

КООРДИНАЦІЯ РОЗПОДІЛЕНИХ КОМАНД РОЗРОБНИКІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У статті описано ґрунтовне дослідження проблем та особливостей розподіленої розробки програмного забезпечення систем керування проектами, визначено рівні співпраці між учасниками розподілених команд, вивчено роль організаційної пам'яті. Проведено порівняльний аналіз відомих програмних продуктів підтримки розподілених СКП. Запропоновано критерії порівняння цих продуктів.

Ключові слова: програмне забезпечення, розподілена розробка, команди розробників, порівняльний аналіз, методологія розробки програмного забезпечення, проектний менеджмент.

Вступ

За останнє десятиліття різноманітні віртуальні організації, що працюють у галузі розробки програмного забезпечення, тобто команди розробників, які працюють над одним і тим самим проектом, однак розкидані по всьому світу, з налагодженою співпрацею, координацією та комунікацією через Internet, стали одним зі стандартів налагодження процесу розробки програмного забезпечення [1; 16; 17; 19]. За цей період відбулося зростання кількості робіт, виконуваних фрілансерами, особливо в галузі високих технологій. Ці тенденції спричинили виникнення нового напрямку в керуванні проектами, а саме – розподіленого керування проектами (distributed project management). Постає завдання розробки ефективного інструментарію, який би підтримував процес інтеграції виконання та постійного навчання віртуальних команд [11]. Він мусить бути доступний у будь-якій точці світу і в будь-який час, бути платформно незалежним і, крім того, не потребувати встановлення додаткового програмного забезпечення.

У нерозподіленому проекті ключовим завданням є планування, розподіл завдань, тоді як у розподіленому ще й додається задача координації та співпраці [8]. До останнього варто приєднати важливу роль «організаційної пам'яті», тобто засобів збирання, зберігання та обміну знаннями про проект між його учасниками [27].

Розробки Web-базованих систем керування проектами (СКП), а також окремих компонент (засобів для планування, реалізацій діаграм Ганта, баг-трекерів, систем контролю версій) активно тривають щонайменше останні десять років. Розробниками створено зразки зазначених вище систем як для комерційного, так і для

вільного поширення, з різним набором можливостей, часом орієнтованих на задачі з різною специфікою. Важливим для подальшого розвитку подібних Web-застосувань є порівняння існуючих розробок і діючих, а також прототипів, що передбачає вироблення критеріїв для співставлення та аналіз вимог до них.

Системи контролю версій є обов'язковим компонентом повноцінної системи керування проектами. Іншою важливою, на нашу думку, складовою СКП має бути система обліку дефектів та покращень. Мається на увазі інтеграція баг-трекера в систему підтримки Web-вузлів для координації проектів, і додання функціоналу для переходу від конкретного дефекту до місця в програмному кодї, де він трапився для кращого розуміння учасниками суті проблеми, і найшвидшого її вирішення. Використання окремих рішень для обліку дефектів, наприклад надбудов до браузерів, вважаємо недоцільним, оскільки це потребує додаткових зусиль від користувачів системи і порушує вимоги крос-платформності та крос-браузерності.

Вагомим фактором успішності виконання проекту є керування знаннями, що передбачає доступ до релевантних знань про проект у цілому, про його окрему частину чи навіть про завдання, виконуване певним учасником проекту в певний момент часу, обмін знаннями та отримання знань із зовнішніх джерел [11]. Таке дослідження допоможе вирішити проблему відсутності неформального спілкування між учасниками віртуальної команди, яка, на думку низки дослідників, значно погіршує координацію та швидкість усунення недоліків у програмному продукті, що розробляється [17].

Вивчення проблем координації та співпраці учасників розподіленої (віртуальної) команди,

а також способи та засоби організації обміну знаннями між учасниками проекту є вкрай важливими при створенні системи керування проектами. Спробуємо проаналізувати їх.

Еволюція рішень

Дослідження проблем координації розподілених проектів набули актуальності одночасно з розвитком аутсорсингу та глобалізації бізнесу, пов'язаного із розробкою програмного забезпечення. Одним із термінів на позначення організації процесу розробки програмного забезпечення, за якого різні групи осіб, що беруть участь у процесі, знаходяться в різних точках країни чи навіть світу, є «географічно розподілена розробка програмного забезпечення» (Geographically Distributed Software Development) [17]. Одночасно з поширенням такої організації роботи розробників розпочалися дослідження проблем і завдань, що виникають у ході географічно розподіленої розробки, зокрема, це: розуміння різних форм колаборативної розробки програмного забезпечення, розробка переліку методів та підходів для вирішення питань комунікації та координації під час географічно розподіленої співпраці, створення моделей, які б поєднували різні форми колаборативної розробки програмного забезпечення, розуміння вимог інфраструктури, розуміння ролі міжкультурного бар'єра під час спілкування [17].

Гербслеб, Грінтер і Перрі в своїй праці [13], присвяченій координації географічно розподілених проектів, описують чотири підходи для розподіленої координації роботи розробників: розподіл функціональних областей, розподіл структури продукту; розподіл кроків процесу розробки та розподіл задач кастомізації. Ефективний поділ праці між учасниками визнається дослідниками як найважливіша задача при координації розподілених проектів.

