**Лекція 27. Керування розробкою програмного забезпечення.** **Керування ризиками. Керування конфігурацією.**

**Керування розробкою програмного забезпечення**

Неможливо описати й стандартизувати всі роботи, виконувані в проекті по створенню ПЗ. Ці роботи досить істотно залежать від організації, де виконується розробка ПЗ, і від типу створюваного програмного продукту. Але завжди можна виділити наступні:

Написання пропозицій по створенню ПЗ.

Планування й складання графіка робіт зі створення ПЗ.

Оцінювання вартості проекту.

Підбор персоналу.

Контроль за ходом виконання робіт.

Написання звітів і подань.

Перша стадія програмного проекту може складатися з *написання пропозицій* по реалізації цього проекту. Пропозиції повинні містити опис цілей проектів і способів їхнього досягнення. Вони також звичайно містять у собі оцінки фінансових і тимчасових витрат на виконання проекту. При необхідності тут можуть приводитися обґрунтування для передачі проекту на виконання сторонньої організації або команді розроблювачів. Написання пропозицій - дуже відповідальна робота, тому що для багатьох організацій питання про те, чи буде проект виконуватися самою організацією або розроблятися за контрактом сторонньою компанією, є критичним. Не існує яких-небудь рекомендацій з написання пропозицій, багато чого тут залежить від досвід.

На *етапі планування* проекту визначаються процеси, етапи й отримані на кожному з них результати, які повинні привести до виконання проекту. Реалізація цього плану приведе до досягнення цілей проекту. Визначення вартості проекту прямо пов'язане з його плануванням, оскільки тут оцінюються ресурси, що вимагаються для виконання плану.

*Контроль за ходом виконання* робіт (моніторинг проекту) — це безперервний процес, що триває протягом усього строку реалізації проекту. Керівник повинен постійно відслідковувати хід реалізації проекту й порівнювати фактичні й планові показники виконання робіт з їхньою вартістю. Хоча багато організацій мають механізми формального моніторингу робіт, досвідчений керівник може скласти ясну картину про стадію розвитку проекту просто шляхом неформального спілкування з розроблювачами. Неформальний моніторинг часто допомагає виявити потенційні проблеми, які в явному виді можуть виявитися пізніше. Наприклад, щоденне обговорення ходу виконання робіт може виявити окремі недоробки в створюваному програмному продукті. Замість очікування звітів, у яких буде відбитий факт "пробуксовки" графіка робіт, можна обговорити з фахівцями намічені програмістські проблеми й не допустити зриву графіка робіт.

Протягом реалізації проекту звичайно відбувається кілька формальних *контрольних перевірок* ходу виконання робіт зі створення ПЗ. Такі перевірки повинні дати загальну картину ходу реалізації проекту в цілому й показати, наскільки вже розроблена частина ПЗ відповідає цілям проекту.

*Час виконання* більших програмних проектів може займати кілька років. Протягом цього часу мети й наміри організації, що замовила програмний проект, можуть істотно змінитися. Може виявитися, що розроблювальний програмний продукт став уже непотрібним або вихідні вимоги до створюваного ПЗ просто застаріли і їх необхідно кардинально міняти. У такій ситуації керівництво організації-розроблювача може ухвалити рішення щодо припиненні розробки ПЗ або про зміну проекту в цілому для того, щоб урахувати цілі, змінилися й наміри організації-замовника.

Керівники проектів звичайно зобов'язані самі *підбирати виконавців* для своїх проектів. В ідеальному випадку професійний рівень виконавців повинен відповідати тій роботі, що вони будуть виконувати в ході реалізації проекту. Однак у багатьох випадках керівники повинні покладатися на команду розроблювачів, що далека від ідеальної. Така ситуація може бути викликана наступними причинами:

* *Бюджет проекту* не дозволяє залучити висококваліфікований персонал. У такому випадку за меншу плату залучаються менш кваліфіковані фахівці.
* Бувають ситуації, коли неможливо знайти фахівців необхідної *кваліфікації* як у самій організації-розроблювачі, так і поза нею. Наприклад, в організації "кращі люди" можуть бути вже зайняті в інших проектах.
* Організація хоче підвищити професійний рівень своїх працівників. У цьому випадку вона може залучити до *участі в проекті* недосвідчених або недостатньо кваліфікованих працівників, щоб вони придбали необхідний досвід і повчилися в більше досвідчених фахівців.

Таким чином, майже завжди підбор фахівців для виконання проекту має певні обмеження й не є вільним. Разом з тим необхідно, щоб хоча б кілька членів групи розроблювачів мали кваліфікацію й досвід, достатні для роботи над даним проектом. У противному випадку неможливо уникнути помилок у розробці ПЗ.

