ТЕМА 18. КЕРУВАННЯ КОНФІГУРАЦІЄЮ ПРОГРАМНИХ ЗАСОБІВ

1 Основні визначення

Під конфігурацією розуміється конкретна версія програмної системи, що містить у собі функції, об‘єднані між собою процедурами, зв‘язками та параметрами, що задають режими функціонування системи.

Версія або конфігурація системи складається з:

1) базису конфігурації, формально створеної основи системи з окремо створених компонентів і документації, що дозволяють проводити подальший розвиток системи;

2) елементів конфігурації, виділених для керування або оброблення функції системи;

3) програмних компонентів, що виконують завдання сформованої версії системи.

Базис – конфігураційні певні технічні рішення, перелік головних елементів конфігурації, значень параметрів і спеціалізовані процедурні зв‘язки та розгортання версії системи з компонентів заданої послідовності.

Чим більше в системі компонентів, тим більша ймовірність того, що деякі з них будуть містити помилки.

Керування конфігурацією (КК) – це процес забезпечення ідентифікації елементів конфігурації системи при її створенні для проведення систематичного контролю, обліку й аудиту внесених змін, а також для перевірки цілісності та працездатності системи.

Відповідно до стандарту IEEE 828-2005, керування конфігурацією містить:

1) ідентифікацію конфігурації;

2) контроль конфігурації;

3) облік статусу конфігурації.

Керування конфігурацією для великих систем створюється за допомогою методів і засобів, що забезпечують:

– ідентифікацію елементів системи;

– контроль внесених змін;

– можливість визначення фактичного стану системи при роз- робленні й експлуатації в будь-який момент часу.

Для досягнення цілей керування конфігурацією повинне проводитися планування й виконання проекту з урахуванням виникаючих обмежень операційних систем і обладнання замовників.

Процес змін містить у собі: визначення типів змін; організацію їх проведення; формування концепції допуску відхилень і відмов стосовно вимог проекту системи.

Результат внесених змін – це нова версія системи, документація з проведення випробувань і документація користувача на систему.

При внесенні змін проводиться контроль поточної версії системи з використанням репозитарію (Rational Clear Case Source і Source Safe of Microsoft – інструменти перевірки змін) і перевірка вихідного коду отриманої версії.

Планування конфігурації залежить від типу проекту, організаційних заходів, обмежень і загальних рекомендацій до керівництва конфігурацією.

До планування керування конфігурацією відносять:

– ідентифікацію;

– визначення статусу й аудиту конфігурації;

– керування змінами конфігурації.

До засобів забезпечення планування належать:

– система керування кодами, переведення й об‘єднання компонентів у конфігураційні системи;

– базові бібліотеки та ресурси;

– спеціальні групи контролю системи та її конфігурацій;

– системи керування для ведення проекту та зберігання змін.

Ідентифікація конфігурації – іменування всіх елементів системи на основі системи класифікації й кодування елементів, а також подання та ведення версій конфігурації з використанням належних до неї елементів. Вона виконується за допомогою методів структуризації, класифікації й іменування елементів системи. При цьому проводиться:

1) визначення стратегії ідентифікації для одержання врахованої версії системи;

2) іменування складених елементів, частин і всієї конфігурації системи;

3) встановлення співвідношення між кількістю завдань, що виконуються, та кількістю пунктів конфігурації;

4) ведення версії системи та її документування;

5) набір елементів у базисі конфігурації і його формальне позначення.

Основи ідентифікації становить конфігураційний базис, набір формально розглянутої й затвердженої документації як основи для подальшого розвитку або розроблення системи.

Версія системи містить елементи конфігурації та систему для одержувача.

Керування версіями полягає:

1) в інтеграції або композиції коректної остаточної версії системи з елементів конфігурації, реалізованих на етапах життєвого циклу, а також з урахуванням апаратних засобів і інструментів побудови системи;

2) у виборі інструментів побудови версії оцінки можливостей середовища та засобів автоматизації процесу побудови окремих версій з коректними конфігураціями ПО й даних;

3) керування варіантами версій, що включають сукупність готових ідентифікованих елементів системи та задовольняють задані вимоги замовника до варіанта.

Після одержання нової версії системи замовникові передаються версія, конфігурація, документація й інструменти керування версіями для самостійного внесення змін при супроводі системи.

Контроль конфігурації – це перевірка й керування змінами системи при формуванні версії й експлуатації.

Предметом проведення контролю конфігурації є:

1) зміни в базисі конфігурації й пов‘язане з ними коректування конфігурації;

2) дефекти й відхилення конфігурації продукту щодо затвердженого базису.

