЕКТИВНОЇ РОЗРОБКИ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ КЕРІВНИКА АПТЕЧНОЇ МЕРЕЖІ

ВСТУП

Розвиток інформаційних технологій та сфери інженерії програмного забезпечення останніх років привів до актуалізації напряму колективної розробки складних програмних систем. Однак, у кожному конкретному випадку, технологічні підходи встановлюються типом програмного забезпечення, що розробляється, та його функціональним наповненням. Особливим класом програмних засобів, у цьому сенсі, є системи підтримки прийняття рішень (СППР). Актуальність процесу проектування та розробки СППР полягає в забезпеченні їх створення з високою якістю в короткі терміни та з обмеженим об'ємом ресурсів. Крім того зазвичай системи розробляються командою від трьох до п'ятнадцяти осіб, що включає в себе різний ступінь кваліфікації працівників та різні області відповідальності разом із навичками різних технологій. Зазначене робить унікальним технологічні процеси колективної розробки СППР.

Постановка проблеми

У розробці СППР задіяні учасники з різними ролями, так завжди є дві основні категорії учасників: розробники та замовники. Замовники висловлюють свої побажання та бачення майбутньої системи, тоді як розробники реалізують конкретні прикладні рішення.

Замовники зазвичай є власниками бізнесу або державними установами, що можуть оптимізувати внутрішні процеси.

Для спрощення аналізу вимог замовників та їх фіксації, виділяється окрема роль бізнес аналіти-

ка. Щоб упевнитись у відповідній якості продукту залучаються інженери з підтримки якості (QA інженери). У міру зростання команди залучається також менеджер проекту, а у міру складності проекту та конфігураційних питань — менеджер конфігурації (DevOps). Як бачимо на звичайний проект залучається велика кількість спеціалістів з різними ролями, тому виникає значна кількість проблем під час розробки, що потребує їх творчого розв'язання.

Мета статі — розроблення технології колективної розробки СППР та виділення ключових найкращих підходів, що дозволять максимально ефективно створювати СППР будь якого масштабу.

Колективні рішення побудови систем

Сьогодні гостро постає питання з вибором технологій які будуть застосовані при розробці СППР. Так, наприклад, є особлива необхідність в розробці системи з підтримкою веб-браузерів, мобільних телефонів із різними операційними системами (Android та IOS) та навіть настільних версій системи для Windows та Mac OS. Таким чином, розробка проводиться на різних мовах і з залученням розробників, які відповідають за різні області.

Так виділяють *front-end* спеціаліста — завдання якого є розроблення веб-клієнта з використанням браузерних технологій та фреймворків (HTML, CSS, JavaScript, Type Script, Angular, React).

У СППР завжди присутня серверна сторона яка, також може розроблятися на різних мовах

(C#, Java, Python, Node JS) та зберігати дані в різних сховищах такі як реляційні бази даних (MS SQL Server, Oracle, My SQL, Postgre SQL) та No SQL базами даних таких як Maria DB, Mongo DB та ін.

З виникненням таких різноманітних технологій складно знайти хороших спеціалістів, тому дехто вдається до реалізації архітектури у бік сервісно-орієнтованої (Service Oriented Architecture), приклад такої архітектури наведено на рис. 1.



Рис. 1. Приклад сервісно-орієнтованої архітектури

Таким чином, наприклад серверна частина може бути реалізована з використанням різних технологій, зокрема .Net та Node JS. Великим мінусом такого підходу є складність підтримки такої системи, її розгортання та комунікація і швидкодія, відповідно зростає кількість необхідних ресурсів, які потрібні для подальшої системи [1]. Тому зазвичай обирають тільки одну мову для серверної частини, одну для клієнтів та одну для написання мобільних клієнтів. На сьогодні існує безліч засобів для покращення колективної розробки систем. Обов'язковою умовою без якої не може обійтись жоден проект, у якому задіяно щонайменше два розробника — це системи контролю версій (СКВ).

Найбільш відома і зручна є Git СКВ. За допомогою можна чітко відслідковувати зміни які відбувались в проекті, колективно вносити зміни в вихідний код та потім їх об'єднувати, можливість побачити стан вихідного коду на певний момент та його автора [2].

Вимоги до системи постійно змінюються під час розробки, тому забезпечення гнучкості до внесення змін в проект є основною необхідністю для подальшого самого життя проекту. Гнучкість при внесенні змін можна задовільнити виконавши чотири основні чинника: проведення розробки функціоналу короткими ітераціями, розробка гнучкої модульної архітектури, покриття коду тестами та його постійний рефакторинг.

Гнучкі методології розробки такі як Scrum та Kanban дозволяють розробляти функціонал короткими ітераціями які тривають від тижня до чотирьох, унаслідок чого буде постійно постачатись робочий функціонал [3].

