12. Кобозева А.А. Основы общего подхода к решению проблемы обнаружения фальсификац цифрового сигнала / Електромашинобудування та електрообладнання. - 2009. - Вип.72. – С.35-41.
13. Кобозева А.А., Рыбальский О.В., Трифонова Е.А. Матричный анализ – основа общего подхода обнаружению фальсификации цифрового сигнала / Вісник Східноукр-го нац-го ун-ту ім. В.Даля. 2008. — №8(126),ч.1. — С.62—72.

Предлагается новый стеганоаналитический алгоритм детектирования наличия секретного сообщени погруженного в цифровой сигнал, который хранится в формате с потерями, с использованием мето модификации наименьшего значащего бита. Приведены результаты вычислительного эксперимента.

Пропонується новий стеганоаналітичний алгоритм детектування наявності секретного повідомлен вбудованого в цифровий сигнал, який зберігається у форматі з втратами, з використанням методу модифіка найменшого значущого біта. Наведені результати обчислювального експерименту

Proposed a new steganalysis algorithm for detecting the presence of secret messages embedded into a digi signal, which stored in lossy format, using the method of modifying the least significant bit. The results computational experiments are presented and discussed.

Рецензент: д.т.н., проф. Козловський В.
Надійшла 02.02.20

УДК 004.236.321

Павлов І.М.

МОДЕЛІ ЖИТТЄВОГО ЦИКЛУ ПРОГРАМНИХ МЕХАНІЗМІВ ЗАХИСТУ КОМПЛЕКСНОЇ СИСТЕМИ ЗАХИСТУ ІНФОРМАЦІЇ

З розвитком методів, технологій створення і експлуатації засобів захисту інформації одного боку, відбувається все більша деталізація етапів життєвих циклів (конструювання), другого – відбувається розширення життєвого циклу, набір все більших об'ємів інформації знань, даних про розробку, виготовлення та функціонування засобів захисту.

Поняття моделі життєвого циклу виникло у зв'язку з необхідністю інтеграції процесі проектування, виготовлення і використання складних технічних систем, з одного боку, розвитком системного підходу як методологічної основи аналізу і синтезу систем – другого боку. Метою введення цього поняття можна рахувати об'єднання різних технологій пов'язаних з процесом існування об'єкта дослідження, в єдину технологію, в якій визначен закони перетворення ролевих функцій компонентів локальних технологій.

Ця інтеграційна мета поділяється на наступні підцілі:

розподіл загального процесу на основні етапи (фази);

забезпечення безперервності фаз в рамках загального процесу;

визначення основних характеристик фаз процесу;

дослідження взаємного впливу етапів;

використання інтегрального критерію ефективності життєвого циклу (функціоналу від часткових критеріїв фаз);

забезпечення управління процесом реалізації фаз життєвого циклу;

Стосовно поняття проектування модель життєвого циклу визначається як упорядкована сукупність змін між початковим і кінцевим станом системи. При цьому початковий стан системи починається з моменту виникнення задуму (ідеї) або початку фінансування процесу її створення. А кінцевий стан починається з моменту закінчення діяльності у зв'язку з фізичним або моральним старінням, заміною або переобладнання в якісно новий об'єкт.

В [1] були розглянуті порядок аналізу та моделювання життєвого циклу комплексної системи захисту інформації (КСЗІ) на основі існуючих моделей інформаційних систем. Крім того показані приклади моделей життєвого циклу КСЗІ, які повинні бути проаналізовані розробниками на етапах проектування та розробки КСЗІ. В подальшому під час аналізу структури КСЗІ розробники, на основі розглянутих моделей життєвого циклу КСЗІ повинні проаналізувати та змоделювати порядок роботи механізмів захисту КСЗІ, які мають свої

особливості та моделі. Тому *метою статті є* визначення моделей життєвого циклу програмних механізмів захисту, які є основою у складі КСЗІ.

Розглянемо моделі життєвого циклу програмних механізмів захисту:

1. Ітераційна модель життєвого циклу програмних механізмів захисту комплексної системи захисту інформації

В ітераційній моделі (рис.1) розробка КСЗІ ведеться ітераціями з циклами зворотного зв'язку між етапами. Коректування між етапами дозволяють враховувати реально існуючий взаємний вплив результатів розробки на різних етапах. Час життя кожного з етапів розтягується на весь період розробки.