Це здійснюється за допомогою формальних процедур ініціалізації, аналізу, прийняття й контролю, виконання управлінських рішень щодо зміни виявлених дефектів і відхилень у конфігурації.

Після того, як зацікавлені учасники проекту досягли взаєморозуміння, відповідні проектні документи вважаються затвердженими і не можуть довільно модифікуватися. Отже, будь-яка потреба в зміні повинна пройти процедури:

1) реєстрації додатка на зміни;

2) аналізу впливу запропонованої зміни на наявний доробок, обсяг, трудомісткість, графік і вартість робіт із проекту;

3) ухвалення рішення про зміну: задовольнити, прийняти або відмовити;

4) реалізація затвердженої зміни й верифікація.

Другою складовою контролю конфігурації є керування невідповідностями між конфігурацією й елементами продукту та базисами конфігурації.

З погляду керування всі невідповідності прийнято ділити на дефекти й відхилення.

До дефектів відносять невідповідності, що мають відношення до цільового використання продукту за його призначенням. Все інше відносять до відхилень.

Якщо дефекти в продуктах носять негативний характер, то вони підлягають усуненню.

Облік статусу полягає в реєстрації та наданні інформації для ефективного керування конфігурацією. Предмет такого обліку – інформація про поточний статус ідентифікованих об‘єктів, запропонованих змінах, а також про виявлені дефекти та відхилення.

Звіт про статус конфігурації є ключовим для прийняття управлінських рішень щодо проекту.

Дані обліку статусу конфігурації – вихідні дані для формування кількісних оцінок або метрик продуктивності.

Аудит конфігурації містить:

1) функціональний аудит конфігурації, що проводиться для підтвердження відповідності фактичних характеристик конфігурації продукту вимогам замовника;

2) фізичний аудит, що є підтвердженням взаємної відповідності документації й фактичної конфігурації продукту.

2. Функціонування систем керування конфігурацією

Системи управління конфігураціями можуть стати потужним засобом поліпшення взаємодії, продуктивності та якості шляхом автоматизації процесів та інтеграції інструментів, що використовуються в даний час більшістю груп інженерів. Управління конфігураціями можна назвати мостом, що зв'язує групи розробників в компанії. Застосування системи управління конфігураціями дозволяє організувати свої зусилля по розробці, в основі яких лежать чіткі і повторювані процеси; при цьому, допомагаючи своїй групі в розстановці пріоритетів та управлінні життєвим циклом розробки, ви більшою мірою здатні забезпечити відповідність вимогам своїх клієнтів.