Системи, у яких не продумана гнучка архітектура, зазвичай не виживають через декілька років експлуатації.

У міру того, що внесення нової функціональності буває просто неможливе, а продуктивність системи просто неприйнятною.

Покриття коду тестами, особливо через підхід Test Driven Development (TDD) дозволить бути впевненим, що з часом, а також при внесенні нових змін будуть дотримуватись всі поставлені вимоги до системи [4]. Сам вихідний код системи з часом загниває та стає важким для сприйняття, що спричиняє складність підтримки, тому необхідно постійно покращувати його якість. Для планування дій розробників зручно використовувати утиліти по управлінню життєвим циклом розробки такі як Jira чи You Track. Дані засоби здатні розділити запланований об'єм роботи на задачі та розподілити між учасниками команди, після чого відслідковувати прогрес та завершеність завдань.

Також хорошою практикою є створення процесів безперервної інтеграції (Continuous integration — CI) та безперервної доставки (continuous delivery — CD), що здатні забезпечити автоматизований процес збірки системи та подальшої її розгортання в інфраструктурі замовника без зайвих зусиль. При такому процесі автоматично збирається збірка, виконуються скрипти міграції баз даних та запускається зібраний пакет на кінцевому серверному компоненті (рис. 2).

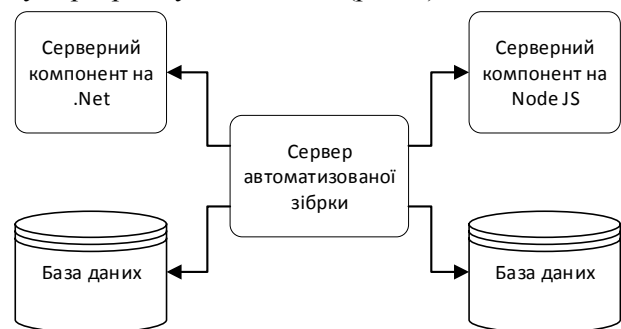


Рис. 3. Приклад CI/CD процесів

Зазвичай для такої роботи наймається окрема людина — менеджер конфігурації (DevOps) або при обмеженому бюджеті цим займаються безпосередньо розробники.

Хорошою практикою для студентів по вирішенню питань з колективної розробки СППР є курс лабораторних робіт з предмету Research Of Software Products and Technologies, що вивчається в Національному авіаційному університеті. Так студенти невеликими командами власними силами реалізують СППР керівника аптечної мережі.

Команди складаються з п'ятьох осіб які мають різні навички в програмуванні так звана крос-функціональна команда. В цих командах зазвичай розподіляють обов'язки так: одна людина займа-

ється створенням клієнтського застосунку, три розробника серверної частини та один архітектор — розробляє базову архітектуру та бере участь в безпосередній розробці, а також виступає як про-

ектний менеджер та лідер команди. Оскільки команди кросфункціональні та учасники яких знайомі з різними технологіями вибудовується сервісно-орієнтована архітектура (рис. 4).