Керівник проекту звичайно зобов'язаний посилати звіти про хід його виконання як замовникові, так і підрядним організаціям. Це повинні бути короткі документи, засновані на інформації, що витягається з звітів про проект виконавців. У цих звітах повинна бути та інформація, що дозволяє чітко оцінити ступінь готовності створюваного програмного продукту.

У рамках курсу «Інженерія програмного забезпечення» виділені наступні ролі в групі по розробці ПЗ:

Керівник - загальне керівництво проектом, написання документації, спілкування із замовником ПЗ.

Системний аналітик - розробка вимог (складання технічного завдання, проекту програмного забезпечення)

Тестер - складання плану тестування й атестації готового ПЗ (продукту), складання сценарію тестування, базовий приклад, проведення заходів щодо плану тестування

Розроблювач - моделювання компонент програмного забезпечення, кодування.

***Планування проекту розробки програмного забезпечення***

Ефективне керування програмним проектом прямо залежить від правильного *планування* робіт, необхідних для його виконання. План допомагає керівникові передбачати проблеми, які можуть виникнути на яких-небудь етапах створення ПЗ, і розробити превентивні міри для їхнього попередження або рішення. План, розроблений на початковому етапі проекту, розглядається всіма його учасниками як керівний документ, виконання якого повинне привести до успішного завершення проекту. Цей первісний план повинен максимально докладно описувати всі етапи реалізації проекту.

Процес планування починається, виходячи з опису системи, з визначення проектних обмежень (тимчасові обмеження, можливості наявного персоналу, бюджетні обмеження й т.д.). Ці обмеження повинні визначатися паралельно з оцінюванням проектних параметрів, таких як структура й розмір проекту, а також розподілом функцій серед виконавців. Потім визначаються етапи розробки й те, які результати документація, прототипи, підсистеми або версії програмного продукту) повинні бути отримані по закінченні цих етапів. Далі починається *циклічна частина* планування. Спочатку розробляється графік робіт з виконання проекту або дається дозвіл на продовження використання раніше створеного графіка. Після цього проводиться контроль виконання робіт і відзначаються розбіжності між реальним і плановим ходом робіт.

Далі, у міру надходження нової інформації про хід виконання проекту, можливий перегляд первісних оцінок параметрів проекту. Це, у свою чергу, може привести до зміни графіка робіт. Якщо в результаті цих змін порушуються строки завершення проекту, повинні бути переглянуті (і погоджені із замовником ПЗ) проектні обмеження.

Звичайно, більшість керівників проектів не думають, що реалізація їхніх проектів пройде гладко, без усяких проблем. Бажано описати можливі проблеми ще до того, як вони виявлять себе в ході виконання проекту. Тому краще становити "песимістичні" графіки робіт, чим "оптимістичні". Але, звичайно, неможливо побудувати план, що враховує всі, у тому числі випадкові, проблеми й затримки виконання проекту, тому й виникає необхідність періодичного перегляду проектних обмежень і етапів створення програмного продукту.

План проекту повинен чітко показати ресурси, необхідні для реалізації проекту, поділ робіт на етапи й часовий графік виконання цих етапів. У деяких організаціях план проекту складається як єдиний документ, що містить всі види планів, описаних вище. В інших випадках план проекту описує тільки технологічний процес створення ПЗ. У такому плані обов'язково присутні посилання на плани інших видів, але вони розробляються окремо від плану проекту.

*Деталізація* планів проектів дуже відрізняється залежно від типу розроблювального програмного продукту й організації-розроблювача. Але в кожному разі більшість планів містять наступні розділи.

*Вступ*. Короткий опис цілей проекту й проектних обмежень (бюджетних, тимчасових і т.д.), які важливі для керування проектом.

*Організація* виконання проекту. Опис способу підбора команди розроблювачів і розподіл обов'язків між членами команди.

*Аналіз ризиків*. Опис можливих проектних ризиків, імовірності їхнього прояву й стратегій, спрямованих на їхнє зменшення.

*Апаратні й програмні* ресурси, необхідні для реалізації проекту. Перелік апаратних засобів і програмного забезпечення, необхідного для розробки програмного продукту. Якщо апаратні засоби потрібно закуповувати, приводиться їхня вартість разом із графіком закупівлі й поставки.

*Розбивка робіт на етапи*. Процес реалізації проекту розбивається на окремі процеси, визначаються етапи виконання проекту, приводиться опис результатів ("виходів") кожного етапу й контрольні оцінки.

*Графік робіт*. У цьому графіку відображаються залежності між окремими процесами (етапами) розробки ПЗ, оцінки часу їхнього виконання й розподіл членів команди розроблювачів по окремих етапах.

