Створення ПЗ складний процес.

2) вироблення умов

3) Вироблення плану кпз

4) Розробка архітектури пз, або високорівневе проектування.

5) Детальне проектування.

6) Кодування і відлагодження.

7) Блочне тестуванння.

8) Інтеграційне тестування.

9) Інтеграція.

10) Тестування системи.

11) Корегувальний супровід.

Таким чином термін КПЗ описує: Детальне створення робочої програмної системи. Саме тому він описує частину процесів при побудові ПЗ:

Визначення проблеми, розробка вимог, корективна підтримка, інтеграція, інтеграційне текстуванння, архітектура ПЗ, Системне тестуванння, планування конструювання, кодування та відлагодження.

Конструювання називають кодуванням, або програмуванням. Хоча воно включає і відлагодження, і детальне проектування, блочне та інтеграційне тестування, та інші процеси. Дана область знань дуже сильно пов'язана з іншими областями. З Проектування та тестуванням. Зв'язки між ними, та іншими областямаи конструювання значно залежить від обраної моделі життевого циклу коснтруювання. На прикладному рівні конструювання дуже сильно залежить від інструментальних та методичних засобів програмної інженерії. Задачі пов’язані з конструюванням:

1. Первірка викнання умов, необхідних для успішного конструювання.
2. Визначення способів подальшого тестуванння коду.
3. Проектування та написання класів і методів.
4. Вибір керуючих структур, вибір блоків команд.
5. Блочне інтеграційне тестування, відлагодження власного коду.
6. Взаємний огляд коду, та низкорівневих програмних структур, членами команди.
7. Форматування та коментування коду.
8. Інтеграція програмних компонентів створених окремо.
9. Оптимізація коду, яка направлена на підвищення його швидкодії. І зниження міри використання ресурсів.

З іншого боку видів діяльності, що проходять в процесі розробки ПЗ, до конструювання не відносяться. Керування процесом розробки, напрацювання вимог, розробка високорівневої архітектури програми, проектування інтерфейсу користувача, тестування системи та її супровід. Для кожно з цих пунктів є своя галузь знань.

Єтапи КПЗ. Основні методи КПЗ

Кпз – охоплює такі стадії:

1 ) передпроектний етап

* 1. Діагностичне обстеження об’єкта. Збір та аналіз інформації про об’єкт конструювання.
  2. Вибір складу підсистем модулю і задач автоматизації.
  3. Розробка тех документації, тех економ обгрунтування, технічного завдання.

2) Проектна стадія та кодування.

3) Стадія введення в експлуатацію.

Методи КПЗ

Усі методи КПЗ класифікуються таким чином :

1. За технологією конструювання.
2. За ступенем коснтруювання.
3. За організацією степенів конструювання.

3.1) Аналіз і синтез.

3.2) Композиція.

3.3) Моделювання та фомалізація.

Ці методи поділяються на типове, оригінальне, автоматизоване конструювання.

Оригінальне – ніяким чином не використовуються не готові рішення, ні авт. Системи. Все пишеться з нуля.

Типове – Використовуються шаблони, рішення, моделі. Поділяється на елментне, підсистемне, об’єектне.

Метод «Зверху ваниз», деформалізація процесу коснтруювання здійснюється у вигляді графа дерева. А конструювання можна розпочинати з будь якої задачі у вигляді «графа дерева» .

Метод неперервності розвитку системи.

Засоби КПЗ. Види забезпечення КПЗ.

Засоби КПЗ, та їх класифікація. При вибору процесу КПЗ застосовують такі засоби : Технічні(локальна мережа конструювання.), мовні.

Усі процеси КПЗ поділябть на локальні та комплексні.

Локальні – для автоматизації окремих конструкторських робіт і можуть використовуватись окремо в процесі коснтруювання окремо один від одного. До них належать:

1. Генератори програм. Призначенні для реалізації типових програми керуванння данних.
2. Системи програмуванння.
3. СУБД (системи управління Базами Даних.)
4. Системи ТЕО розробки, призначення для розробки інтерактивної взаємождії користувачів та еом.
5. Системи прикладних програм.
6. Окремі елементи комплексних засобів.
7. Автоматизації планування черговості плануванння робіт.
8. Котролю управління обробкою даних.
9. Типові проекти АСУ.
10. Системи підтримки.