У праці Кереші, Ліу та Фогеля [26] досліджено три проблеми, що постають у результаті аналізу роботи віртуальних команд, а саме комунікацію, координацію та адаптацію. Питання координації віртуальної команди тут розглянуто у світлі організації роботи у різних часових поясах. Проблема забезпечення ефективної передачі експертних знань у роботі розподіленої команди висвітлено в контексті задач комунікації.

Цікавим з практичного погляду є аналіз різних методів реалізації систем підтримки Web-вузлів для координації проектів. Так, у праці [24] проведено аналіз можливості адаптації

системи електронних зустрічей (electronic meeting system) для координації розподілених проектів. Однак на прикладі впровадження такого підходу в австралійських та новозеландських компаніях було доведено його низьку ефективність через складність адаптації до виробничого процесу.

Для ретельного аналізу еволюції програмного продукту є важливим розгляд історії релізів у поєднанні з відфільтрованою інформацією з повідомлень про виявлені дефекти. У працях Фішера, Пінцгера і Гала [9; 10] показано, як аналіз такої інформації дає можливість виявлення прихованих зв'язків між складовими програми. Підхід, запропонований цими дослідниками, передбачає, перш за все, розробку і облік складових розроблюваного програмного продукту, потім встановлення відношень між змінами в коді та звітами про проблеми в цих складових, після чого візуалізацію цих складових для ілюстрування їхніх не явних залежностей. Така візуалізація допомагає виявляти місцезнаходження недоліків в архітектурі програмної системи. Такий підхід випробувано в широковідомому відкритому проекті Mozilla.

Типи знань у керуванні проектами охарактеризовано у праці [7], у якій дано рекомендації щодо реалізації засобів збирання, збереження та використання знань у розподілених проектах. Авторами також розглянуто централізований, децентралізований та гібридизований підходи до організації роботи зі знаннями в розподілених проектах.

Інші підходи до організації та використання попереднього досвіду в розробці проектів та їх впровадження компанією Infosys Technologies Ltd. описано у Панкая Джалота [18]. Автор називає ці підходи так:

- специфікація процесу – опис вхідних та вихідних даних, обмежень, учасників, опис дій та огляд процесу;
- активи процесу – вказівки, контрольні таблиці та заготовки;
- база даних процесів – сукупність розкладів, характеристик, розмірів, обліків дефектів, програмних кодів попередніх проектів;
- звід знань – інструкції до використання певних інструментів, поради як обійти певну проблему та ін.

У праці [6] авторами розглянуто питання безпеки в засобах керування знаннями в системах для розподіленого керування проектами. Задачу візуалізації залежностей між складовими програми, засновану на аналізі повідомлень про дефекти, детально проаналізовано у [12].

Інтеграція системи контролю версій для координації Web-проектів є необхідною умовою повноцінності засобів керування проектами. Приклад інтеграції системи контролю версій (CVS) у систему для розробки Web-проектів наведено в [20]. Ми пропонуємо використовувати як складову для контролю версій подібну систему, а саме Subversion. Детальний опис цього програмного засобу можна знайти в книзі Колінза, Фіцпатріка і Пілато [17].

Низка інших підходів до реалізації систем контролю версій та їх практичне втілення описано у [3; 14; 25].

Ідеї та приклади реалізації розподіленого обліку дефектів представлено в статті Джонатана Корбета [5]. Запропонована ним концепція, згідно з якою обробку інформації про стан дефекту та відповідні виправлення в коді програми можна робити поза центральним сервером, навіть у режимі off-line, і за потреби завантажувати їх на центральний сервер чи в репозиторій коду відповідно. Наведено і перелік спроб реалізації цього підходу.

Ще однією проблемою в розподілених проектах є забезпечення наявності експертних знань при розробці тієї чи іншої складової програмного продукту певною частиною учасників проекту.

Ефективність розподіленої розробки програмного забезпечення доведено на прикладі різних методологій керування проектами, зокрема на прикладі методології Scrum, при цьому особливо відчутною була важлива роль ефективно налагодженої комунікації та обміну знаннями між учасниками проекту [29].

Чимало дослідників зазначають, що основними проблемами в розподіленому керуванні проектами є труднощі в керуванні ресурсами, брак ефективних механізмів комунікації, проблеми з безпекою передачі даних і приватності та проблеми з ефективною комунікацією між учасниками проекту [29].

Керування проектами може бути визначено як особливий спосіб організації, планування та контролю для здійснення визначених змін чи створення продуктів з передбаченими витратами, часом виконання та якістю [24]. Таким чином керування проектами включає керування фінансовими питаннями, ризиками, ресурсами, розкладами робіт, якістю та процесами.

Кожна галузь повинна вдало керуватися для успішності проекту, однак керування процесами набуває важливішої ролі в багатосторонніх проектах, оскільки включає ідентифікацію завдання, розподіл завдання, відстеження завдання

та звітування. Вдале керування проектами передбачає забезпечення учасникам проекту чіткого бачення того, хто і що робить, а також статусу певного завдання. Видимість ходу виконання проекту є критичним фактором для успіху проекту.