*Механізми моніторингу* й контролю за ходом виконання проекту. Описуються надавані керівником звіти про хід виконання робіт, строки їхнього надання, а також механізми моніторингу всього проекту.

План повинен регулярно *переглядатися* в процесі реалізації проекту. Одні частини плану, наприклад графік робіт, змінюються часто, інші більше стабільні. Для внесення змін у план потрібна спеціальна організація документопотоку, що дозволяє відслідковувати ці зміни.

***Загальні відомості про вимоги до інформаційних систем***

*Проблеми*, які доводиться вирішувати фахівцям у процесі створення програмного забезпечення, дуже складні. Природа цих проблем не завжди ясна, особливо якщо розроблювальна програмна система інноваційна. Зокрема, важко чітко описати ті дії, які повинна виконувати система. Опис функціональних можливостей і обмежень, що накладаються на систему, називається вимогами до цієї системи, а сам процес формування, аналізу, документування й перевірки цих функціональних можливостей і обмежень - розробкою вимог.

*Вимоги* підрозділяються на користувальницькі й системні. Користувальницькі вимоги - це опис природною мовою (плюс діаграми, що пояснюють) функцій, виконуваних системою, і обмежень, що накладаються на неї. Системні вимоги - це опис особливостей системи (архітектура системи, вимоги до параметрів устаткування й т.д.), необхідних для ефективної реалізації вимог користувача.

Перші кроки по розробці вимог до інформаційних систем - аналіз реалізуємості. Розробка вимог — це процес, що включає заходи, необхідні для створення й затвердження документа, що містить специфікацію системних вимог. Для нових програмних систем процес розробки вимог повинен починатися з аналізу реалізуємості. Початком такого аналізу є загальний опис системи і її призначення, а результатом аналізу — звіт, у якому повинна бути чітка рекомендація, продовжувати чи ні процес розробки вимог проектованої системи. Інакше кажучи, *аналіз реалізуємості* повинен освітити наступні питання.

1. Чи відповідає система загальним і бізнес-цілям організації-замовника й організації-розробника?

2. Чи можна реалізувати систему, використовуючи існуючі на даний момент технології й не виходити за межі заданої вартості?

3. Чи можна об'єднати систему з іншими системами, які вже експлуатуються?

Критичним є питання, чи буде система відповідати цілям організації. Якщо система не відповідає цим цілям, вона не представляє ніякої цінності для організації.

Виконання аналізу реалізуємості включає збір і аналіз інформації про майбутню систему й написання відповідного звіту. Наприклад, цю інформацію можна одержати, відповівши на наступне:

Що відбудеться з організацією, якщо система не буде введена в експлуатацію?

Які поточні проблеми існують в організації і як нова система допоможе їх вирішити?

Яким образом система буде сприяти цілям бізнесу?

Чи вимагає розробка системи технології, що до цього не використовувалася в організації?

Далі необхідно визначити джерела інформації. Це можуть бути менеджери відділів, де система буде використовуватися, розробники програмного забезпечення, знайомі з типом майбутньої системи, технологи, кінцеві користувачі й т.д.

Після обробки зібраної інформації готується звіт по аналізі реалізуємості створення системи. У ньому повинні бути дані рекомендації щодо продовження розробки системи. Можуть бути запропоновані зміни бюджету й графіка робіт зі створення системи або пред'явлені більше високі вимоги до системи.

**Методи керування ризиками у проекті.**

***Ризик*** – це небажана подія, що може мати непередбачені негативні наслідки. Якщо в проекті ідентифіковано безліч можливих подій ризику, які можуть викликати негативні наслідки, то такий проект схильний до ризику. Причиною виникнення ризиків у проекті є деякі невизначеності в плані обсягу робіт на кожного працюючого та ін. Ризики можуть бути «відомі», які визначені і оцінені, їх планують, а також планують і ризики «невідомі», які можуть з’явитися.

У новій версії стандарту PMBOK є шість наступних процедур з ризиків.

1. Планування робіт з керуванням ризиків шляхом вибору підходів і методів діяльності з їх находження.

2. Ідентифікація ризиків як визначення тих, які здатні вплинуть на реалізацію проекту і його документацію.

3. Якісна оцінка ризиків, як аналіз ризиків і умов їхнього виникнення з метою визначення їхнього впливу на успіх проекту.

4. Кількісна оцінка, як кількісний аналіз імовірності виникнення й впливу наслідків ризиків на проект.

5. Планування реагування ризиків, як визначення процедур і методів зменшення негативних наслідків ризикових подій і використання можливих переваг.

6. Моніторинг і контроль ризиків, як визначення ризиків, що залишаються, виконання плану керування ризиками проекту й оцінка дій з мінімізації ризиків.

