

ІНТЕЛЕКТУАЛІЗАЦІЯ ПРОГРАМНИХ СИСТЕМ ДЛЯ ФОРМУВАННЯ ЕФЕКТИВНИХ ГРУП РОЗРОБНИКІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Розглянуті сучасні принципи та методи відбору кандидатів для великих програмних проектів, а також їх практичне застосування в умовах сучасних програмних проектів. Проаналізовані наукові принципи формування та управління виробничими колективами, основні професійні, індивідуальні, психофізіологічні вимоги до розробників програмного проекту, розроблені структурні та функціональні моделі програмного експертного комплексу формування груп розробників. Також запропоновані алгоритми функціонування програмних модулів інтелектуальної системи. Приведені підходи до розробки структури бази даних і бази знань інтелектуальної системи. Сформульовані вимоги до програмного проекту зі сторони промислового середовища. Розглянуті заходи по впровадженню програмного проекту у експлуатаційне середовище. Удосконалені методи та алгоритми формування груп розробників для створення ефективних програмних продуктів. Оптимізовані існуючі функціональні схеми із підбору кадрів, приведені рекомендації до оптимізації роботи відділу кадрів та менеджерів із відбору персоналу. Розроблені критерії відбору, мотиваційні заходи, які збільшують ефективність командної розробки та якість програмного продукту у межах промислового середовища. Розроблено автоматизовану інтелектуальну систему відбору кандидатів розробників для створення сучасного ефективного програмного забезпечення. Практичне значення одержаних результатів полягає у використанні розробленої інтелектуальної комп'ютерної системи для формування ефективних колективів розробників великих програмних проектів.

Проблема і її зв'язок з науковими і практичними завданнями. На сьогодні розвиток бізнесу та технологій вимагає від розробників програмного забезпечення високої якості в межах існуючого бюджету та відведеного на те часу. У створенні великих програмних продуктів беруть участь різні фахівці, які, як правило, об'єднуються в колективи (команди). Колективи включають у себе співробітників організації розробника та замовника, субпідрядників та тимчасово залучених співробітників. Учасники колективу розробників програмного забезпечення можуть територіально знаходитися в різних місцях, або розміщуватися у межах одного приміщення. Команду можна охарактеризувати за певними ознаками ефективності, які притаманні для будь-якої організаційної робочої групи. Проте існують деякі характерні риси, які присутні лише команді. А саме, перш за все – це направленість усієї команди на кінцевий результат. Ініціативність та творчий підхід, висока продуктивність та орієнтованість на найкраще рішення серед всіх наявних, активне обговорення та вирішення нагальних проблем, наявність ефективних засобів комунікації між членами команди, а також – вільного доступу до необхідної інформації.

Раціональна організація процесів розробки програмного забезпечення, а також великих програмних проектів описується у міжнародних, державних, корпоративних стандартах – методології розробки програмного забезпечення. Методології розробляються провідними виробниками програмних систем та їх командами розробників з врахуванням певних особливостей програмного забезпечення. Особливості залежать від сфери застосування даного програмного продукту.

У наш час все більшого поширення в українських організаціях та на великих підприємствах отримують гнучкі методології розробки програмного забезпечення, та великих програмних проектів, у яких основна увага зосереджена на створенні якісного продукту, а не на підготовку документації за проектом. Великий акцент, при цьому, робиться на організацію ефективного управління командою розробників даного програмного забезпечення.

Найефективнішими методиками командної розробки, які використовуються по відношенню до великих програмних проектів та програмного забезпечення є наступні методики.

Корпоративна методологія розробки програмного забезпечення. Методологія описує керування людьми та робочими процесами при розробці інноваційних рішень, які спрямовані на задоволення бізнес-потреб замовника. Головною особливістю командної моделі є відсутність офіційного лідера. Усі учасники колективу розробки програмного забезпечення, проекту мають однаковий рівень зацікавленості, а також направлені на досягнення високих результатів. Сьогодні дана методологія використовується у продуктах Microsoft Solutions Framework (MSF) і

представляє собою узагальнення кращих проектних практик, що використовуються командами розробників Microsoft. MSF має набір рольових кластерів, які спрямовані на досягнення якісного результату в цілому.