Проблеми обміну інформацією між учасниками віртуальних команд

Явне представлення проектною інформації є суттєвим для ефективної та дієвої комунікації, особливо в розподілених ситуаціях. Документування проектною інформації зі значним рівнем деталізації зробить можливим всебічне розуміння проекту учасниками команди. Наприклад, у нерозподілених проектах учасники команди можуть працювати окремо кожен над своїм завданням, формально писати звіти і усно доповідати про свої результати. В розподілених проектах учасники повинні документувати значно більше деталей, включаючи процеси (кроки виконання завдань, використані засоби), обґрунтування (причини використання того чи іншого інструменту чи методології) та контекст (учасники виконання завдання, визначені дати початку та закінчення, фактична дата закінчення).

Ефективна комунікація також потребує чітких специфікацій та одноголосного погодження такої важливої проектною інформації, як ключові поняття, ідеї, процеси проекту та відповідальність учасників. Все це має бути документовано та збережено для отримання довідки учасниками проекту. Для забезпечення такого рівня деталізації засоби розподіленого керування проектами мусять не тільки підтримувати, а й спонукати учасників документувати контекст та процес виконання завдань. Крім явного представлення проектною інформації, інструменти розподіленого керування проектами потребують автоматичного обліку статусів виконання завдань, повинні давати можливість учасникам обговорювати та коментувати роботу інших. Явне представлення не є тотожним ефективній комунікації, однак є важливим кроком до її відтворення.

Явна документація призводить до ефективнішої комунікації, зокрема тому, що інформація є систематизовано збереженою. Структурована природа деталізованої інформації пришвидшує процес пошуку специфічної інформації або тенденцій чи зразків у завданнях чи навіть цілих проектах.

Інша проблема в розподілених командах – це недостатня кількість неформального спілкування. Неформальне спілкування – розмови за

кавою, за обідом, під час зустрічей у коридорі, мають неочікувано велике значення для координації проектів.

Неформальне спілкування відіграє значну кількість критичних ролей у розробці програмного забезпечення, особливо у випадках, коли є потреба у різного гатунку синхронізації [17]. Ця взаємодія потребує постійної уваги, оскільки зміни в одній частині коду можуть спричинити зміни у поведінці іншого модуля – відношення залежності. Ці залежності часто виникають між фрагментами коду в одній і тій самій підсистемі, а колеги, що працюють поруч, часто знають про ці залежності. Інколи шляхом виявлення цих залежностей є випадкова розмова між людьми, що зустрілися «в коридорі» й обмінялися думками.

Неформальне спілкування допомагає також учасникам команди ознайомитися з робочим стилем команди, поглядами на спосіб виконання певних процесів, ієрархію в організації, культурні відмінності будь-якого роду, а також допомагає вирішувати інші проблеми. В цілому, відсутність неформального спілкування спричиняє більшу інтенсивність виникнення проблем, оскільки таким чином розробники позбавлені швидкого способу обміну інформацією та досвідом для їх подолання.

Варто зазначити, що неформальне спілкування є лише одним зі способів для учасників команди координувати та узгоджувати їхню роботу. Іншим підходом є побудова організації роботи та розподіл завдань таким чином, який дозволив би скоротити об'єм потрібного неформального спілкування. Хоча це не знімає проблеми в неформальному спілкуванні, метою є її зменшення на керованому рівні. Підходом, що дозволяє реалізувати такий вид системи, є колаборативний у керуванні проектами (collaborative project management). Колаборативне керування проектами зосереджується на явному представленні інформації про проект, а також своєчасному обміні інформацією. Головною метою є забезпечення правильною інформацією потрібних людей у потрібний час [4; 15; 27].

Під час розгляду системи керування проектами як інструменту звітів про проекти, йдеться радше про вихідні дані з проектів, ніж про аналіз процесів, які продукують ці вихідні дані. Наприклад, можна створити ПЕРТ діаграму чи діаграму Ганта для розподілення робіт. Якщо розробники зберуться разом і докладуть зусиль для розробки діаграми (розкладу робіт), так вони пройдуть низку етапів: поділ проекту на певні керовані завдання, оцінку, час виконання

кожного завдання і вибору альтернатив для зменшення ризиків. Значна частина додаткової інформації, пов'язаної з проектами, залишається формально не зібраною і випадає з організаційної пам'яті.

З другого боку, при розгляді керування проектами як інструменту аналізу проекту, продуктом стануть діаграма, інформація про завдання, обґрунтування рішень та інші пов'язані з цим артефакти. Засоби колаборативного керування проектами повинні полегшувати учасникам проекту генерацію та організацію ідей, вибір альтернатив. До того ж, результати мусять зберігатися в постійному сховищі для подальшого аналізу проекту, коли при змінах у розкладі робіт учасники проекту дістануть пов'язану з ним інформацію для передбачення можливих ефектів чи послідовностей змін, що, можливо, виникнуть унаслідок змін, та уникнути негативних послідовностей змін. Колаборативний підхід у керуванні проектами допоможе забезпечити потужний інформаційний потік між учасниками команди розробників і таким чином забезпечити ефективне та дієве спілкування. Ефективне спілкування і вчасний обмін інформацією є суттєвою основою успіху.