Всі ці процедури взаємодіють одна з одною, а також з іншими процедурами. Кожна процедура виконується, принаймні, один раз у кожному проекті. Незважаючи на те, що ці процедури розглядаються як дискретні елементи із чітко визначеними характеристиками, на практиці вони можуть частково збігатися й взаємодіяти.

***Планування керування ризиками*** – це процес прийняття рішень з застосування й планування керування ризиками для конкретного проекту. Цей процес може містити в собі рішення про організацію, кадрове забезпечення, вибір кращої методології, джерел даних для ідентифікації ризику та аналізу ситуації. Сплановані роботи з керування ризиками є адекватними як рівню й типу ризику, так і важливості проекту для організації.

**Ідентифікація ризиків та їх якісна та кількісна оцінка.**

***Ідентифікацію ризиків*** проводять менеджери проекту, замовники, користувачі й незалежні фахівці шляхом опису ризиків, які здатні вплинути на проект. Ця процедура виконується як ітераційний процес і не буде ефективною, якщо проводиться нерегулярно протягом реалізації проекту. Ідентифіковані менеджерами проектів ризики переглядають аналітики проекту. Практично роботу з цими ідентифікованими ризиками виконують розробники проекту. Для формування об'єктивної оцінки стану проекту відносно наявності ризиків чи їх відсутності залучаються незалежні фахівці на завершальному процесі.

***Якісна оцінка ризиків*** – це процес проведення якісного аналізу ідентифікації ризиків, з метою швидкого реагування на них. Така оцінка визначає ступінь важливості ризику й вибір способу реагування. Доступність супровідної інформації допомагає легше розставити пріоритети для різних категорій ризиків. Якісна оцінка ризиків – це оцінка умов виникнення ризиків і визначення їхнього впливу на проект за допомогою стандартних методів і засобів. Вони допомагають частково уникнути невизначеностей, які часто зустрічаються в проекті. Постійна переоцінка ризиків відбувається протягом ЖЦ проекту.

***Кількісна оцінка ризиків*** – це визначення ймовірності виникнення ризиків і впливів їх наслідків на проект, а прийняття правильних рішень. Ця оцінка визначає::

– імовірність досягнення кінцевої мети проекту;

– ступінь впливу ризику на проект й обсяги непередбачених витрат і матеріалів, які можуть знадобитися;

– ризики, що вимагають якнайшвидшого реагування й більшої уваги, а також впливу їх наслідків на проект;

– фактичні витрати й передбачувані строки закінчення робіт в проекті.

Кількісна і якісна оцінки ризиків базується на ідентифікації ризиків і проводять їх окремо або разом, залежно від часу й бюджету.

**Планування і реагування ризиків, їх моніторинг і контроль.**

***Планування і реагування ризиків*** – це розроблення методів і технологій з зниженню негативного впливу ризиків на проект. Ці методи призначені для ефективного захисту проекту від впливу на нього ризиків. Планування як ідентифікація і розподіл кожного ризику за категоріями потребує їхнього реагування і визначення наслідків впливу ризиків (позитивно або негативно) на проект. Ця стратегія повинна відповідати типам ризиків, рентабельності ресурсів і часових параметрів. Ризики, що обговорюються під час зустрічей, повинні бути адекватними завданням на кожному процесі проекту й погоджені з усіма членам групи виконавців з менеджером проекту. Може бути кілька варіантів стратегій реагування на ризики.

***Моніторинг і контроль*** – це процедури спостереження за ідентифікацією ризиків, забезпечення виконання плану ризиків і оцінки його ефективності з урахуванням зниження ризику. Показники ризиків, пов'язані зі здійсненням умов виконання плану, фіксуються. Моніторинг і контроль супроводжує процес впровадження проекту в життя.

Якісний контроль виконання проекту надає інформацію для прийняття ефективних рішень для запобігання виникнення ризиків. Для надання повної інформації про ризики у проекту необхідна взаємодія між усіма менеджерами проекту і розробниками.

Ціль моніторингу й контролю полягає в з'ясуванні таких ситуацій:

– реакцію на ризики впроваджено відповідно до плану й необхідності змін;

– зміна ризиків у порівнянні з попередніми значеннями;

– визначення впливу ризиків і вживання необхідних заходів;

– реакція на ризики відповідно плану.

Контроль може викликати вибір альтернативних стратегій, прийняття коректив, перепланування проекту для досягнення базового плану. Між менеджерами проекту й групою ризику повинна бути постійна взаємодія й фіксація всіх змін і явищ. Звіти про виконання проекту й системи ризиків регулярно формуються.