Види КПЗ:

1. Перефирія.
2. Математичне забезпечення. Суккупність методів моделей та алгоритмів необхідних для автоматизованого коснтруювання.
3. Програмне забезпечення. Суккупність програм, описів та інструкцій складених на основі математичного забезпечення. І призначених для реалізації проектних процедур на ЕОМ.
4. Лінгвістичне забезпечення – Сукупність мов програмуванння, програмування, та правил формалізації цих мов.
5. Інформаційне забезпечення – сукупність відомостей поданих у базах даних, і базах знань, що містять нормативи/довідкові дані, Типові проектні рішення, а також правила проектного конструювання.
6. Методичне забезпечення – сукупність документів, що встановлюють правила та інструкції з експлуатації підсистем Сапорт.
7. Організаційне забезпечення – сукупність документів, що встановлюють організаційну установу КПЗ, форми та порядок проходження документів, порядок взаємодії посадових осіб, підрозділів та відділів проектної організації.

КПЗ за sweebok

Галузь знань КПЗ поділяється на наступні елементи:

КПЗ –>

1. Основи коструювання
   1. Мінімізація
   2. Очікування змін
   3. Коструювання з можливістью перевірки
   4. Стандарти в конструюванні.
2. Керування конструюванням

2.1) Моделі конструювання

2.2) Планування конструюванння

2.3) Зміни в коснтруюванні.

3) Конструювання ПЗ

3.1) ПРоектування в конструюванні.

3.2) Мови конструювання

3.3) Кодування

3.4) Тестування

3.5) Повторне використання

3.6) Якість

3.7) Інтеграція.

Мінімізація складності – мінімізація складності в КПЗ. Особливо важлива для процесу перевірки (верифікації),і тестування результатів конструювання. Вона досягається а допомогою звертання особливої уваги на створення простого коду. Який легко читається іноді жертвуючи прагненням зробити його ідеальним. Це не означає що повинно обмежуватись застосування тих чи інших розвинених мовних можливостей засобів програмування. Мається на увазі лише надання більшої значимості, читабельності, простоті тестування. Прийнятного рівня продуктивності , та задоволенню заданих критеріїв. Замість постійного вдосконалення коду. Без огляду на терміни функціональність та інші характеристии і обмежання проекту. Мінімізація складається зокрема слідуванням стандартів. Використання низки специфічних технік кодування. І підтримкою практик - спрямованих на забезпечення якості в конструюванні.

Очікування змін – існує безліч причин тому, що більшість програмних систем з часом змінюються. Також є однією з рушійних сил КПЗ. Більше того, програмні засоби є частиною середовища що змінюється і повинні змінюватисьразом з ним. А іноді і бути причиною змін самого середовища. Підтримується рядом технік кодувань.

Конструювання з можливістью первірки. Конструювання для перевірки припускає що побудова програмних систем повинна вестись таким чином, щоб сама програмна система допомагала вести пошук причин збоїв. Будучи прозорою для застосування причин первірки. Як на стадії незалежного тестування, так і в процесі операційної діяльності експлуатації. Коли особливо важлива можливість швидкого виявленя і виправлення помилок. Серед технік на досягнені такого результату конструювання вирізнають:

- Погляд оцінка коду.

- Модульне тестування.

- Структурування коду для, і сплільно з застосуванням Автоматизованих засобів тестування.

- Застосування складних, або важких для розуміння мовних структур.

Стандарти конструювання.

Стандарти які безпосередньо застосовуються при конструюванні включають

1. Комунікаціні методи. (Правила форматів документів, оформлення вмісту.)
2. Мови програмування, та стилі програування.
3. Платформи. (Стандарти програмних інтерфейсів.)
4. Операційні системи (Win32 api)
5. Інструменти не термінів середовищ розробки, але можливих засобів конструювання.
6. Аспекти поведінки коду.

Використання зовнішніх стандартів.

КПЗ залежить від зовнішніх стандартів. Пов'язаних з мовами програмування, інструментальним забезпеченням. Тех інтерфейсами і взаємним впливом КПЗ та інших галузей програмної інженерії.