Низка авторів пропонують п'ять рівнів в ієрархічному підході до організації співпраці: комунікативний, колективний, кооперативний, координаційний та узгоджувальний [21; 22; 23]. Програмне забезпечення для керування проектами також можна класифікувати згідно з цими чотирма рівнями, причому програмні продукти найвищого рівня матимуть усі властивості нижчих рівнів.

Комунікативний рівень програмного забезпечення є веб-базованим і забезпечує тільки центральне сховище для файлів. Кожен учасник проекту має до нього доступ у будь-який час і з будь-якого місця. Люди обмінюються даними через це сховище, кожен працює окремо над своїми завданнями. Однак це не дає підтримки для взаємодії між учасниками команди, а взаємодія є дуже важливою для успішної співпраці.

Колективний рівень програмного забезпечення забезпечує сховище для інформації, завдань, послідовностей виконання проекту та розподілу ресурсів. Люди працюють для досягнення спільних цілей і можуть перевіряти оновлену роботу інших людей. Однак інтеграція роботи учасників є мінімальною.

Координаційний рівень надає можливість створювати, обмінюватись, переглядати та редагувати документи, перевіряти календарі та

координувати розклади виконання робіт, а також переглядати завдання, що дає можливість людям бути співавторами документів у послідовний спосіб. Цей рівень програмного забезпечення для керування проектами містить можливості колективного рівня у поєднанні з можливостями групового календаря та системи спільної роботи над документами. Люди працюють над спільною метою і взаємодіють один з одним.

Узгоджувальний рівень дозволяє працювати над документами паралельним чином, а також забезпечує групове синхронне та асинхронне прийняття рішень.

Низькі рівні співпраці зосереджуються головним чином на обміні інформацією. Обмін інформацією є важливим, проте недостатнім. Учасникам проекту потрібно виконувати свої завдання залежно від роботи інших людей. Взаємодія між учасниками проекту набуває суттєвого значення. Вона включає узгодження цілей, розподіл завдань, укладання розкладів виконання робіт, розподіл ресурсів, спільну роботу над одним і тим самим документом чи завданням. Без ефективної підтримки взаємодії учасники проекту матимуть серйозні проблеми на шляху до досягнення спільних цілей.

Важливим у реалізації колаборативного підходу в керуванні проектами є поняття організаційної пам'яті, оскільки передбачає створення спільного репозиторію для зберігання організаційного досвіду команди, що є необхідним у віртуальних командах.

Організаційна пам'ять визначається як «засоби, за допомогою яких досвід із минулих проектів переноситься на поточну діяльність, таким чином маючи за результат покращення чи погіршення організаційної відповідальності». Процесами організаційної пам'яті є збирання інформації з усіх можливих джерел; підтримання інформації в належному вигляді; зберігання інформації для ефективної її обробки; засоби пошуку для підтримки прийняття рішень та вирішення проблем [28]. Сховища знань набувають особливого значення, оскільки дозволяють учасникам команди обмінюватися досвідом, потрібним для виконання проектів та бути в курсі існуючих напрацювань та розробок. Організаційна пам'ять тісно пов'язана з керуванням знаннями завдяки створенню механізмів зберігання знань у стійкому форматі. Знання можна класифікувати як явні та неявні. Існують способи перетворення знань з однієї категорії в іншу: комбінація – поєднання явних знань з явними для отримання явних знань; соціалізація – перетворення неявних знань на явні; інтерналізація – перетворення

явних знань на неявні; екстерналізація – перетворення неявних знань на явні.

Засоби керування знаннями для накопичення неявних знань, перетворення їх на явні та зберігання в доступному вигляді для подальшого використання є важливим процесом. СКП потребують компонент для керування знаннями, накопичення та зберігання досвіду для застосування в поточних та майбутніх проектах.

Порівняння програмного забезпечення СКП

Для аналізу та порівняння ми обрали відомі програмні продукти: Trac, Bugzilla, GForge, SharpForge, Codendi, Collabtive, DotProject, Redmine та jotBug.

Bugzilla. Система є програмним продуктом з відкритим кодом, розповсюджується згідно з ліцензією Mozilla Public License. Цей програмний продукт написано мовою Perl і для роботи потребує певний Web-сервер з підтримкою CGI та систему керування базами даних MySQL, PostgreSQL або Oracle.

У цілому цей програмний продукт є системою обліку дефектів – bugtracker. Однак досить потужний функціонал Bugzilla дозволяє розглянути її як систему корисну при керуванні проектами. Важливою рисою цього програмного продукту є потужний Google-подібний механізм пошуку інформації, повнотекстовий пошук інформації про баги, занесеної до системи, та пошук за різними параметрами, зокрема за часовими. До цього слід додати інструмент для перегляду історій зміни коду, зокрема з полагодження певного дефекту. Інформацію, накопичену в систему, можна переглянути у вигляді звітів про баги в різних форматах багів (Atom, iCal, CSV) та виглядах, можливість генерації графіків та діаграм.