Така модель ще має назву моделі життєвого циклу КСЗІ з проміжним контролем, що є розвитком каскадної моделі (см.[1]).

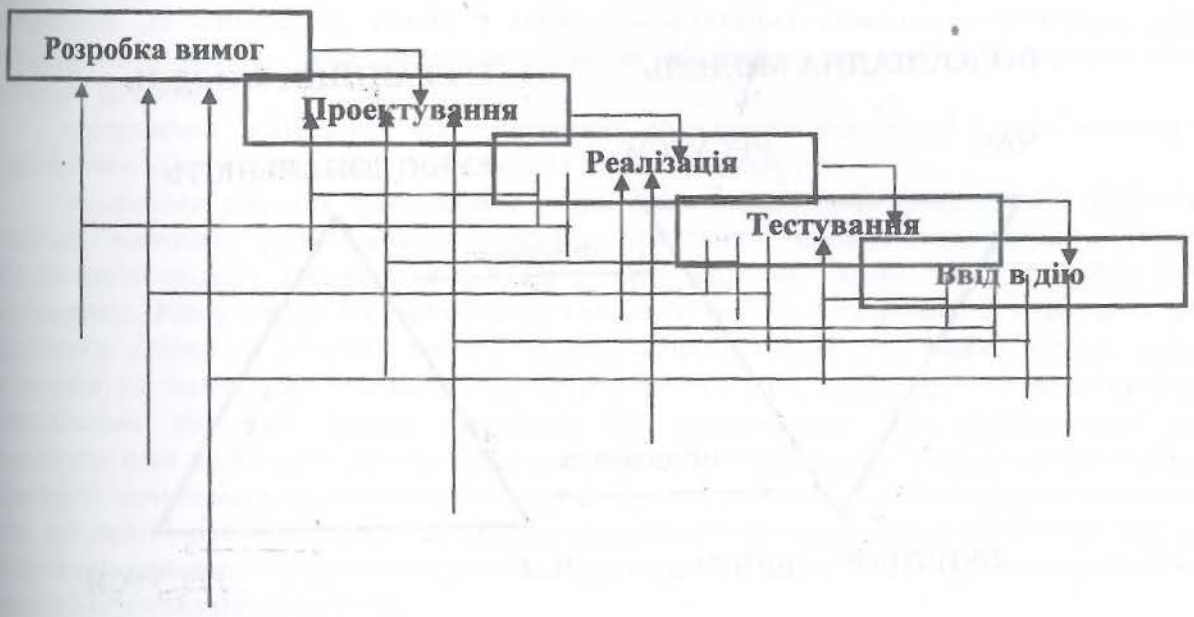

Рис. 1. Ітераційна модель життєвого циклу КСЗІ (модель життєвого циклу КСЗІ з проміжним контролем)

Основні переваги ітераційного підходу можна сформулювати так:

можливість зменшити вплив критичних ризиків на ранніх стадіях проекту з мінімальними витратами;

можливість організувати плідний зворотній зв'язок з користувачами з метою створення програмних продуктів захисту, які реально відповідають їх потребам;

приділяється основний акцент на найбільш важливі і критичні напрямки розробки програмних продуктів захисту;

безперервне ітераційне тестування кінцевого продукту, яке дозволяє оцінити успішність всієї роботи в цілому;

первинне виявлення невідповідностей між вимогами, моделями і програмним кодом;

більш рівномірна загрузка розробників та замовників;

ефективне використання досвіду, який мається у розробників проекту;

реальна оцінка існуючого стану проекту.

Як недолік цієї моделі – ця модель не дозволяє оперативно враховувати зміни і уточнення вимог до системи, які раптово виникають. Узгодження результатів розробки з користувачами проводиться тільки в точках, які плануються після завершення кожного з етапів робіт, а загальні вимоги до систем зафіксовані в технічному завданні на весь час

створення системи. Таким чином, користувачі частіше отримують систему, яка задовольняє їх реальним потребам.

2. Водоспадна модель життєвого циклу програмних механізмів захисту комплексної системи захисту інформації

Водоспадна модель життєвого циклу програмних механізмів захисту КСЗІ визначає послідовне виконання різних етапів діяльності, включаючи аналіз вимог, проектування, кодування і тестування окремих модулів (компонентів), тестування зборок і інтегроване тестування всього кінцевого продукту. При цьому пропонується чітке розгородження етапів, на яких набір документів, відпрацьований на попередніх етапах, передається в якості вхідних даних для наступного етапу. Таким чином, кожен вид діяльності виконується на однієї фази життєвого циклу програмних механізмів захисту КСЗІ. Рух в зворотному напрямку неможливий.