**Керування ризиками**

***Керування ризиками*** ґрунтується на двох основних типів ризику: загальний ризик і специфічний. До загального типу ризику належить ризик, що виникає, коли наявне недостатнє розуміння вимог, брак професіоналів або недостача часу на тестування. Ризик другого типу виражається недоліками проекту (незавершеність проекту за обіцяний строк та ін.).

Для кожного можливого ризику визначається показник ступеню його ймовірності й показник витрат, пов'язаних з ризиком. Під час проведення регресивного тестування здійснюється пошук критичних помилок. Залежно від того, наскільки ця помилка критична й від того, які показники ризику діють, обчислюються збитки ризику. Діяльність керування ризиком пов’язана з виконанням таких завдань: зменшення ризику, планування ризику, резолюцію на виявлений ризик. Зменшення ризику можна досягти, якщо уникати ризику при зміні вимог, перерозподіляти ризик, відслідковувати ризик й керувати ним.

Систему керування ризиком можна представити у вигляді відношення:

***(збиток до мінімізації – збиток після мінімізації) : ціна мінімізації ризику****.*

Мінімізації ризику можна досягти прототипуванням.

Боєм [3] ідентифікував 10 найпоширеніших причин ризику в проекті:

1. Скорочення штату або набір некваліфікованих співробітників.

2. Нереалістичні плани й бюджети в проекті.

3. Розроблення функціонально неправильних програмних елементів.

4. Розроблення невдалого інтерфейсу користувача.

5. Невдала постановка вимог.

6. Постійна зміна вимог.

7. Недоліки у внутрішній організації робіт.

8. Недоліки взаємозв'язку із замовником.

9. Невміння працювати в реальному часі.

10. Обмежені комп'ютерні ресурси.

На кожному проекті з наведених причин можуть бути присутні деякі, а не усі разом ризики. Але їх треба враховувати менеджерам, що розробляють нові проекти, щоб ризиків було як менше. Тоді проект не буде провальним.

**Керування конфігурацією системи**

*Конфігурація системи* визначає конкретну версія ПС для різних ОС, комп'ютерів і містить у собі функції, об'єднані між собою процедурами зв'язку (або розгортання) і параметрами, які задають режими функціонування системи в операційному середовищі. Випуск версії різних варіантів системи здійснюється з метою постачання замовникові. Процес отримання конкретної версії системи можна представити схемою (рис.1).

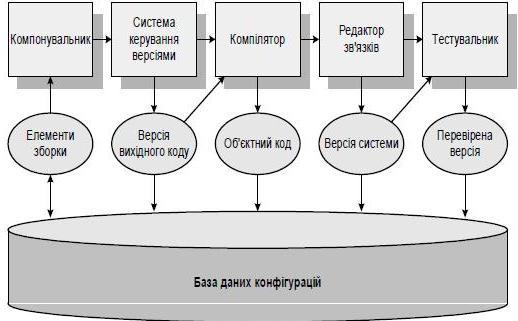


Рис.1. Схема формування версії ПС

**Конфігурація системи, її елементи**

Елементами конфігурації є:

– одиниця конфігурації (Configuration Item) – елемент, виділений для цілей керування й обробки на процесорах комп'ютера системи;

– базис конфігурації (Configuration Baseline) – набір формально розглянутої й затвердженої основи системи із складу елементів і документації, що встановлює можливість подальшого розвитку системи;

– програмні компоненти системи.

Конфігурація складається з наведених елементів і спеціалізованих процедур їхнього об'єднання в єдине ціле для функціонування й виконання компонентів у заданій послідовності. Чим більше в системі компонентів, тим більша ймовірність того, що окремі з них можуть змінюватися у зв'язку з виявленими помилками, уточненнями або доповненнями як нових функцій, так і устаткування. В ній іменуються всі елементи, що базуються на структуризації, схемі класифікації й кодування елементів, а також на методах представлення й ведення версій конфігурації з використанням вхідних елементів.

Метою керування конфігурацією є забезпечення цілісності системи з спостереженням за змінами, структурою й елементами конфігурації. Керування конфігурацією **–** дисципліна забезпечення ідентифікації елементів конфігурації системи при її створенні для проведення систематичного контролю, обліку й аудиту внесених змін, а також підтримки цілісності й працездатності системи.

До елементів керування конфігурацією також належать фізичні й функціональні характеристики, схема й версія конфігурації. Згідно з діючим стандартом IEEE Std.610–90 керування конфігурацією містить у собі такі основні завдання:

1. Ідентифікація конфігурації (Configuration Identification).

2. Контроль конфігурації (Configuration Control).

3. Облік статусу конфігурації (Configuration Status Accounting).

4. Аудит конфігурації (Configuration Audit).