Координація учасників розподіленої команди здійснюється через низку інструментів: повідомлення через e-mail про зміни та надсилання звітів щодо роботи системи у вказаний термін; можливість «спостерігати» за іншими користувачами; можливість надсилати прохання користувачам тощо. Існує можливість створення різних користувацьких груп, з можливістю створення змісту з доступом для певної групи.

Важливим інструментом для керування роботою команди є трекінг часу, тобто: встановлення певного часу для залагодження дефекту тощо.

Trac. Розроблений компанією Edgewall Software є програмним продуктом з відкритим кодом, розповсюджуваним за ліцензією Modified BSD. Trac написано мовою Python.

Для роботи потребує певний Web-сервер з підтримкою CGI та систему керування базами даних MySQL, PostgreSQL або SQLite та засоби підтримки мови Python.

Ця система є однією з найпотужніших цього виду, що підтверджує перелік наявних у ній модулів та інструментів, зокрема засобів для планування проєктів (Roadmaps, Milestones). Система Trac включає засіб розподілу завдань (також і bugtracker). Кожне завдання описується за автором завдання (тікету), природа завдання, компонент програмного продукту, що розробляється, якого стосується завдання, пріоритет, ключові слова, опис проблематики тощо.

Важливою рисою Trac є вбудований веб-інтерфейс до низки систем контролю версій (SVN, Git, Mercurial, Bazaar та Darcs). Вбудована система Wiki є зручним засобом групового створення та підтримання документації, обміну досвідом, з можливістю контролю версій документів. Система забезпечує одночасне ведення кількох проєктів. У системі Trac присутній також механізм генерації звітів з можливостями різнобічної кастомізації. Свідченням досконалості цього програмного продукту є його використання в підрозділах NASA.

Redmine. Це програмний продукт з відкритим кодом, розповсюджуваний за ліцензією GNU GPL, написаний мовою Ruby, для роботи потребує систему керування базами даних MySQL, PostgreSQL або SQLite та засоби підтримки мови Ruby.

Головною перевагою Redmine є можливість використання багатьма командами одразу, зокрема шляхом самостійної реєстрації користувачів. Гнучка система розподілу повноважень та надання доступів дає змогу адміністраторові будь-яким чином формувати команди розробників для виконання певних задач. Кожен з інтегрованих модулів можна активувати чи деактивувати залежно від того, чи потрібне його застосування в певному проєкті.

Redmine містить гнучку систему відстеження (трекінгу) задач, що дозволяє визначати їхні типи і статуси та встановлювати потік робіт для цих задач. У системі реалізовано роботу з діаграмами Ганта та календарем, зокрема є можливість їх автоматичної побудови за датами початку та завершення робіт.

Також до Redmine інтегровано ряд систем контролю версій (SVN, Git, Mercurial, Bazaar та Darcs). Вбудована система Wiki для групового створення та підтримання документації, обміну досвідом, можливістю контролю версій доку-

ментів, а також вбудовані форуми для обговорення кожного завдання учасниками проєкту. Останнє також дозволяє зменшити недоліки відсутності неформального спілкування між учасниками віртуальних команд.

Важливо зазначити наявність пакетів локалізації Redmin на велику кількість мов, зокрема українську.

Collabtive є програмним продуктом з відкритим кодом, який розповсюджується за ліцензією GNU, написаний мовою PHP5 з використанням JavaScript\AJAX. Для роботи потребує систему керування базами даних MySQL вище версії 4.0 та Web-сервер з підтримкою PHP5.

Серед засобів планування проєктів у системі реалізовано визначення підгруп завдань (Milestones), створення та керування списків завдань, а також перегляд завдань через календар. До Collabtive включено механізми експорту даних (Zip, XML, RSS, iCal) та імпорту даних. Також вбудовано файловий менеджер.

DotProject. Система є програмним продуктом з відкритим кодом, розповсюджуваним за ліцензією GNU, написаний мовою PHP. Для роботи потребує систему керування базами даних MySQL вище версії 4.0 та Web-сервер з підтримкою PHP.

У DotProject реалізовано механізм відстеження завдань та створення розкладів робіт, зокрема з використанням діаграм Ганта. Присутній механізм керування ресурсами. Обмін організаційним досвідом здійснюється через систему форумів. Є також система збереження історії дій учасників команди.

Codendi розроблено компанією XEROX і є програмним продуктом з відкритим кодом, розповсюджуваним за ліцензією GNU GPL, написаний мовою PHP. Для роботи потребує систему керування базами даних MySQL вище версії 4.0 та Web-сервер з підтримкою PHP.

Система містить потужний універсальний засіб відстеження завдань та дефектів, який також включає інструменти для генерації звітів, зокрема у формі діаграм різного виду, засоби побудови діаграм Ганта. В системі відстеження (трекінгу) присутні інструменти для підтримки різних методологій розробки програмного забезпечення (SCRUM, CMMi).