Рис. 2. Співвідношення водоспадної та ітераційної моделей життєвого циклу програмних продуктів КСЗІ

На приведеному рис.2. добре проглядається різниця водоспадної і ітераційної моделей життєвого циклів. Водоспадний підхід припускає фіксацію функціональності програмного забезпечення і можливість змін часу і ресурсів. При водоспадному підході замовник притягається до участі в проекті тільки на ранньому етапі (для визначення вимог) або в випадку необхідності внесення змін в проект. Він може оцінити тільки кінцевий результат, який може не відповідати його уяві.

В ітераційному підході пропонується участь замовника в проекті на усіх етапах. Ітераційний підхід дозволяє полегшити і спростити процес зміни функціональності програмних продуктів захисту.

Розглянемо етапи життєвого циклу водоспадної моделі:

Аналіз. На етапі аналізу визначається задача, яку повинні виконувати програмні механізми захисту. Результатом виконання цієї фази є сукупність вимог до програмного продукту.

Проектування. На цьому етапі вимоги, які виявлені під час аналізу перетворюються в опис принципів: рішення – документ, у відповідності з яким приймаються конкретні рішення при реалізації програмних продуктів захисту. Основним висновком другої фази є отримання проекту, який може включати до себе текст програми, алгоритми, таблиці, математичні

формули, тощо. Детальне проектування визначає виділення компонентів програмного забезпечення, визначення структури програм та методів їх взаємодії.

Реалізація. По завершенні проектування наступає етап реалізації, на якому створюються і тестуються конкретні програмні модулі захисту, які були визначені на етапі проектування. Головним результатом цього етапу є модулі вихідного коду і автономні тести модулів. Після реалізації переходять до тестування системи, а потім до здачі програмних продуктів в експлуатацію.

Впровадження і експлуатація. Готовий програмний продукт передається до дослідної експлуатації, і після усунення недоліків програмні механізми захисту ставляться на супроводження де починається виробнича експлуатація програмних продуктів до їх морального старіння і виводу з експлуатації.

Недоліки водоспадної моделі життєвого циклу:

–накопичення різних помилок, які допускаються на ранніх етапах проекту. Коли тільки до кінця проекту стає відомим, що виникли помилки, які були допущені, то будь-яке повернення до попередніх стадій з метою виправлення помилок становиться дорожче запланованого. Метод "водоспаду" не дозволяє ефективно виявляти і блокувати наслідки подібних ризиків;

–неоправдане збільшення часу реалізації, збільшення бюджету і риск повного зриву проекту внаслідок накопичення помилок від етапу до етапу;

–Усі ключеві рішення приймаються тоді, коли у аналітиків і розробників нема повного уявлення системи. Дуже важко покласти реальний процес створення програмного забезпечення захисту інформації в КСЗІ в таку жорстку схему, тому постійно виникає необхідність повернення до попередніх етапів з метою уточнення і перегляду раніше прийнятих рішень. З початку проекту перед розробниками стає важка задача: повністю визначити усі вимоги до системи захисту інформації. Для цього необхідно визначитися з замовниками: що вони хочуть захищати, яку інформацію? При проектуванні можуть виникнути нові проблеми, які необхідно, також, обговорювати з замовниками, в разі чого виникнуть нові вимоги до системи захисту. В процесі реалізації і тестування частіше виникає таке, що деякі з раніше прийнятих рішень неможливо виконати або виявляється, що вимоги не були достатньо деталізовані і їх реалізації некоректна. Потрібно повертатися назад на етап аналізу і переглядати ці вимоги.

Метод водоспаду не дає можливостей швидкої адаптації до змін, особливо на кінцевих стадіях життєвого циклу програмних продуктів захисту інформації. Кінцевий продукт може виявитися непотрібним у зв'язку неточного викладення вимог або зі змінами, які були пов'язані з великим часом створення програмного продукту.

Водоспадна модель життєвого циклу добре працює в проектах, де вимоги до програмного продукту чітко викладені і не повинні змінюватися, використовувати замовника в процесі розробки не треба. Це стосується проектів, де складність визначається необхідністю реалізації складних алгоритмів а роль і об'єм інтерфейсу користувача невеликий.