Гнучка методологія розробки програмного забезпечення та великих програмних проектів – це методологія, яка орієнтується на використання ітеративного підходу, за яким програмний проект створюється поступово, невеликими частинами, які включають у себе реалізацію певного набору вимог. Команди за гнучкої методологією формуються з універсальних розробників, які здатні виконувати різні задачі в процесі створення програмного продукту. При залученні гнучких методологій мінімізація ризиків здійснюється шляхом зведення розроблення до серії коротких циклів – ітерацій.

Завдяки концепції наукового управління, сьогодні виділяють окрему ланку наукових досліджень з управління персоналом у великих програмних проектах та на великих підприємствах.

Методи організації праці колективу, який працює над програмним проектом, розділяють на декілька груп.

Метод розподіленої відповідальності. Під даним методом розуміють розділення сфер відповідальності між членами колективу, спрямоване на досягнення поставлених цілей проекту. Кожна із сфер вирішує задачі, розділені за методами їх вирішення (рис. 1).

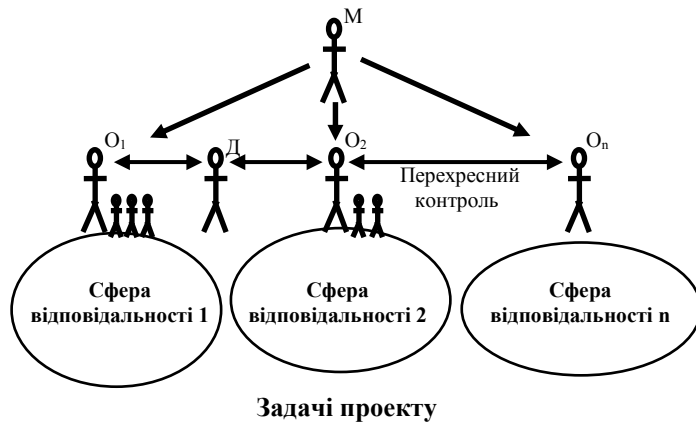


Рис. 1. Схема з розподілом відповідальностей та підвищеною надійністю

Деперсоніфікований метод. Розподіл колективної праці за деперсоніфікованим методом передбачає, що проектні рішення приймаються колективно. Кожен учасник групи вникає у всі ключові задачі та цілі проекту, та може кваліфіковано аргументувати вибір прийнятих проектних рішень. Відповідальність за компоненти проекту розповсюджується на кожного члена команди, незалежно від автора

компоненту (рис. 2).

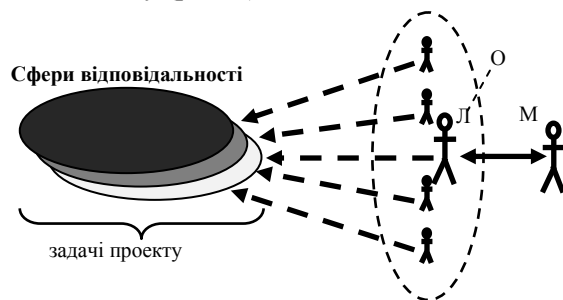


Рис. 2. Схема деперсоніфікованого методу

Метод змішаного планування та організації колективної роботи. Передбачає ієрархічний розподіл задач менеджменту між колективами виконавців проекту, у яких вибудовується власна самостійна організація колективної діяльності. Колективи розробників програмного забезпечення, повинні мати окрім професійних навичок вузької спеціалізації, технічні, а також індивідуальні

якості, такі як: абстрактне мислення, системне, динамічне сприйняття, зосередженість та цілеспрямованість.

Головною складовою розробки та впровадження успішного проекту є наявність професійної, злагодженої команди, яка, за умови відповідального керування, може досягти виконання поставлених задач та цілей, у короткий проміжок часу, для отримання максимальних прибутків підприємством. Якісний підбір учасників команди є відповідальним кроком для кожного проекту.

Аналіз досліджень та публікацій. Еріх Гамма зазначав: «ключ до своєчасної поставки продукту - не процеси, а люди» [1]. Цілеспрямованість та самоорганізація команди розробників проекту дозволяє створювати високоякісні програмні продукти, а також розробляти великі програмні проекти у стислі терміни. Командна організація спрямована на підвищення ефективності роботи організації, підприємства в цілому.