До складу Codendi входить також тестовий фреймворк Salome, що забезпечує організацію процесів тестування в ієрархічну деревовидну структуру, визначає тести в ручному режимі

(послідовності дій) та автоматичному (програми чи скрипти), включає також керування помилками (error management) та керування вимогами (requirements management). Присутній механізм генерації звітів у HTML форматі.

У Codendi інтегровано систему контролю версій (SVN, CVS) з підтримкою навігації по репозиторію, пошуку та графічного відображення змін.

Вбудована система керування документами дозволяє кілька можливих режимів перегляду, підтримку контролю версій документів, додання до документів мета-даних, потужний механізм повідомлення про події та інтеграцію документації і Wiki.

Взаємодія та спілкування між учасниками команди підтримуються завдяки системі форумів та миттєвих повідомлень.

eGroupWare. Система розроблена компанією Open Source Project і є програмним продуктом з відкритим кодом, розповсюджуваним за ліцензією GNU GPL. Цей програмний продукт написаний мовою PHP. Для роботи потребує систему керування базами даних MySQL вище версії 4.0 та Web-сервер з підтримкою PHP.

eGroupWare призначена для використання в підприємствах малого чи середнього рівня. Головне призначення керування контактами, зустрічами, проектами та переліками завдань до виконання. Локалізація доступна на понад 25 мов.

eGroupWare включає в себе календар з функціями складання розкладів для груп, ресурсів та контактів. До календаря також інтегровано інструмент керування ресурсами та окремі елементи бухгалтерії.

Присутній модуль, що реалізує функції відстеження дефектів (bug tracker). Обмін організаційним досвідом у цьому програмному продукті можливий через вбудовану систему Wiki.

GForge. Ця система є програмним продуктом з відкритим кодом, розроблена для використання

в проекті SourceForge – відомому Internet репозиторії програмного коду. Він написаний мовою PHP, з чого випливають відповідні системні вимоги до Web-сервера, обраного для розміщення.

У GForge реалізовано засоби для відстеження завдань, систему звітів з експортом у різні формати та засоби для планування робіт.

Система включає інтегровані системи контролю версій, а саме SVN, SVC та Perforce. Інтеграція дозволяє, зокрема, відображати зміни в коді на трекер, повідомляти через електронну пошту про здійснення операції фіксації тощо. Набір додатків (plugins) дозволяє налагоджувати роботу з іншими програмними комплексами, призначеними для розробки програмного забезпечення, а саме з Visual Studio, MS Project та Eclips.

До системи вбудовано засоби для документування з підтримкою контролю версій документів, система Wiki та механізм з творення форумів.

Висновки

У статті здійснено дослідження проблем та особливостей розподіленої розробки програмного забезпечення СКП, визначено рівні співпраці між учасниками розподілених команд, обґрунтовано важливість неформального спілкування між учасниками віртуальної команди та наведено шляхи подолання її відсутності, досліджено ряд аспектів контролю за процесом розробки продукту в розподіленій команді та проблеми інтеграції продукту, вивчено роль організаційної пам'яті, тобто обмін та збереження досвіду в віртуальних командах.

Проведено порівняльний аналіз відомих програмних продуктів підтримки, розподілених СКП. Запропоновано критерії порівняння цих продуктів, виходячи, зокрема, з міркувань поліпшення механізму координації, обміну досвідом та налагодження ефективної співпраці між учасниками розподіленої команди.

Список літератури

1. Глибовець М. М. Формальна модель координаційно-орієнтованої мережі для колаборативної системи навчання / М. М. Глибовець, Д. К. Гломозда // Проблеми програмування. – 2006. – № 2–3. Спец. вип. – С. 402–412.
2. Коллінз-Сассман Б. Пилато Управление версиями в Subversion [Електронний ресурс] / Бен Коллінз-Сассман, Б. У. Фитцпатрик, К. Майкл. – Режим доступу: <http://svnbook.red-bean.com/nightly/ru/svn-book.html>. – Назва з екрана.
3. Allen Larry W. Version control system for geographically distributed software development [Electronic resource] / W. Allen Larry et al. – Mode of access: <http://www.google.de/patents/US5675802>. – Title from the screen.
4. A Collaborative Project Management Architecture [Electronic resource] / Fang Chen, Nicholas C. Romano, Jr, Jay F. Nunamaker, Robert O. Briggs // 36th Hawaii International Conference on System Sciences – 2003. – Mode of access: <http://www.hicss.hawaii.edu/HICSS36/HICSSpapers/CLDCP01.pdf>. – Title from the screen.
5. Corbet Jonathan. Distributed bug tracking [Electronic resource] / Corbet Jonathan. – 2008. – Mode of access: <http://lwn.net/Articles/281849>. – Title from the screen.
6. Daniela Damm and Martin Schindler. Security issues of a knowledge medium for distributed project work / Daniela Damm and Martin Schindler // International Journal of Project Management. – 2002. – Vol. 20. – P. 37–47.