3. V-подібна модель життєвого циклу програмних механізмів захисту комплексної системи захисту інформації

V-подібна модель була запропонована для того, щоб усунути недоліки ітераційної та водоспадної моделей. А назву ця модель отримала за своє специфічне графічне представлення (рис.3). В літературі ця модель ще має назву шарнірної.

Рис. 3. V-подібна модель життєвого циклу програмних продуктів КСЗІ

V – подібна модель дала можливість значно підвисити якість розробки програмних продуктів за рахунок орієнтації на тестування, а також в цілому вирішила проблему відповідності створеного продукту вимогам, які надаються замовником, так як в цій моделі маються процедури верифікації і атестації на ранніх стадіях розробки (пунктирні лінії рис. 3 вказують на залежність етапів планування, постановки задачі та приймання продукту).

Однак в цілому V – подібна модель є модифікацією ітераційної моделі і має її недоліки. Тобто, вона також слабо пристосована до можливих змін вимог замовника. Коли процес розробки займає великий час (до років), то отриманий в результаті продукт може оказатися фактично непотрібним замовнику, оскільки його потреби істотно змінилися. В той час актуальним і є питання науково-технічного прогресу: вимоги до програмного забезпечення виходять з досягнень в області апаратно-програмного забезпечення, однак ІТ-сфера розвивається швидко, і процес розробки, який затримується, спроможний привести до створення продукту, який базується на застарілих технологіях і стає неконкурентоспроможним ще до своєї появи.

Важливим є питання планування показників функціональності, оскільки в цих моделях воно є не більш як допущення: тобто, визначити, яку швидкість обробки даних забезпечить програмний продукт, скільки він буде займати пам'яті. Коли подібні вимоги чітко фіксуються в умовах договорів між замовником і виконавцем, то по закінченні проекту це стане не важливим або застарілим.

4. Модель життєвого циклу програмних механізмів захисту комплексної системи захисту інформації на основі створення прототипів

Оскільки попередні моделі були побудовані в області інформаційних систем, то вони не в повному обсязі враховували специфіку виробництва програмних продуктів. Однак наступні моделі більш орієнтовані на особливості цього виду діяльності, який має багато принципових відмінностей. Однією з таких моделей є модель життєвого циклу на основі створення прототипів рис.4.

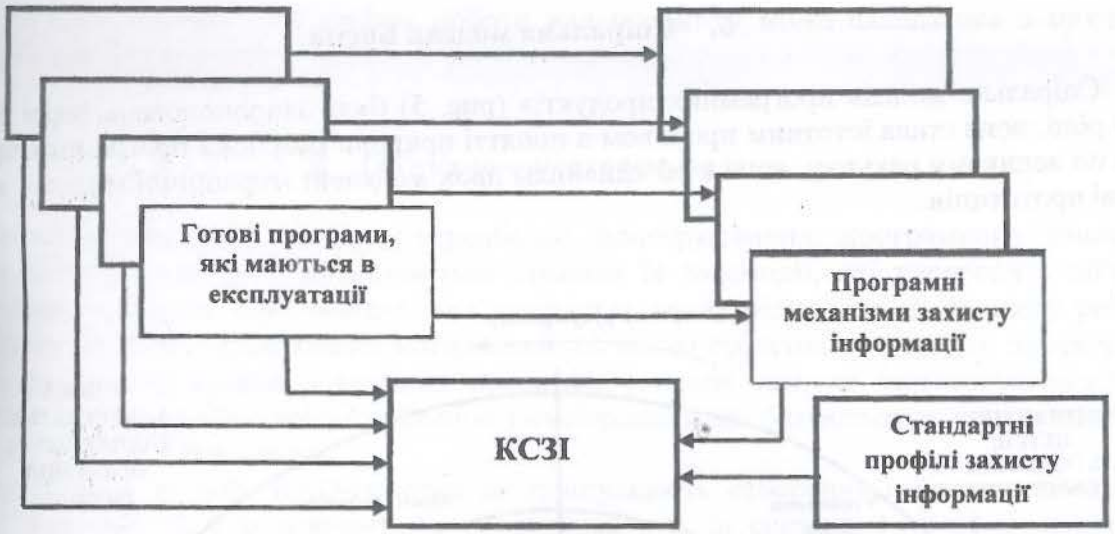

Рис. 4. Порядок створення програмних продуктів захисту інформації на основі прототипів

У зв'язку з тим, що замовник часто не є фахівцем в області захисту інформації, він достатньо погано уявляє специфіку програмного продукту. Для того, щоб подолати інформаційний бар'єр між замовником і розробником та знизити ризик отримання продукту, який не відповідає вимогам захисту інформації, став застосовуватися підхід, який направлений на створення прототипів, які уявляють собою повністю або частково робочі моделі готової системи. Він дозволяє значно покращити відношення між усіма учасниками процесу за рахунок послідовного, еволюційного розвитку системи захисту на основі ітераційного уточнення прототипів.