Керування конфігурацією великих систем здійснюють методами і засобами забезпечення ідентифікації її елементів, контроль внесених змін і можливість визначення фактичного стану системи для подання в експлуатацію в будь-який момент часу її готовності. Це керування базується на точній і достовірній інформації про стан системи й плани проведення змін.

З формальної точки зору керування конфігурацією полягає в дисциплінованому застосуванні технічних, адміністративних методів спостереження за функціональними і фізичними характеристиками окремих пунктів конфігурації й елементів системи, а також їх змін і підготовці звітів про внесені зміни і перевірки правильності версії системи за висунутими вимогами.

Керування конфігурацією, як правило, виконує спеціальна служба, яка визначає можливі обмеження на функціонування системи в заданих умовах операційного середовища, порядок внесення змін, перевірку різних частин системи, збирання даних і облік внесених змін у систему й в документ про конфігурацію. До діяльності цієї служби належить також питання керування проектом, контролю якості й цілісності побудови конфігурації системи, здатної для її супроводу.

Структура служби залежить від складності системи, кількості виконавців та підтримуючих процесів розвитку проекту системи за вимогами замовника. Від її діяльності залежить ефективність побудови конфігурації системи (рис. 2.). Результатом керування конфігурацією є звіт з проведення змін проміжних версії системи, документації, носія системи та документа передачі версії замовнику і користувачеві.

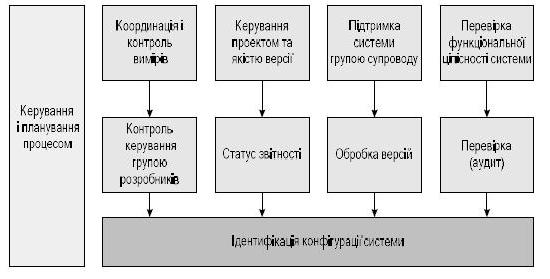


Рис. 2. Види діяльності керування конфігурацією

**Завдання керування конфігурацією системи**.

Задачі керування конфігурацією планують й виконують з урахуванням виникаючих обмежень ОС і наявності відповідного устаткування у замовника. Планування виконують менеджери вказаної служби. Вони пропонують пропозиції щодо зміни компонентів системи, проведення аналізу й визначення доцільності їх внесення у версію системи і у конфігурацію, а також оцінювання вартості цих робіт. Все це подають у вигляді переліку змін для їхньої реалізації.

Цей перелік містить у собі типи змін, строки і організацію їхнього проведення, а також дані про допуск відхилень і відмов з урахуванням вимог до проекту системи. Результатом внесення змін є нова версія системи, опис документу про проведення на ній випробувань і інструкцій для користувача. Замовник оцінює пропозиції з внесення змін і дає дозвіл на проведення найважливіших змін, що впливають на її технічні характеристики або вартість. Аналіз і контроль проведення змін конфігурації системи проводить спеціальна група служби керування. Вона також виконує систематичний облік внесених змін на всіх процесах ЖЦ.

План змін конфігурації системи затверджується формальними процедурами, розрахунками оцінок впливу змін на вартість, прийняттям рішень про зміни або відмову від них. Запити на внесення змін виконуються за цими процедурами на процесах розроблення ПП і супроводу системи. Необхідні зміни можуть проводитися одночасно з розробкою, трасуванням і при побудові нових версій системи. Кожна проведена проходить детальний аудит.

Після внесення змін проводять контроль поточної версії системи з використанням вихідних кодів систем, що у репозиторію, інструментів контролю які є у фірмах Rational's ClearCase й SourceSafe of Microsoft, а також формується версія системи. Після завершення змін і випробування системи проводять тиражування версії системи з її конфігурацією та документації для передачі замовникові.

У конфігурацію системи входять відомості про апаратні ї програмні елементи системи. В них задають обмеження з урахуванням вимог контрактів із замовником на систему, також тих, що пов’язані з аудитом, інформацією з різних джерел (специфікації вимог, описів програм в МП, звітів тощо), інструментальних засобів і рекомендацій державних або міжвідомчих стандартів.

**Планування конфігурації.**

***Планування конфігурації*** залежить від типу проекту, організаційних заходів, обмежень і загальних рекомендацій з керівництва конфігурацією. До видів цього планування належать: ідентифікація, визначення статусу й аудиту конфігурації, зміни конфігурації.

До засобів планування належать:

– система керування кодами компонент, їхнім перекладом й об'єднанням у конфігурацію системи;

– базові бібліотеки й ресурси;

– спеціальні групи контролю системи і її конфігурацій;

– СКБД для ведення проекту й зберігання змін у системі.