7. Desouza Kevin C. Knowledge Management in Non-Collocated Environments: A Look at Centralized Vs. Distributed Design Approaches [Electronic resource] / Kevin C. Desouza, Anuradha Jayaraman, J. Roberto Evaristo. – Mode of access: <http://www.hicss.hawaii.edu/HICSS36/HICSSpapers/CLDCP02.pdf>. – Title from the screen.
8. Evaristo R. A typology of project management: emergence and evolution of new forms Teams / R. Evaristo, P. C. van Fenema // International Journal of Project Management. – 1999. – Vol. 17, № 5. – P. 275–281.
9. Fischer Michael Populating a Release History Database from version control and bug tracking systems / Michael Fischer, Martin Pinzger, Harald Gall // IEEE Computer Society. – USA : Washington, 2003. – 23 p.
10. Fischer Michael. Analyzing and Relating Bug Report Data for Feature Tracking [Electronic resource] / Michael Fischer, Martin Pinzger, Harald Gall // Proceedings of the 10th Working Conference on Reverse Engineering (WCRE'03). – Mode of access: http://seal.ifi.uzh.ch/fileadmin/User_Filemount/Publications/fischer-wcre03.pdf. – Title from the screen.
11. Frank Maurer. Integrating Process Support and Knowledge Management for Virtual Software Development Teams [Electronic resource] / Frank Maurer, Harald Holz. – Mode of access: <http://sern.cpsc.ualgary.ca/~milos/papers/2002/MaurerHolz2002.pdf>. – Title from the screen.
12. Gall Fischer and Harald MDS-Views: Visualizing Problem Report Data of Large Scale Software using Multidimensional Scaling [Electronic resource] / Gall Fischer and Harald. – Mode of access: <http://prog.vub.ac.be/FFSE/Workshops/ELISA-submissions/19-FischerGall-full.pdf>. – Title from the screen.
13. Grinter E. Rebecca The Geography of Coordination: Dealing with Distance in R&D Work [Electronic resource] / Rebecca E. Grinter, James D. Herbsleb, Dewayne E. Pery. – Mode of access: [http://conway.isri.cmu.edu/~jdh/collaboratory/research_papers/Group_99\(final\).pdf](http://conway.isri.cmu.edu/~jdh/collaboratory/research_papers/Group_99(final).pdf). – Title from the screen.
14. Hamilton Oliveira. Odyssey-VCS: a Flexible Version Control System for UML Model Elements [Electronic resource] / Hamilton Oliveira, Leonardo Murta, Claudia Werner. – Mode of access: http://www2.ufpa.br/cdesouza/teaching/es/odysseyvcs_SCM-12.pdf. – Title from the screen.
15. Helbrough B. Computer assisted collaboration – the fourth dimension of project management? / Helbrough B. // International journal of project management : the journal of the International Project Management Association. – 1995. – Vol. 13, № 5. – P. 329–333.
16. Herbsleb J. D. Global software development [Electronic resource] / Herbsleb J. D., Moitra, D. // IEEE Software. – 2001. – Mode of access: <http://herbsleb.org/web-pubs/pdfs/herbsleb-global-2001.pdf>. – Title from the screen.
17. Herbsleb James D. Geographically Distributed Software Development. Bell Labs, Lucent Technologies [Electronic resource] / James D. Herbsleb Rebecca E. Grinter, and Lawrence Votta Jr. – Mode of access: http://kluedo.ub.uni-kl.de/volltexte/2000/217/pdf/no_series_210.pdf. – Title from the screen.
18. Jalote Pankaj Knowledge Infrastructure for Project Management [Electronic resource] / Jalote Pankaj. – Mode of access: <http://www.cse.iitk.ac.in/users/jalote/papers/KnowledgeMgmt.pdf>. – Title from the screen.
19. Lanubile Filippo. Global software development: technical, organizational, and social challenges [Electronic resource] / Filippo Lanubile, Daniela Damian, Heather L. // ACM SIGSOFT Software Engineering. – 2003. – Vol. 28, № 6. – Mode of access: <http://www.di.uniba.it/~lanubile/papers/sigsoft2003.pdf>. – Title from the screen.
20. Marinalva Soares Dias. Versionweb: a Tool for helping Web Pages Version Control Internet and Multimedia Systems and Applications [Electronic resource] / Marinalva Dias Soares, Renata P. Mattos Fortes, Dilvan de Abreu Moreira; 2000; Las Vegas, Nevada, USA. – Mode of access: <http://java.icmc.usp.br/dilvan/papers/imsa2000.pdf>. – Title from the screen.
21. Maurer F. Working Group report on Computer Support in Project Coordination [Electronic resource] / Maurer F. // Proceedings of the the Project Coordination Workshop of the IEEE Fifth Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE). USA. : Stanford University. – 1996. – Mode of access: <http://favorstar.com/book/download/lit/Working+Group+Report+on+Computer+Support+in+Project+Coordination>. – Title from the screen.
22. Nunamaker J. F. A Framework for Collaboration and Knowledge Management [Electronic resource] / J. F. Nunamaker, Jr., Romano, N. C., Jr. and Briggs R. O. // Proceedings of the Thirty-third Hawaii International Conference on System Sciences, Maui, HI, USA: IEEE Computer Society Press. – 2000. – Mode of access: http://www.hicss.hawaii.edu/HICSS_34/PDFs/CLUSR07.pdf. – Title from the screen.
23. Nunamaker J. F. Information Technology and the Future Enterprise: New Models for Managers / J. F. Nunamaker, Jr. Briggs, O. Robert, de Vreede, Gert-Jan. – New Jersey : Prentice Hall. – 312 p.
24. Pervan G. Adoption and Use of Electronic Meeting Systems in Large Australian and New Zealand Organizations [Electronic resource] / Pervan G. Lewis L. F. and D. S. Bajwa. // Group Decision and Negotiation. – 2004. – P. 403–414. – Mode of access: <http://www.springerlink.com/index/U78N322542135270.pdf>. – Title from the screen.
25. Plaice, John and Wadge. A New Approach to Version Control [Electronic resource] / Plaice, John and Wadge, William W. – Mode of access: <http://eprints.kfupm.edu.sa/19464/1/19464.pdf>. – Title from the screen.
26. Qureshi Sajda. The Effects of Electronic Collaboration in Distributed Project Management [Electronic resource] / Sajda Qureshi, Min Liu and Doug Vogel // Group Decision and Negotiation. – 2006. – Vol. 15. – № 1. – Mode of access: <http://142.244.104.252/mact/aod/files/-1/25/The+Effects+of+Electronic+Collaboration+in+Distributed+Project+Management.pdf>. – Title from the screen.
27. Romano C. Nicholas Jr. Collaborative Project Management Software [Electronic resource] / Nicholas C. Romano, Jr., Fang Chen, Jay F., Nunamaker Jr. // Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03). – Mode of access: <http://csdl2.computer.org/comp/proceedings/hicss/2002/1435/01/14350020b.pdf>. – Title from the screen.
28. Stein E. W. Actualizing organizational memory with information systems [Electronic resource] / E. W. and V. Zwass // Information Systems Research: A Journal of the Institute of Management Sciences. – 1995. – Vol. 6. – № 2. – P. 85–117. – Mode of access: http://eprints.utm.my/3113/1/Organizational_Memory_Information_System_Case_Study_in_Faculty_of_Computer_Science_&_Information_System_UTM_2006.pdf. – Title from the screen.
29. Sutherland Jeff. Distributed Scrum: Agile Project Management with Outsourced Development Teams [Electronic resource] / Sutherland Jeff, Ph. D. Anton Viktorov. – Mode of access: <http://jeffsutherland.com/SutherlandDistributedScrumHICCS2007.pdf>. – Title from the screen.