Отже, команда - це сформоване за певними ознаками, правилами колективне товариство, яке має необхідний потенціал, що дозволяє ефективно досягати поставлених цілей із гармонійним поєднанням усіх інтересів та пропозицій його членів.

Розглянемо найефективніші парадигми командної розробки, які використовуються по відношенню до великих програмних проєктів [1,2].

Корпоративна методологія розробки програмного забезпечення. Ця методологія описує керування людьми та робочими процесами при розробці інноваційних рішень, які спрямовані на задоволення бізнес-потреб замовника. Сьогодні дана методологія використовується у продуктах Microsoft Solutions Framework (MSF) і представляє собою узагальнення кращих проєктних практик, що використовуються командами розробників Microsoft. Дана методологія входить до великого програмного комплексу для професійних розробників - Visual Studio Team System. Visual Studio Team System представляє собою інструмент, що дозволяє створювати програмні системи різного призначення в командах з різними підходами до управління процесами розробки програмного забезпечення [2].

До основних принципів MSF належать:

гнучкість - готовність до змін, яка забезпечує можливість уточнення та зміни вимог в процесі розробки програмного забезпечення, та оперативного, своєчасного реагування на поточні зміни проєкту при постійній ефективності управлінської діяльності;

єдине бачення проєкту, що передбачає розуміння усіма зацікавленими особами задач та цілей створення програмного проєкту;

заохочення до спілкування у вільному форматі, яке передбачає відкритий, чесний обмін інформацією, як у середині команди, так і з головними відповідальними особами в цілому;

концентрація на пріоритетах бізнесу – передбачає створення продукту (програмного проєкту) високої якості для споживача, та формування певної нагороди, вигоди для організації у якості одержання прибутку.

У MSF є певний набір рольових кластерів, які спрямовані на досягнення якісного результату в цілому. Кластер представляє собою унікальну точку зору на проєкт [3].

Розглянемо кластери MSF:

управління продуктом - основна задача кластеру забезпечити, у замовника задоволеність результатом;

управління програмою - забезпечує відстеження планів виконання, відповідальність за бюджет, вирішення труднощів при розробці продукту, мінімізація перешкод;

розробка - кластер забезпечує розробку додатку;

тестування - виконується тестування програмного проєкту;

задоволення споживача - розробка дизайну інтерфейсу користувача, забезпечення легкості в експлуатації, створення документації;

управління випуском - передбачає функціонування рішень проєктної групи;

архітектура - організація та виконання проєктування рішення, створення та управління специфікаціями програмного забезпечення, проєкту.

Гнучка методологія розробки програмних проєктів орієнтується на використанні ітеративного підходу, за яким програмний проєкт створюється поступово, невеликими частинами, які включають у себе реалізацію певного набору вимог. Передбачається, що вимоги мають змінний характер. Команди за гнучкої методологією формуються з універсальних розробників, які здатні виконувати різні задачі в процесі створення програмного продукту. При залученні гнучких методологій мінімізація ризиків здійснюється шляхом зведення розробки до серії коротких циклів – ітерацій. Ітерація представляє собою набір задач, запланованих на виконання в певний період часу. В кожній ітерації створюється працюючий варіант програмної системи, у якій реалізуються найбільш пріоритетні вимоги замовника, на даному етапі розробки [4].

Засновником та розробником ідей наукових принципів формування та управління колективами є Фредерик Уїнстон Тейлор [5, 6]. Головна мета системи Тейлора – забезпечити максимальний прибуток власнику підприємства, змушуючи працівників працювати на рівні надможливостей, шляхом раціоналізації трудових ресурсів.

До основних положень наукового управління Тейлора можна віднести наступні:

нормалізація методів та засобів роботи;

виконання робітником лише тієї роботи, на яку він здібний;

відбір персоналу, який проводиться на основі їх здібностей до навчання та тренінгу, знання точних інструкцій робітником;

облік та контроль за всіма видами робіт на підприємстві.

Наступним вченим, що досліджував принципи наукового управління є Гант. Згідно ученням Ганта, головними відмінностями між системами сьогодення та старими системами є наявність способів планування розподілом задач з системою мотивації співробітників. Гант виступає засновником системи заробітної плати з елементами почасової оплати, мотивуючи працівників до перевиконання поставлених задач, тим самим скорочуючи час на виконання проекту в цілому [6].