*Основними завданнями цього планування* є:

– фіксація різних завдань на зміни й вибір інструментарію для їхнього виконання;

– визначення людино-годин й інструментальних ресурсів, стандартів, витрат для внесення змін та ін.;

– встановлення зв'язків із замовником з проведення контролю системи і її конфігурації, а також оцінки системи;

– визначення послідовності робіт керування конфігурацією.

Результати планування відображають в плані керування конфігурацією проекту, а також у документі внесення змін у версію, конфігурацію або в саму систему.

**Ідентифікація елементів конфігурації.**

Ідентифікацію елементів конфігурації виконують методами структуризації, класифікації й іменування елементів системи і її версій. С ідентифікацією зв’язано:

* визначення стратегії ідентифікації для отримання врахованої версії системи;
* іменування елементів системи і всієї її конфігурації;
* встановлення співвідношення між кількістю виконуваних завдань і кількістю пунктів конфігурації;
* ведення версії системи (або її частин) і документування;
* вибір елементів базису конфігурації і його формальне позначення.

До засобів підтримки ідентифікації відносять бібліотеку елементів, версій і змін системи. Основу ідентифікації становить конфігураційний базис – набір формально розглянутої й затвердженої конфігураційної документації, як основи для подальшого розвитку або розроблення системи.

Виділення в продукті контрольованих одиниць конфігурації є складним завданням, яке, як правило, виконується системними архітекторами. Побудова адекватної схеми класифікації й ідентифікації об'єктів конфігураційного керування виконується одночасно зі структуризацією продукту й полягає у визначенні правил унікальної ідентифікації (кодування, маркірування):

* конфігурації продукту і її версій;
* контрольованих одиниць конфігурації і їхніх версій;
* всіх складових конфігураційного базису і їхніх редакцій.

Результатом створення і застосування схеми ідентифікації є можливість швидко й гарантовано відрізняти: різні продукти один від одного, версії одного продукту між собою, одиниці конфігурації продукту і їхньої версії.

**Формування версій й контроль конфігурації**

Версія системи містить у собі елементи конфігурації й варіант версії системи для передачі замовнику. Керування версіями полягає у виконанні дій:

* *інтеграція* або композиція коректної й остаточної версії системи з елементів конфігурації, які реалізовані на процесах ЖЦ. Функціонування коду системи залежить від апаратних засобів й інструментів, за допомогою яких будувалася система;
* *вибору інструментарію* побудови версії, оцінки можливостей середовища й засобів автоматизації процесу побудови окремих версій з коректною конфігурацією ПС і даних;
* *керування варіантами версій* із сукупності готових ідентифікованих елементів системи, що задовольняють заданим вимогам замовника.

При формуванні версій системи враховуються обмеження на розробку системи під час виконання процесів ЖЦ, які, як правило, породжують ряд відхилень від вимог на розробку елементів конфігурації системи. Наприклад, приймається рішення про зміну конфігурації і вони не погоджені із замовником.

Коли нову версію системи отримано, замовнику передають остаточну версію, конфігурацію, документацію й інструменти керування версіями для самостійного супроводу системи і внесення змін у її елементи.

**Контроль конфігурації**

***Контроль конфігурації* –** це перевірка її правильності і випробування розгортки конфігурації в процесі експлуатації системи. Він містить у собі безперервні коректування, які стосуються вже погодженого та/або затвердженого конфігураційного базису. Це обумовлює такі об’єкти контролю:

– зміни в затвердженому базисі і пов'язані з ними коректування елементів конфігурації;

– дефекти й відхилення в конфігурації продукту з затвердженого базису.

Для їхнього опису використовують формальні процедури ініціалізації, аналізу, прийняття й контролю виконання управлінських рішень з приводу запропонованих змін, виявлених дефектів і відхилень у конфігурації та/або елементів конфігурації продукту.

*Формальна обробка запитів на зміну базису.* Після досягнення взаєморозуміння з приводу вимог, архітектури та інших технічних рішень, відповідні проектні документи вважаються затвердженими й не можуть довільно модифікуватися. Тобто будь-яка потреба в зміні, що виходить від будь-якого учасника проекту, повинна пройти формальну процедуру з таких кроків:

1. Реєстрація пропозиції /запиту на зміну.

2. Аналіз впливу запропонованої зміни на наявний заділ, обсяг, трудомісткість, графік і вартість робіт з проекту.

3. Прийняття рішення з виконання цього запиту (наприклад, задовольнити, відмовити або відкласти).

4. Реалізація затвердженої зміни і її верифікація.