A. Afonin, A. Glybovets, O. Bobko

COORDINATING DISTRIBUTED SOFTWARE DEVELOPMENT

This article describes a deep research of problems and characteristics of distributed software development project management systems, the levels of cooperation between members of distributed teams investigated the role of organizational memory. It was made a comparative analysis of known software products of distributed UPC support. It proposes criteria to compare these products.

Keywords: software, distributed development, team development, comparative analysis, methodology of software development, project management.

Матеріал надійшов 12.05.2014

УДК 004.891.2

Глибовець М. М., Конюшенко О. В.

ВИКОРИСТАННЯ ДІЯЛЬНОСТІ ВІРТУАЛЬНИХ СПІЛЬНОТ ДЛЯ ПОКРАЩЕННЯ ІНФОРМАЦІЙНОГО ПОШУКУ

У статті обґрунтовано задачу розробки системи персоналізованого пошуку, яка здійснювала б пошук документів тематичного порталу з урахуванням результатів кооперативної діяльності віртуального співтовариства порталу за оцінкою документів.

Ключові слова: персоналізований пошук, віртуальні спільноти, генетичні алгоритми, тематичні портали.

Вступ

У зв'язку зі збільшенням об'єму даних, розміщених в Інтернет і корпоративних базах даних, актуальною стає проблема підвищення якості пошукових і навігаційних систем [2]. Наразі більшість пошукових систем можуть дізнаватися про інформаційну потребу користувача тільки за допомогою ключових слів, заданих у запиті. Причому, система не має в своєму розпорядженні відомостей про контекст, в якому користувач вживає ці слова. Та й самому користувачеві часом важко підібрати ключові слова, які точно описують тематику, що його цікавить. Пошукова система надає користувачеві результати, відсортовані за релевантністю до запиту. При цьому враховується текст документів або структура посилань між ними, але не враховується ряд інших факторів, які могли б покращити якість пошуку [1; 5], зокрема:

- коло інтересів і індивідуальні особливості користувача, який веде пошук;
- результати кооперативної діяльності віртуального співтовариства за оцінкою якості ресурсів.

Перший з цих факторів можна охарактеризувати як проблему впровадження персоналізованого пошуку, другий – поширення впливу результатів діяльності одного користувача (зокрема, оцінювання якості ресурсів) на результати пошуку інших, що в перспективі може призвести до підвищення ефективності пошуку.

Актуальність досліджень у цій галузі підтверджується як діяльністю багатьох наукових спільнот у цьому напрямі, так і впровадженням систем, які реалізують ці ідеї, великими корпораціями, що представлені на ринку пошуку інформації. Зокрема, не так давно компанія Google представила свій сервіс персоналізованого пошуку «Вікіпошук»