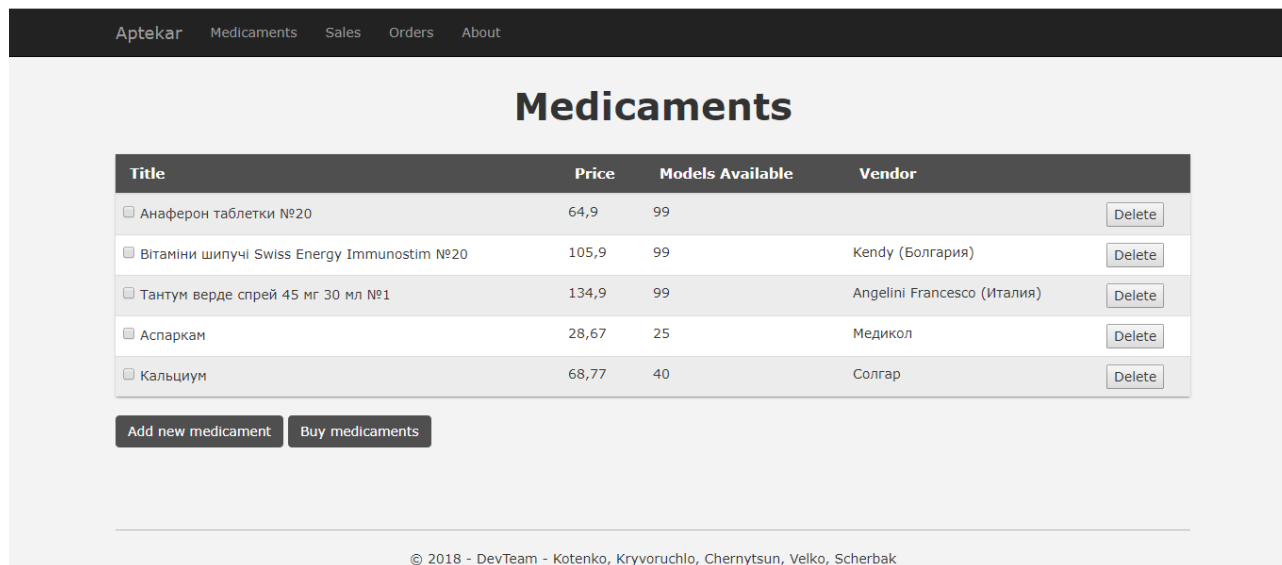


Рис. 4. Скріншот роботи системи підтримки прийняття рішення аптечної мережі

Стек *технологій розробки системи підтримки прийняття* та відповідальності розподілені на прикладі однієї з команд виглядає так: основна частина логіки системи реалізується на одному сервісі з використанням .Net технологій та взаємодії з базою даним MS SQL Server, що розробляється двома розробниками з використанням середовища розробки Visual Studio 2017, окремо будується сервіс на технології Node JS та середовищі розробки Intelij IDEA, що може містити власну базу даних та виконує певну функціональність, наприклад, по прогнозу кількості препаратів які необхідно закупити.

Одна людина яка займається розробкою графічного інтерфейсу та комунікацією з сервером з використанням веб технологій (HTML, CSS, Java Script) в середовищі Visual Studio Code.

Таким чином, студенти на практиці беруть участь в розробці системи, розподіляють обов'язки та завдання між собою, проектують архітектуру та реалізують поставлені задачі. Такий практичний підхід до засвоєння навчального матеріалу є хорошим для майбутніх інженерів-програмістів так як дозволяє отримати практичні навички, досвід по роботі в команді та бути конкурентоспроможними на ринку праці після отримання навчального ступеню.

Висновки

В статті висвітлено ключові проблеми з якими стикаються при розробці СППР кросфункціональними командами та представлені засоби для покращення та здешевлення розробки і супроводу на прикладі розробки командою СППР керівника аптечної мережі. Описано технологічний підхід для розробки СППР коли колектив розробників має різні навички роботи з автоматизованими середовищами колективної розробки програм, а також коли володіють різними мовами програмування. Дані підходи та засоби здатні суттєво покращити якість розроблюваного продукту, покращити комунікацію та коригувати спільні дії та розроблювати функціональність в заплановані терміни.

ЛІТЕРАТУРА

1. **Мартин Р.**, Мартин М. Принципы, паттерны и метрики гибкой разработки на языке C#. Пер с. англ. СПб: Символ-плюс, 2011. 768с.
2. Вступ - Про систему контролю версій. URL: <https://git-scm.com/book/uk/v2/Вступ-Про-систему-контролю-версій> (дата звернення 25.09.2018)
3. **Эндрю С.**, Дженнифер Г. Постигаая Agile. Ценности, принципы, методологии. Пер с. англ. СПб: Манн, Иванов и Фербер, 2017. 573с.
4. **James Bender**, Jeff McWhether Professional Test Driven Development with C#. Wiley Publishing, Inc. 327 p.

**Писарчук О. О., Котенко Д. В., Ткачук Д. О., Ганношина І. М.
ТЕХНОЛОГІЯ КОЛЕКТИВНОЇ РОЗРОБКИ СИСТЕМИ ПІДТРИКИ ПРИЙНЯТТЯ РІШЕНЬ
КЕРІВНИКА АПТЕЧНОЇ МЕРЕЖІ**

Існують проблеми, що виникають при розробці системи підтримки прийняття рішень, яка включає у тому числі спільну роботу команди, управління й хід роботи, безперервну інтеграцію, розробку проекту декількома людьми і роботу над єдиним початковим кодом. Ці проблеми з'являються в кожній команді і заважають правильній роботі, тому підходи до їх рішення повинен знати кожен інженер-програміст. У статті описується, як вирішити ці проблеми на прикладі групової розробки проекту під назвою «Система підтримки прийняття рішень для менеджера аптечної мережі».

Мета роботи — аналіз існуючих практик колективної розробки систем підтримки прийняття рішень і визначення передових практик, що дозволяють спільною роботою команди створити найбільш ефективні системи підтримки прийняття рішень будь-якої складності.

Для системи підтримки прийняття рішень дуже важливе питання вибору технологій розробки. Тобто необхідно розробити систему для використання у веб-браузері і мобільних телефонах і ноутбуках з різними мобільними операційними системами — Android, IOS, Windows або Mac OS. Як наслідок, розробка ведеться з використанням різних мов програмування і залучення фахівців з різними навичками.

Отже, ця стаття показує ключові проблеми розвитку багатифункціональної команди і визначає основні шляхи їх запобігання, а також підвищення продуктивності команди.