Стандарти ствоюються різними організаціями наприклад Консорціум ОMG, наприклад Корба Юмл, МДА. Міжнародними організаціями зі стандартизаціями, виробниками платформ оп середовищ.

Дозволяють визначити повний набір правил, які застосовуються в комнаді, або в організації в цілому.

Використання внутрішніх стандартів.

Певні стандарти та процедури також можуть бути створені всередині організації або комнади що працює над проектом. Ці стандарти підтримують координацію, між певними видами діяльності. Мінімізують складність. Можуть бути пов’язаними з питаннями очікування та обробки змін. І конструювання для перевірки і подальшого тестування. Потрібно визначити загальні правила гри для всіх членів команди, домовившись про терміни процедури, та інші значущі угоди. Не залежно від ступеня формалізації процесів КПЗ зокрема процесів життєвого циклу. Керування конструюванням, моделі конструювань.

Керування конструювань. Моделі конструювань.

Моделі КПЗ визначають комплекс операцій, що включають послідовність, результати. Та інші аспекти пов’язані із загальним життевим циклом розробки. У більшості випадків моделі конструювання визначаються використовуваним стандартом життєвого циклу. Методологіями і практиками. Деякі стандарти ж циклу за своєю природою – орієнтовані на КПЗ. Наприклад XP. Інші розглядають КПЗ у неперервному зв’язку з проектуванням, наприклад RUP. Існує безліч моделей розробки життєвого циклу ПЗ. Деякі з них - більш лінійні з точки зору КПЗ. Наприклад: Водоспадна, поетапна. Ці моделі розглядають КПЗ як діяльність, яка починає проводитись тільки після завершення певних обов’язкових до виконання робіт, що включають : Детальне визначення вимог, детальний дизайн, і детальне планування. Основною сутністю є кодування. Інші моделі – більш ітеративні наприклад: Еволюційна модель, XP,Scrum. Ці підходи сходяться до розгляду КПЗ як діяльності яка ведеться одночасно з іншими видами робіт, зі створення ПЗ. І перетинаючись з ними. Включаючи визначення вимог, планування і проектування. Ці моделі змішують проектування, кодування і тестування. Часто розлядаючи їх комбінацію, як власне – КПЗ.

Моделі ЖЦ, та конструювання.

Каскадна модель.

Досить часто замовник не можу сформулювати докладні вимоги, що до введення. Обробки або висновку данних для майбутнього ПЗ. З іншого боку розробник може сумніватися в сумісності ПЗ та операційної системи у формі діалогу з користувачем. Або у еффективності реалізованого алгоритму. В цих випадках доцільно використовувати макетування. Основна його мета – зняти невизначенності у вимогах замовника. Модель може приймати одну з 3 форм.

1. Паперовий макет, або пакет на основі ПК. (наприклад зображує машино людинний діалог.)
2. Робочий макет.
3. Існуюча программа.

Характеристики і функціонал можуть бути покращені.

Макетування полягає у багаторазовому повторенні ітерацій у яких беруть участь розробник і замовник.

1-> 2->3->1

1. Очікування замовника.
2. Побудова макета.
3. Оцінка макета замовником.

Послідовність дій після макетування.

1->2->3->4->5->6->7->8->9 ->3

1. Макетування
2. Збір та уточнення вимог.
3. Швидке проектування.
4. Побудова макету.
5. Оцінка макета замовником.
6. Уточнення макету.
7. Продовжити? Якщо да, то переходимо до п.3, якщо йдем далі.
8. Конструювання.
9. Вихід

Моделювання забезпечує визначення повних вимог до ПЗ

Недоліки макетування:

Замовник може прийняти макет за продукт.

Розробник може прийняти макет за продукт.

Стратегії КПЗ.

Існує 3 стратегії КПЗ.

1. Одноразовий прохід.
2. Інкриментна стратегія. На початку визначаються всі системні вимоги, а решта КПЗ виконується у вигляді послідовності версій.
3. Еволюційна. Система також є послідовністью версій. Але на початку процесу, визначені не всі вимоги.

Сучасним є XP програмування. – дуже мале збільшення функціональності.

Планування та конструювання.

Вибір методології, моделі, або стратегії констуювання є ключовим аспектом для планування КПЗ. Оскільки наприклад модульне тестування є частиною робіт після написання відповідного функціонального коду. В гнучких тактиках написання юніт тестів пишеься до того, як пишеться відповідний код для тестування.