Застосування прототипів подібно зменшеним макетам споруд в архітектурі – вони дають можливість наглядно уявити кінцевий результат, їх побудова і зміни менш складні по відношенню з побудовою самої споруди.

Однак, не зважаючи на переваги, ця модель також не стала панацеєю. Основні її проблеми лежать в площині взаємовідношень "замовник-розробник": перший зацікавлений в створенні як можна більш складних прототипів для того, щоб знизити ризик отримання неадекватної системи, в той час як для другого кожен новий прототип означає додаткові витрати часу і ресурсів, що приводить до зниження рентабельності проекту.

5. Інкріментна модель життєвого циклу програмних механізмів захисту комплексної системи захисту інформації

Інкріментна модель дозволяє вводити в експлуатацію програмні продукти по частинам, а відповідно розробляти і постачати елементи програмних продуктів поступово. На цьому і побудована ця модель, яка дозволяє дроблення програмних продуктів на відносно незалежні складові частини, які розробляються і вводяться в експлуатацію кожна окремо.

Така модель вигідна як для замовника, так і для розробника системи, оскільки дозволяє рухатися вперед, узгоджуючи інтереси обох сторін.

Однак у цієї моделі є недоліки. Ділення на функціональні блоки в цілому уповільнює процес, так як виникає необхідність забезпечення їх взаємодію. Для багатьох рішень цій метод недоцільний, оскільки з цих рішень неможливо викреслити окремі складові, які можуть бути поставлені і функціонувати незалежно. Істотно підвищується навантаження на керівний персонал у зв'язку з ускладненням задач по координації робіт над окремими складовими системи, підвищується вартість внесення змін в готові компоненти, які вже встановлені і працюють у замовника.

6. Спіральна модель Боема

Спіральна модель програмних продуктів (рис. 5) була запропонована Баррі Боемом 1988 році, вона стала істотним проривом в понятті природи розробки програмних продукті хоча по великому рахунку, вона є об'єднанням двох моделей: ітераційної моделі і моделі н основі прототипів.