У статті освітлюються ключові проблеми, що виникають при розробці систем підтримки прийняття рішень міжфункціональними командами, а також представлені інструменти і підходи для поліпшення і зменшення витрат на розробку і супровід на прикладі менеджера аптечної мережі. Описаний підхід до розробки систем підтримки прийняття рішень з командою розробників, які мають різні знання технологій і мов програмування і різні навички роботи з автоматизованими середовищами. Ці підходи і інструменти можуть значно поліпшити якість продукту, що розробляється, поліпшити взаємодію усередині команди, скоректувати завдання і розвинути функціональність в заплановані терміни.

Ключові слова: система підтримки прийняття рішень; міжфункціональна команда; колективна розробка; сервіс-орієнтована архітектура; зборка програмних систем через механізми.

**Pysarchuk O. O., Kotenko D. V., Tkachuk D. O., Hanoshina I. M.
TECHNOLOGIE OF TEAM COLLABORATION OF DEVELOPMENT OF DECISION SUPPORT
SYSTEM FOR MANAGER OF PHARMACY**

There are some problems that arise during development of the decision support system that includes team collaboration, management and progress of the work, continuous integration and continuous delivery processes, development of the project by several people and working on single source code. This problem appears in every cross-functional team and interfere of proper work, so approaches for solving them must to know every software engineer. At this topic describes how this problem can be solved using example of team development of the project called “Decision support system for manger of pharmacy network”.

The purpose of this paper is to analyze existing practices for the collective development of decision support systems and to identify key best practices that will allow the most efficient creation of decision support system at any scale with team collaboration.

By now is very important question about choosing technologies for development of decision support system. So there is a high necessity in developing such system to be used in web browser and mobile phones with various mobile operation system like Android or IOS besides even for laptop OS like Windows or Mac OS. As follows development is conducted using different programming languages and engagement of the specialists with different skills.

As the result this topic shows key problems of the cross-functional team development and defines main ways of their prevention and also increase team productivity.

In this article highlights the key issues encountered in the development of the decision support systems by cross-functional teams, and presents tools and approaches for improving and reducing the cost of the development and maintenance using example of the manager of the pharmacy network. An approach is described of the development of decision support systems with the team of developers that have different knowledge of technologies and programming languages and different skills of working with automated environments. These approaches and tools can significantly improve the quality of the product that is developing, improve communication inside team, adjust tasks and develop functionality within the scheduled time frame.

Keywords: decision support system; crossfunctional team; collective development; service-oriented architecture; assembly of software systems through mechanisms.

Писарчук А. А., Котенко Д. В., Ткачук Д. А., Ганношина И. Н.
ТЕХНОЛОГИЯ КОЛЛЕКТИВНОЙ РАЗРАБОТКИ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ РУКОВОДИТЕЛЯ АПТЕЧНОЙ СЕТИ

Существуют проблемы, возникающие при разработке системы поддержки принятия решений, которая включает в том числе совместную работу команды, управление и ход работы, непрерывную интеграцию, разработку проекта несколькими людьми и работу над единым исходным кодом. Эти проблемы появляются в каждой команде и мешают правильной работе, поэтому подходы к их решению должен знать каждый инженер-программист. В статье описывается, как решить эти проблемы на примере групповой разработки проекта под названием «Система поддержки принятия решений для менеджера аптечной сети».

Цель работы — анализ существующих практик коллективной разработки систем поддержки принятия решений и определение передовых практик, позволяющих совместной работой команды создать наиболее эффективные системы поддержки принятия решений любой сложности.

Для системы поддержки принятия решений очень важен вопрос выбора технологий разработки. Т. е. необходимо разработать систему для использования в веб-браузере и мобильных телефонах и ноутбуках с различными мобильными операционными системами — Android, IOS, Windows или Mac OS. Как следствие, разработка ведется с использованием разных языков программирования и привлечения специалистов с разными навыками.

Статья показывает ключевые проблемы развития многофункциональной команды и определяет основные пути их предотвращения, а также повышения продуктивности команды.

Освещены ключевые проблемы, возникающие при разработке систем поддержки принятия решений межфункциональными командами, а также представлены инструменты и подходы для улучшения и снижения затрат на разработку и сопровождение на примере менеджера аптечной сети. Описан подход к разработке систем поддержки принятия решений с командой разработчиков, которые имеют различные знания технологий и языков программирования и различные навыки работы с автоматизированными средами. Эти подходы и инструменты могут значительно улучшить качество разрабатываемого продукта, улучшить взаимодействие внутри команды, скорректировать задачи и развить функциональность в запланированные сроки.

Ключевые слова: система поддержки принятия решений; межфункциональная команда; коллективная разработка; сервис-ориентированная архитектура; сборка программных систем через механизмы.

Стаття надійшла до редакції 15.11.2018 р.
Прийнято до друку 05.12.2018 р.