Планування КПЗ визначає порядок, в якому створюються компоненти, та інші частини КПЗ, проводяться роботи по забезпеченню якості отриманого ПЗ, розподіляються завдання і відповідні ресурси між членами команди. Все це відбувається дотримуючись правил, що визначаються методами, моделями та стратегіями що використовуються. Отже використаний підхід до конструювання має можливості зменшення( мінімізації ) складності, готовності для змін і конструювання з можливістью перевірки. Вимірювання конструювань. Велика частина результатів та й самої діяльності з КПЗ може бути виміряна в тому числі й кількісно. Це насамперед вихідний програмний код, ступінь повторного використання. Модифікований код. І багато інших характеристик. Ці вимірювання (результати аудиту та метрики коду) мають велику користь як для оцінки ризиків, та якості, так для керування КПЗ і керування програмними проектами в цілому.

Якщо ми не взмозі передбачити, ні тривалості, ні вартості окремих завдань, ні імовірність виникнення дефектів, проти заданих параметрів прийнятної якості ПЗ, що розробляють. Код є одним з найбільш чітко детермінованих активів проекту. Поступово такими стають і моделі. Що будуються на основі структур метаданних.

Останнім часом програмісти все більшої уваги приділяють рефакторінгу коду. Як методу реструктурування без зміни функціональності ПЗ з метою мінімізації складності. З метою очікування змін (гнучкість), прозорості до коментування, та багатьох інших чинників КПЗ.

Практичні особливості КПЗ. Проектування в КПЗ.

Одною із значущих сутностей КПЗ є проектування. Проте деякі проектування мають бгато значущих робіт через проекти КПЗ. Мови конструювання включають в себе усі комунікації, за допомогою яких людина може задати вирішення проблеми, що виконуються на ПК. Вони поділяються на такі типи:

1. Конфігураційна – мова, що дозволяє задавати параметри виконання програмної системи.
2. Інструментальна мова - мова КПЗ з повторно використовуваних елементів, зазвичай будується як сценарна мова «Скріпт», що виконується у відповідному середовищі.
3. Мова програмування – це найгнучкіший тип мов конструювання, що містить мінімальний обсяг інформації, про конкретні області додатків, і процес розробки, а вимагає найбільше зусиль на вивчення і напрацювання досвіду для ефективного застосування для вирішення деяких задач.

Існує три основних види нотацій при визначенні мови програмування:

1. Лінгвістична нотація – що характеризується зокрема використанням рядків тексту, що містять спеціалізовані слова, що представляють складні програмні конструкції, і комбіновані в шаблони, шо нагадують речення, обумовленні відповідно до визначеного синтаксису. В разі коректного використання таких нотацій – кожен рядок, що отримується має строге смислове навантаження. Це забезпечує інтуїтивне використання ПЗ, що побудоване з використанням такої мови.
2. Формальна нотація є менш лінгвістична ніж перша, і часто базується на точних формальних (математичних) визначеннях. Формальні нотації у вигляді конструкцій і формальних методів є ядром практично всіх форма системного програмування. Точніше поведінки системи в часі. Такі нотації збільшують якість отриманого коду, до тестування. Що важливіше ніж просто можливість відображення на звичайну людську мову. Формальні конструкції також використовують точний метод визначення комбінацій символів, що використовуються Що дозволяє уникнути не однозначностей властивих конструкціям властивих мов.
3. Візуальна нотація – Припускає безпосередню інтрепритацію візуальних конструкцій в процесі виконання описаної логіки. При цьому в логіці задається розташування відповідних візуальних сутностей відповідальних за ті чи інші операції чи структури данних. Використання візуальних конструкцій обмежене складністю візуально представлення складних виразів і тверджень тільки за рахунок переміщення візуальних сутностей на діаграмі.

Кодування

Практика КПЗ показує активне застосування таких міркувань і технік:

Техніки створення легкого для розуміння вихідного коду. На основі використання угод про іменування, форматування та структурування коду. Використання класів, перелічуваних типу змінних іменованих констант та інших виразних сутностей. Організація вихідного тексту. Вирази шаблони класи пакети модулі та інші структури. Використання структур керування: Запобігання можливим прогалинам у безпеці, наприклад переповнення буферу, або вихід за межі індекса і масивів. Викорстаня ресурсів на основі застосування механізмів виключення( або блокування) і порядку доступу до паралельно використовуваних ресурсів.