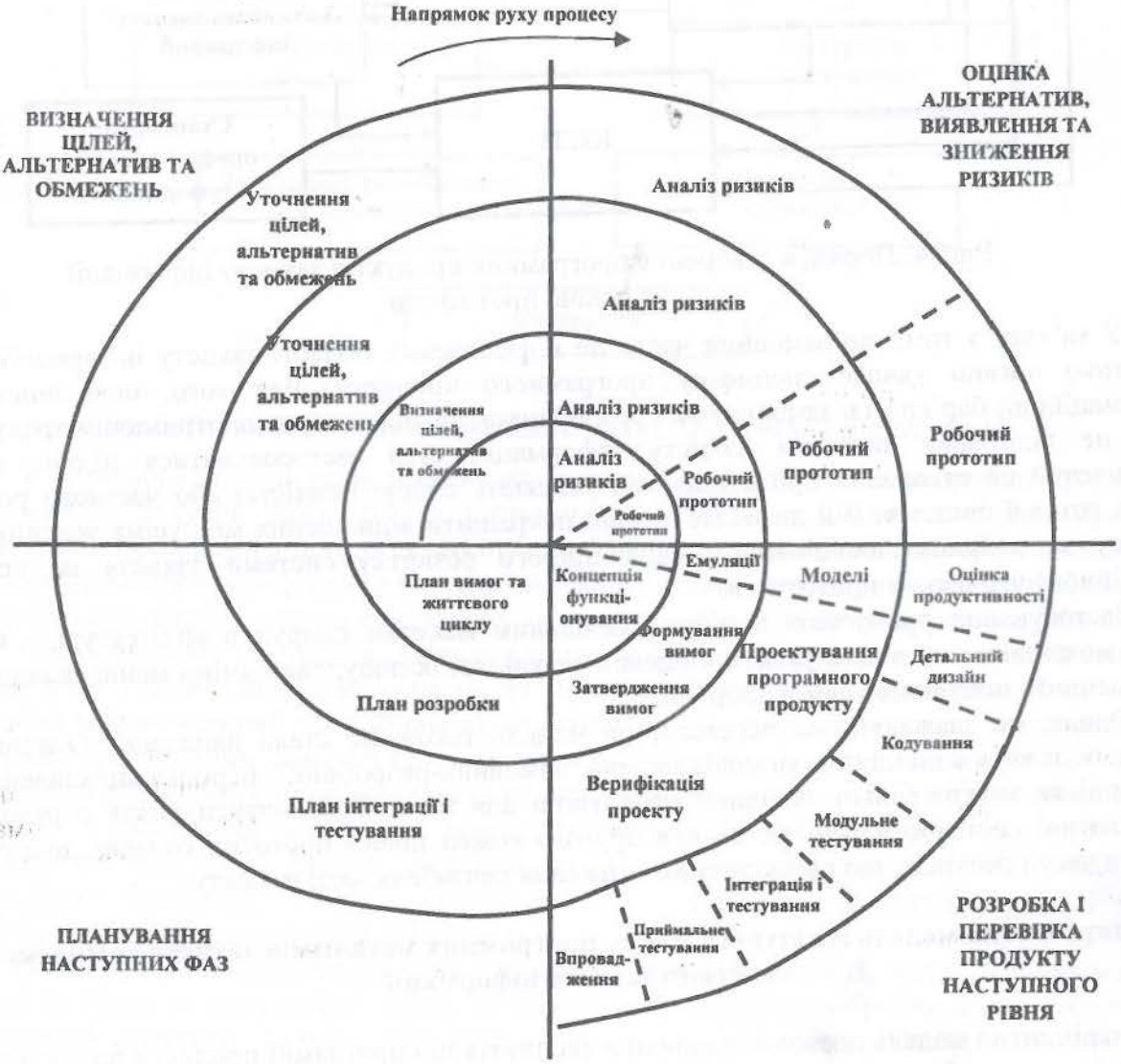

Рис. 5. Спіральна модель життєвого циклу програмних механізмів захисту КСЗІ

Спіральна модель Боема сфокусована на проектуванні. Розробка програмних продуктів, як правило, проходить на останньому вітку спіралі, однак перед цім проходить декілька ітерацій проектування на основі створення прототипів, при цьому кожна ітерація включає до себе стадію виявлення і аналізу ризиків та найбільш складних задач.

Оскільки спіральна модель, в основному, охоплює проектування, то в первинному вигляді вона не отримала великого розповсюдження в якості методу управління всім життєвим циклом створення програмних продуктів захисту інформації. Однак головна ідея

моделі полягає в тому, щоб процес роботи над проектом може складатися з циклів, які проходять одні і теж етапи.

7. Об'єктно-орієнтовна модель.

Об'єктно-орієнтовна модель передбачає конструювання програмного рішення з готових об'єктів, для яких визначаються правила їх взаємодії, які переводять об'єкти з одного стану в інший. Така модель, яка припускає повну відповідність процесу розробки положенням об'єктно-орієнтованої методології (об'єктно-орієнтований аналіз, проектування, програмування), ефективна в крупних проектах, а також там, де застосовуються засоби швидкої розробки (RAD–Rapid Application Development), які базуються на цих технологіях і мають готові бібліотеки класів.

Однак самі по себе RAD-системи не припускають створення об'єктно-орієнтованих рішень. Програмісти, в основному, не використовують ці системи і не оформлюють свої рішення у вигляді класів, які придатні для повторного використання. Таким чином ця модель використовується в великих проектах, де приділяється увага етапам аналізу і проектування, а також жорстко контролюються правила, які встановлені розробниками.

Рис. 6. Об'єктно-орієнтовна модель життєвого циклу програмних продуктів механізмів захисту КСЗІ

Таким чином моделі життєвого циклу програмних механізмів захисту КСЗІ і ступінь їх практичного застосування, в якості обов'язкового або рекомендованого документу, залежить від ролі конкретного програмного продукту в КСЗІ. Повинна бути визначена відповідна модель життєвого циклу системи, в якій програмний продукт стає її частиною. Встановлення цього допоможе визначити, можна лі використати конкретну модель для розробки, експлуатації або супроводження програмного продукту. Програмні засоби можуть бути постійно (резидентно) розміщені в комп'ютерах, вбудовані як частина програмно-апаратних засобів або інтегровані в об'єкт технічних засобів захисту інформації. В будь якому випадку замовлення, поставку, розробку, експлуатацію або супроводження програмних засобів захисту інформації КСЗІ необхідно координувати з аналогічними процесами, які відбуваються в КСЗІ та інформаційній системі в цілому.