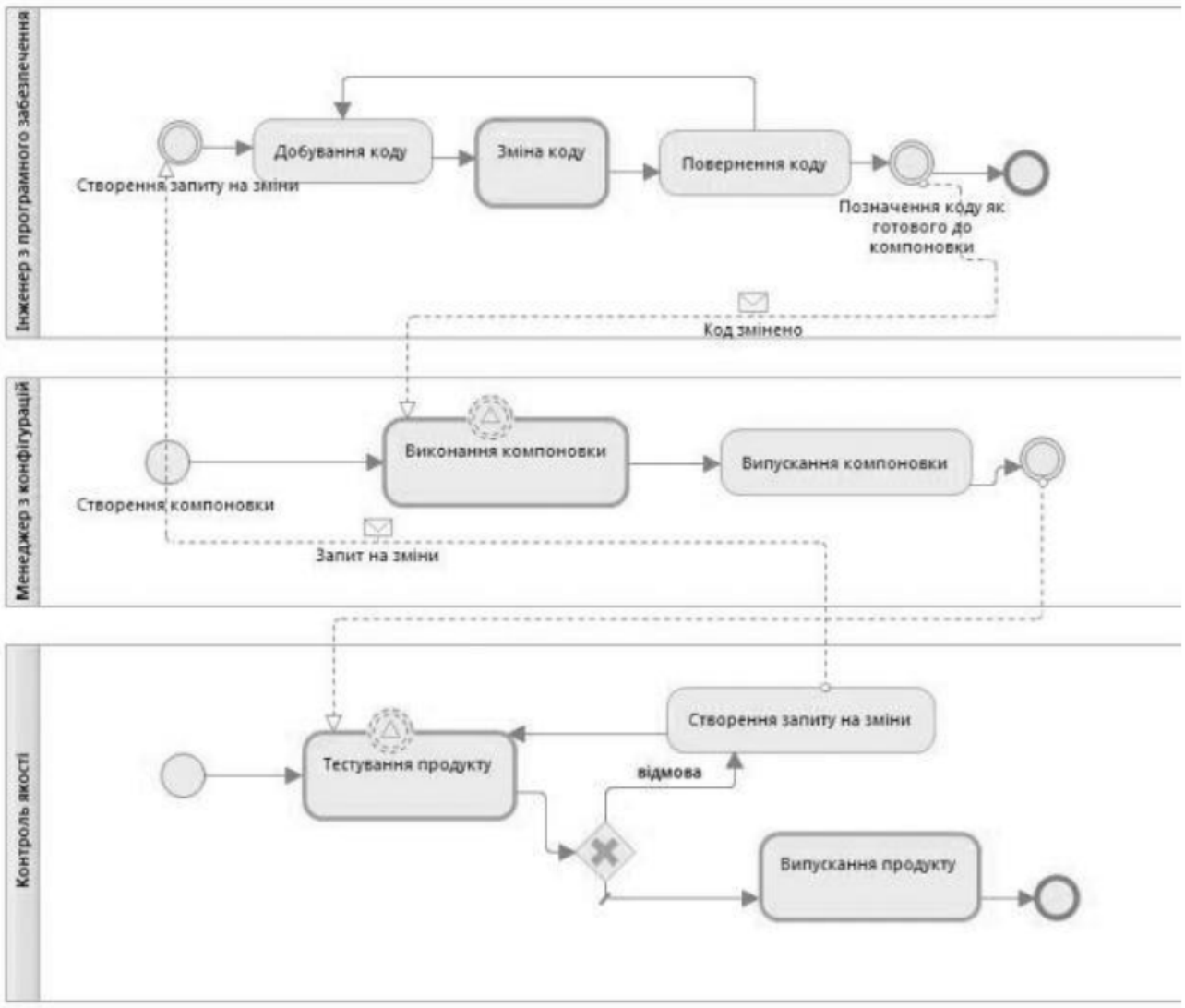

Рисунок 1 – Приклад процесів стандартної системи керування конфігурацією

При переході від проекту в чистому вигляді до того, що вимагає доставки користувачам, потрібно зібрати весь код разом. Після проведення тестування та інтеграції ваш клієнт випускає створений вами продукт.

Компоновки або збірки призначені головним чином для відділів розробки і тестування і можуть також використовуватися для пошуку помилок шляхом тестування та налагодження.

Після того як продукт був зібраний, групи тестування і розробки відразу кваліфікують продукт як готовий до випуску.

Якщо у вас розробка виконується кількома підрозділами, можна налаштувати час зборки таким чином, щоб виконувати кілька збірок за 24 години. З іншого боку, якщо збірка займає мало часу, можна виконувати її кілька разів в день.

Випуски продуктів залежать від успішності тестування компоновки та від вимог клієнтів.

Є кілька причин для автоматизації компоновки. Перша причина- узгодженість. Якщо не збирати продукт однаковим способом кожен раз, як дізнатися, відбулася реальна помилка або помилкова помилка? Ніщо так не сповільнює цикл розробки, як неузгодженість процесу складання.

Друга причина - повторюваність. Дуже часто буває так, що група розробників вже працює над наступним випуском, тоді як клієнт знаходить помилки в попередньому випуску. У таких випадках зміни в код потрібно внести в попередньому випуску, і ці попередні випуски потрібно повторно зібрати і випустити. Якщо збірка не автоматизована, цей процес може бути дуже виснажливим і схильним помилок.

І нарешті, потрібна можливість поширення зборок по декількох ресурсів в мережі обчислень (computation grid) або мережі даних (data grid). Це дозволить скоротити час циклу складання та тестування та підвищити якість і продуктивність вашої групи.

Так як зазвичай група розробників складається не з однієї людини, зв'язок між учасниками групи дуже важлива. Існує два основних види зв'язків, які потрібно враховувати при розробці системи збирання і випуску: негайне повідомлення і накопичення інформації.

При негайному повідомленні інженер з компоновки або інженер по випуску оповіщається про виникнення проблем із збіркою або про її успішному виконанні. Для цього краще всього використовувати електронну пошту.

При накопиченні інформації, вонна повинна містити повні відомості про виконану збірці. Кількість інформації про збірку, повністю залежить від вашої організації. Зазвичай для такого роду зв'язку використовується веб-сайт. Можна накопичувати такі відомості як: попередження та помилки компіляції; попередження та помилки очищення; результати тестування: пройдено/ не пройдено; результати інтерпретації (lint); файли журналів для різних етапів процесу; інформація про час; платформи та комп'ютери.