Документування коду

Тонке налаштування коду.

Рефакторинг.

Тестування конструювання.

При кпз використовується 2 форми тестування, що проводяться інженерами, що безпосередньо створюють вихідний код. 1 модульне тестування (юніт тест) 2 інтеграційне тестування.

Головна мета тестування в КПЗ – зменшити часовий розрив між моментом появи помилок, наявних кодів і моментом їх виявлення. Окремі види тестів відносяться до галузі знань «Тестування ПЗ»

Системне тестування.

Тестування навантаження.

Оцінка прозорості використання.

Реюз.

Завдання пов’язані з повторним використанням в процесі КПЗ і тестування включають

1. Вибір одиниць. Баз даних тестових процедур. Даних отриманих у результаті тестів і самих тестів які підлягають повторному використанню.
2. Оцінка потенціалу повторного використання коду і тестів.
3. Відстеження інформації і створення звітності що до повторного використання у новому коді. У тестових процедурах і данних отриманих у результаті тестів.
4. Таким чином реалізація повторного використання ПЗ має на увазі і тягне за собою щось більше.
5. Воно вимагає формалізації повторного використання на основі інтеграції процесів і діяльності, що до повторного використання сам життєвий цикл ПЗ.
6. Якість конструювання. Існує ряд техні призначених для забезпечення якості коду, що використовують у міру його конструювання: Покрокове кодування. Використання процедур тверджень. Технічний огляд та оцінки.
7. Інтеграція. (Впровадження.) Одна з ключових діяльностей що здійснюється в процесі КПЗ. Це інтеграція окремо зконструйованих операцій класів компонентів і підсистем, класів компонентів модулів. На додаток деякі програмні системи потребують спеціальної інтеграції з іншим програмним апаратним і програмним забезпеченням. Крім цього сюди відноситься : Планування послідовності в якій інтегру.ться компонентів. Забезпечення підтримки проміжних версій ПЗ. Задання глибини тестування, на основі приктерій прийнятної якості. Та інших робіт що до забезпечення якості інтегрованих в подальшому компонент. Визначення етапних точок проекту, коли будуть тестуватися проміжні версії ПЗ що конструююється.

Проектування експортних систем в КПЗ.

Для створення такої системи потрібно: вивчити процес мислення людини. Яка розвязує певні задачі, або приймає рішення у певній ділянці знань. Виділити основні позиції процесу, і розробити програмні засоби, які б реалізували ці позиції за допомогою ПК.

Інакше кажучи – штучний інтелект передбачає простий структурний підхід до розробки складних програмних систем прийняття рішень. При цьому функціонал залежить від системи знань, якими вона володіє, а не лише від частини схем. Тобто щоб зробити програму інтелектуальною її потрібно наповнити множиною високоякісних спеціальних знань про певну предметну область. Комплекс таких програм кожна з яких є експертом у певній предметній області, називається експортними системами. Технології побудови експертних систем часто називають інженерією знань. Це процес взаємодії автора експертної системи або інженера знань. З одним або кількома експертами у первній проблемній сфері. ІЗ видобуває з експертів стратегії процедури правила які вони використовують під час розвязання задач. І вбудовує отримані знання в експортну систему як показано на малюнку.

1 Предметний експерт –> 2 Інженер знань. => 3 експортна система.

Взаэмодія 1 і 2 - стратегії правила області.

Характеристики експортнгої системи.

Експортна система – це система поєднань можливостей ПК із знаннями і досвідом експертів у такій формі, що вона може запропонувати розумну пораду, або здійснити розумне розв'язання поставленої задачі. Основою ЄС є база знань. Знання у такій базі відображеня так, щоб спростити прийняття рішень. Якщо розглядати знання розвитку, то їх можна поділити на факти та Евристики. Факти – добре відомі в певній області знань обставини. Евристики – грунтуються на власному досвіді експерта. Крім того знання можна поділити на факти і правила (знання для прийняття рішень.) Факти є знаннями типу « Сірко – це пес». А правила знаннями типу «Якщо то».