Висновок. Для проекту системи повинен бути проведений вибір однієї або декіль відповідних моделей життєвого циклу. Необхідно встановити, є модель життєвого цик програмного механізму захисту складовою частиною моделі життєвого циклу КСЗІ. Кожн модель життєвого циклу має процеси, які можуть бути виконані послідовно, повторно аб комбіновано. Процеси повинні бути відображені в моделі, яка обрана, з точки зору створен модифікованого, структурованого і запланованого продукту. Результати одного процесу моделі життєвого циклу повинні бути передані наступному. В цьому випадку відповід документи створюються до закінчення визначеного процесу та початку наступної роботи.

Список літератури:

1. Павлов І.М. Неформальний опис життєвого циклу комплексної системи захисту інформації [Текст] І.М. Павлов, В.О. Бірюков // Сучасний захист інформації. – Киев.: 2011. – № 2. – С. 58 – 69.
2. Жук К.Д. Системные методы в программировании жизненных циклов новой техники [Текст] / К.Д Жук // Автоматизация проектирования сложных систем. – Москва.: 1999. – С. 15-26.
3.Хорошко В.А. Модель системы защиты информации [Текст] / В.А. Хорошко // Захист інформації. Київ.: 1999. – №. 1. – С. 5 – 11.
4. Третьякова О.О. Модель комплексной системы защиты информации [Текст] / О.О. Третьякова Сборник научных трудов НАУ "Защита информации". – Киев.: 2002. – Вып. 1. – С. 3 – 10.
5. Павлов И.Н. Проектирование систем защиты информации. Фор-мальный подход [Текст] / И.Н Павлов // "Правове, нормативне та метрологічне забезпечення систем захисту інформації в Україні". – Київ 2005. – Вип. 11. – С. 54 – 59.
6. Михайлов С.Ф. Информационная безопасность. Защита информации в автоматизированны системах. Основные концепции. / С.Ф. Михайлов, В.А. Петров, Ю.А. Тимофеев // – М.: Связь, 1995. – 56 с.
7. Герасименко В.А. Комплексная защита информации в современных средствах обработк информации / В.А. Герасименко // Зарубежная радиоэлектроника. – Москва.: 1993. – № 2. – С. 35 – 38.
8. Малюк А.А. Информационная безопасность: Концептуальные и методологические основы защит информации. / А.А. Малюк // – М.: Высшая школа, 2004. – 280 с.
9. Павлов І.М., Радзівілов Г.Д. Формальное описание процесса проектирования комплексных систе защиты информации в информационно − телекоммуникационных системах [Текст] // Вісник ДУІКТ. Київ 2010. – Т. 8. – №1. – С.84 – 93.

В статті розглянуті моделі життєвого циклу програмних механізмів захисту, які є основою у складі КСЗІ
Ключові слова: комплексна система захисту інформації, моделі життєвого циклу, проектування програмні механізми захисту інформації.

В статье рассмотрены модели жизненного цикла программных механизмов защиты, которые являютс основою в составе КСЗИ.
Ключевые слова: комплексная система защиты информации, модели жизненного цикла, проектирование программные механизмы защиты информации.

The models of life cycle of protection program mechanisms as a part of complex system of informatio protection are highlighted in the article.
Keywords: complex system of information protection, life cycle model, designing, program mechanisms o information protection.

Рецензент: д.т.н., проф. Лєнков С.В
Надійшла 28.01.201

УДК 004.347.7
Скачек Л.Н. (ДУІКТ)

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ ПРЕДПРИЯТИЯ

Анализ состояния проблемы и постановка задачи исследования. В последнее врем наблюдается активизация процесса приватизации промышленных предприятий Украины, чт приводит к образованию на них нового негосударственного сектора информационного ресурса Указанный сектор охватывает коммерческую (промышленную) тайну предприятия и по своем правовому статусу и области применимости принципиально отличается от традиционного дл