Методи організації праці колективу програмного проекту прийнято розділяти на декілька груп. Метод розподіленої відповідальності. Під даним методом розуміють розділення сфер відповідальності між членами колективу, спрямоване на досягнення поставлених цілей проекту. Кожна із сфер вирішує задачі, розділені за методами їх вирішення [7]. Розбиття колективу на тематичні групи - головний обов'язок менеджера проекту. Сфери відповідальності розділяють між виконавцями проекту, в залежності від їх кваліфікаційних навичок, рівня зайнятості та індивідуальних якостей.

Ризики невиконання задач більші, ніж при чітко регламентованому розподілі відповідальностей. Недоліки методу компенсуються завдяки мобільності та дружності колективу [8]. Метод змішаного планування та організації колективної роботи передбачає ієрархічний розподіл задач менеджменту між колективами виконавців проекту, у яких вибудовується власна самостійна організація колективної діяльності [9].

Більшу кількість розробників програмного забезпечення умовно розділяють на дві групи: «Інспектор» та «Аналітик».

Люди за типом «Інспектор» мають добре розвинуті наступні властивості:

спокійний, наполегливий тип характеру, справедливий;
рішучість до конкретних дій у критичних ситуаціях, традиційні;
працьовитість, та наполегливість у роботі, оперують конкретними даними, інформацією;
гарно розуміються на вирішенні складних завдань, та їх реалізації.

Особливості якостей розробників за типом «Аналітика»:

самовпевнені люди, вірять у власні можливості. Живуть власним життям, концентруючись на можливостях;

проявляють не аби який інтерес до майбутнього, вміло застосовують теоретичні знання;

покладаються лише на власні розсуди, незважаючи на думки оточуючих;

відстоюють власні інтереси, думки, ігнорують думки інших, що може нашкодити подальшому кар'єрному зростанню [10].

Для ефективної організації діяльності проектної команди необхідно визначити типи сумісності командної праці [11].

Постановка завдання. На основі проведеного аналізу сформовано мету роботи: розробити комп'ютерну систему комплектування груп розробників для реалізації ефективних програмних проектів. Для досягнення мети у необхідно вирішити наступні задачі: розробити математичну модель, структуру та алгоритми програмного комплексу автоматизації формування команди розробників програмного забезпечення; розробити структурну та функціональну моделі програмного комплексу формування груп розробників; розробити алгоритми функціональних модулів програмного комплексу; розробити структурну модель баз даних параметрів відбору розробників; розробити базу знань для прийняття рішень при формуванні ефективних груп, команд розробників; розробити програмні заходи автоматизації формування ефективних груп розробників програмного забезпечення; розробити бази даних, бази знань програмного комплексу; розробити програмні модулі автоматизованої системи; розробити інтерфейс користувача програмного забезпечення; сформулювати вимоги до програмному проекту; сформулювати ефективну команду розробників програмного забезпечення.

Викладення матеріалу та результати. Програмний модуль представлений у вигляді частини програми, яка має власні завдання, функції та алгоритми обробки даних.

Частини програми пов'язані між собою інформаційними зв'язками.

Модульність забезпечує внутрішню структуру, полегшує завдання управління у межах великих програмних проектів. У розробці програмних модулів беруть участь колективи розробників із десяти і більше чоловік [10-12].

Структуризація використовується у першу чергу для зручності розробки, програмування, впровадження та внесенням змін у проект.

Загальна структурна схема програмного комплексу зображена на рис. 3.

Інтерфейс користувача має чітку структуровану систему.

Система включає у себе набір форм, елементів управління, кнопок, тощо.

Розробка інтерфейсу користувача є одним з головних етапів розробки програмного проекту.

Правильний підхід до функціональності та адаптуванню інтерфейсу під потреби користувача системи є головними пріоритетами при створенні програмного проекту.

Розроблений інтерфейс користувача складається з наступних форм:

головна форма;

форма анкетних даних кандидатів;

форма індивідуальних ознак кандидатів;

форма професійних ознак кандидатів;

форма психологічних ознак кандидатів;

форма звітності;

допоміжні, службові форми.