*Керування дефектами й відхиленнями від затвердженого* базису*.* Другою важливою складовою контролю конфігурації є керування невідповідностями між конфігурацією або елементами конфігурації продукту й конфігураційним базисом. З погляду керування всі невідповідності поділяються на дефекти й відхилення. ***До дефектів*** відносять ті невідповідності, які безпосереднє стосуються цільового використання продукту за його призначенням. Усе інше належить до відхилень. Якщо дефекти програмного продукту є негативними, то вони підлягають усуненню за такою схемою:

* реєстрація інформації про отриманий дефект /відхилення;
* аналіз та діагностика місця й причини дефекту /відхилення, оцінка обсягу, трудомісткості, строків і вартості переробок;
* прийняття рішення з усунення дефекту/відхилення, реалізація й верифікація цих недоліків.

Такого роду рішення є керованими, їх приймають керівники відповідного рівня або їхні повноважні представники. Як правило, рівень прийняття рішення про зміну програмного продукту повинен бути прийнятий на рівні узгодження або затвердження документів відповідного конфігураційного базису. Найзручнішою формою реалізації такого рішення є рада керівників з контролю конфігурації CCB (Configuration Control Board).

**Облік статусу й аудит конфігурації**

Зміст цього обліку полягає в реєстрації й наданні інформації з ефективного контролю конфігурації. Предметом обліку є інформація про поточний статус ідентифікованих об'єктів конфігурації, запропоновані і виконанні зміни, а також виявлені дефекти й відхилення від затвердженого конфігураційного базису.

Звітність про статус конфігурації є ключовим чинником прийняття проектних рішень до системи або проекту. Більше того, дані обліку статусу конфігурації, що оперативно реєструються та регулярно оновлюються, є вихідним матеріалом для формування кількісних оцінок, а саме, метрик продуктивності і якості робіт на проекті. Застосування цих метрик дозволяє приймати не тільки правильні, а й ефективні проектні рішення з створення програмного проекту.

У системі обліку статусу конфігурації накопичують зведені звіти про кількість виявлених і виправлених дефектів, що надійшли, й реалізованих запитів на зміни, динаміку внесення змін у конфігурацію в часі та ін. Цю звітність використають практично всі учасники проекту: замовники, аналітики, розробники, тестувальники, служби впровадження та якості й керівництво проекту. На її основі проводять кількісну оцінка продуктивності і якості проекту.

**Аудит конфігурації**

***Аудит конфігурації*** – це ревізія або перевірка випуску чергової версії ПС або перездачі системи замовнику. В обох випадках аудиторська робота здебільшого пов'язана з розглядом й оцінкою документації, даних, звітів і результатів іспитовій версії системи.

Аудит конфігурації проводять безпосередньо перед виходом нової версії продукту, його частини, тобто практично завжди виходять із відповідальності моменту з тих або інших зобов'язань перед замовником.

*Конфігураційний аудит* – це:

– функціональний аудит конфігурації для підтвердження відповідності фактичних характеристик конфігурації/одиниць програмного продукту висунутим вимогам до системи;

– фізичний аудит конфігурації, як підтвердження взаємної відповідності документації з фактично створеної конфігурації готового продукту системи.

*Функціональний аудит* – не є верифікацію або валідацією програмного продукту, а є перевірка того, що тестування проведено у встановленому обсязі, результати документовані й підтверджують відповідність характеристик продукту висунутим до нього вимогам. При цьому всі зміни реалізовано, критичні дефекти усунуто, а про всі виявлені відхилення від конфігураційного базису прийнято адекватні проектне рішення. Цей аудит полягає у звіренні готового продукту з документами конфігураційного базису, а також перевірки того, що цю конфігурацію побудовано відповідно до встановлених процедур і з коректних версій відповідних компонентів. Конфігураційний аудит проводять незалежними експертами, наприклад, представниками служби якості.

*Для самостійного вивчення*: Поглибити матеріал лекції за наданою літературою. Вивчення лекційного матеріалу та додаткових джерел. Розгляд запитань і виконання завдань для самостійної роботи, запропонованих на лекції.

*Література*

1. Лавріщева К.М. Програмна інженерія. Електронний підручник. URL: <http://csc.knu.ua/uk/library/books/lavrishcheva-6.pdf>.

2. Соммервил И. Инженерия программного обеспечения / И.Соммервил.–М. : Издательский дом «Вильямс», 2002. –623 с.

3. *Боэм Б.У.* Инженерное проектирование программного обеспечения. – М.: Радио и связь, 1985. – 511 с.

*Контрольні запитання для самоперевірки*.

1. Визначте поняття керування ризиком.
2. Що розуміється під процесом керування конфігурацією ПЗ?
3. Наведіть основні завдання керування конфігурацією.
4. Які дії виконуються в процесі керування версіями ПЗ?
5. Сформулюйте основні завдання обліку й аудиту.