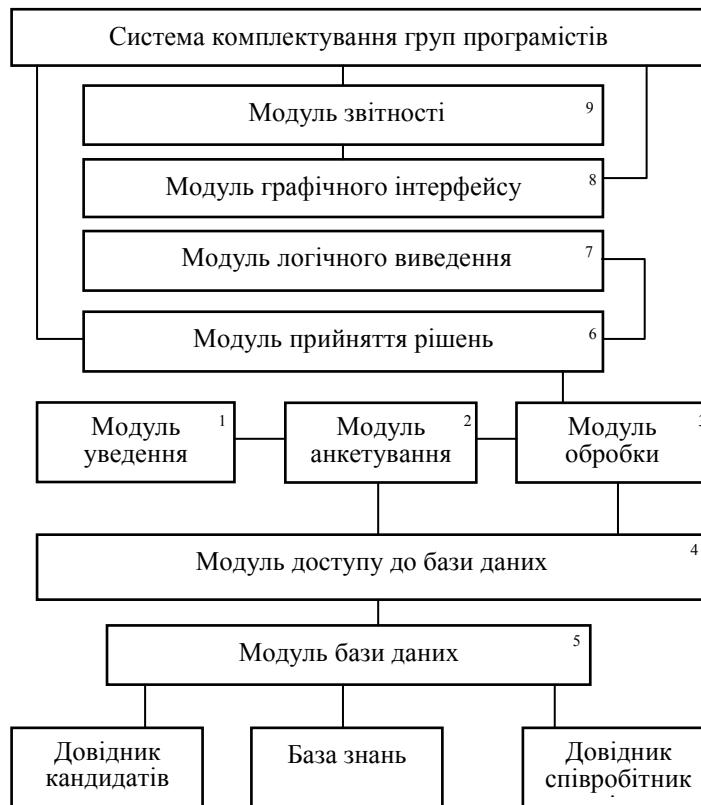


Рис. 3. Загальна структурна схема комплексу формування груп розробників програмного забезпечення

Під базою знань розуміють наступне визначення. База знань – це сукупність моделей, правил і фактів (даних), на основі яких аналізують існуючу інформацію та приймають відповідні висновки щодо знаходження правильних рішень складних завдань в деякій предметній області. Виділені і організовані у вигляді окремих, цілісних структур інформаційного забезпечення знання про предметну область стають явними і відокремлюються від інших типів знань. Бази знань дозволяють виконувати міркування не лише на основі формальної логіки, а й на основі досвіду, фактів, евристик, тобто вони більш наближені до людської логіки [12, 13]. База знань зберігається у вигляді текстового файлу, який містить набір фактів і правил, які базуються на основі формальної логіки та досві-

ду експерта у даній сфері діяльності. Файл бази знань має структурований вигляд. Детально структуру файлу бази знань розглянуто нижче. Файл бази знань представлений так

<OR><Україна><Польща><Вибір Країна>

<OR><Дніпропетровська><Київська><Сумська><Львівська><Вибір Область>

<OR><Дніпропетровськ><Павлоград><Нікополь><Вибір Місто>

<OR><Багатинський><Гембарська><Жаврук><Прізвище>

<AND><Багатинський><Володимир><Іванович><Паспортні дані>

<AND><Багатинський><Володимир><Іванович><2 роки><Стаж роботи>

Висновки та напрямок подальших досліджень. На основі проведеного дослідження можна зробити наступні теоретичні та практичні висновки: проведено аналіз наукових принципів формування та управління виробничими колективами; проведено аналіз основних професійних, індивідуальних, психологічних вимог до групи розробників програмного проекту; проведено аналіз сучасних методів відбору та управління командами розробників програмного забезпе-

чення; розроблено математичну модель, структуру та алгоритми програмного комплексу автоматизації формування команди розробників програмного забезпечення; проведено аналіз моделей мотивації та розробки функціональної моделі формування та мотивації команди розробників програмного забезпечення; розроблено структурну та функціональну моделі програмного комплексу формування груп розробників; розроблено алгоритми функціональних модулів програмного комплексу; розроблено структурну модель баз даних параметрів відбору розробників; розроблено базу знань для прийняття рішень при формуванні ефективних груп, команд розробників; розроблено програмні заходи автоматизації формування ефективних груп розробників програмного забезпечення; розроблено бази даних, бази знань програмного комплексу; розроблено програмні модулі автоматизованої системи; розроблено інтерфейс користувача програмного забезпечення; сформовані вимоги до промислового програмному проекту; реалізовано систему та сформовано ефективну команду розробників програмного забезпечення. Результати проведеного дослідження можуть використовуватись у галузях важкої, легкої промисловості, фінансовій галузі.

Список літератури

1. **Somerville, Ian** Software engineering / Ian Somerville. — 9th ed. - Addison-Wesley. – 2011 – 773 p.
2. **J.D. Meier, Jason Taylor, Alex Mackman, Prashant Bansode, Kevin Jones** Team Development with Visual Studio Team Foundation Server. – Microsoft Corporation, 2007. – 495 p.
3. **Michael A. Cusumano, Stanley Smith** Beyond the Waterfall: Software Development at Microsoft – Working Paper #3844-BPS-95 - Draft: August 16, 1995. – 33 p.
4. **B. W. Boehm** A Spiral Model of Software Development and Enhancement, IEEE Computer, 21 May 1988, pp. 61–72.
5. **Новіков Ф. А., Опалева Е. А., Степанов Е. О.** Науковий посібник/ Управління проектами розробки ПЗ. – СПб:СПБГУ НТМО – 2006 – 256 с.
6. **Winslow F.** The Principles of Scientific Management.-Taylor Harper & Brothers, 1971.–77 p.
7. **Скопин И. Н.** Основы менеджмента программных проектов. Курс лекций. Учебное пособие. / И. Н. Скопин. – М.: Интуит.ру «Интернет-Университет Информационных технологий», 2004. – 336 с.
8. **Степанов С.** В поисках человечности: Абрахам Маслоу (1908 - 1970) // Шкільний психолог. - 2002. - №12 (декабрь). - С. 4-5
9. **Салливан Э.** Время – деньги. Создание команды разработчиков программного обеспечения / Пер. с англ. – М.: ИТД Русская редакция, 2002. – 368 с.
10. **Drucker, Peter F.** The Effective Executive. New York: Harper & Row, 1985 – 178 p.
11. **Архипенков С.** Управління командою розробників програмного забезпечення. – М., 2008 – 80 с.
12. **Maslow A.** New frontiers of human nature / A. Maslow. – М.: Meaning, 1999 – 496 p.
13. **Рубінштейн С.** Основы загалльної психології. – СПб: Пітер, 2002 – 720 с.

Рукопис подано до редакції 30.04.15

УДК 519.6:371.214

В.С. МОРКУН, д-р техн. наук, проф., П.В. БУРНАСОВ, ст. викладач.
Криворізький національний університет

РОЗРОБКА СИСТЕМИ УПРАВЛІННЯ РЕСУРСАМИ ВИШУ ПРИ СКЛАДАННІ РОЗКЛАДУ ЗАНЯТЬ

Розроблено структурну схему та алгоритм функціонування системи оперативного управління ресурсами вишу, особливостями якої є динамічне поєднання автоматичного режиму складання розкладу з ручним. При неможливості подальшого автоматичного формування розкладу через жорсткі обмеження, виконується їх автоматичне пом'якшення. За неможливістю автоматичного пом'якшення обмежень, або за інших потреб їх можливо пом'якшити вручну у процесі складання розкладу. В узагальнений критерій врахування інтересів студентів входять чотири часткових критеріїв оптимальності: число "вікон" у розкладі навчальних груп, баланс навантаження в розкладі навчальних груп, виконання правил угруповання занять по дням тижню, виконання правил угруповання занять по порядку занять в один день. До складу критерію врахування інтересів викладачів входять наступні частинні критерії: число "вікон" у розкладі викладачів, виконання обмеження на максимальне число зайнятих днів у тиждень для викладачів, виконання обмеження на мінімальне число занять у довільний день тижня для викладачів, особисті побажання викладачів. Узагальнений критерій оптимальності розкладу викладачів враховує ступінь оптимальності індивідуального розкладу кожного викладача. Для кількісного порівняння та ранжування часткових критеріїв оптимальності вводиться числовий еквівалент ступеня важливості кожного часткового критерію оптимальності.