

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра обчислювальної техніки

Лабораторна робота №4

з навчальної дисципліни

«Технології забезпечення якості програмних продуктів»

## Виконав: студент VI курсу ФІОТ, гр. ІП-41м

## Бляшук Олег

## Київ – 2015

**Тема:** Основи версіонування ПЗ. Розробка доказу концепції (Proof of concept).

**Ціль:** Ознайомитись з процесами розробки програмних продуктів, версіями. Створення прототипу продукту – версія 0.1.

**Теоретичні відомості:**

**Software Development Process** – структура накладена на розробку ПЗ.

Процеси розробки ПЗ складаються з великої кількості етапів. Нижче наведені основні активності, які розташовані в послідовності «водоспадного» процесу (Waterfall process). Послідовність етапів в інших процесах може бути змінена.

**Аналіз вимог (Requirements Analysis)**

Складання вимог, яким повинен відповідати розроблювальний продукт, є першим кроком до створення необхідного ПЗ. Доки замовник, можливо, вважає, що він знає, що повинне робити необхідне ПЗ, може знадобитися знання й досвід у розробці ПЗ, щоб розпізнати недостатність, двозначність і протиріччя у вимогах.

**Специфікації (Specification)**

Ця активність точно описує ПЗ, яке потрібно розробити в строгій математичній формі. У реальності самі вдалі специфікації пишуться на зрозумілі й точно врегульовані додатки, які були успішно розроблені в минулому. Специфікації найбільш важливі для замовників які хочуть переконатися в стабільності майбутнього продукту.

**Програмна архітектура (Software Architecture)**

Архітектура ПЗ є абстрактним представленням розроблювального додатка. Основна мета даного етапу - переконатися що система буде відповідати вимогам, а також майбутні вимоги зможуть бути реалізовані. Ще одною важливою метою на даному етапі є розгляд всіх інтерфейсів між розроблювальним продуктом та іншим ПЗ, а також визначення необхідних апаратних засобів і ОС для успішної розробки.

**Програмування (Coding)**

На даному етапі створюють дизайни (design у цьому випадку - це фаза в розробці ПЗ, на якій функціональні специфікації транслюються в текст програми) і реалізують їх за допомогою певних технологій і мови програмування. Спрощення програмування за допомогою дизайнів є очевидною частиною розробки ПЗ, проте складання дизайнів не повинне бути більшою частиною даного етапу.

**Тестування (Testing)**

Тестування частин системи, особливо в тих місцях де код двох та більше розроблювачів повинен працювати разом.

**Документування (Documentation)**

Важлива частина (якій найчастіше не приділяють значення) - документування внутрішніх дизайнів ПЗ для подальшої підтримки і розширення. Документація - найважливіший етап для замовника.

**Підтримка (Maintenance)**

Підтримка й розширення ПЗ разом зі знайденими проблемами або новими вимогами можуть вимагати набагато більше часу, ніж розробка самого ПЗ. Не тільки додавання коду, що не був написаний відповідно до дизайну, а й визначення, як ПЗ працює з різних сторін після закінчення розробки, може потребувати значних зусиль зі сторони інженерів. Близько 2/3 роботи всіх інженерів - підтримка, проте така статистика може ввести в оману. Лише мала частина підтримки це виправлення помилок. Більша частина - це додавання нових можливостей системи.

Версії: Pre-Alpha, Alpha, Beta, RC, RTM, GA

При швидкому прототипуванні ми створюємо макет, який на певному етапі буде залишений і не стане частиною готової системи.

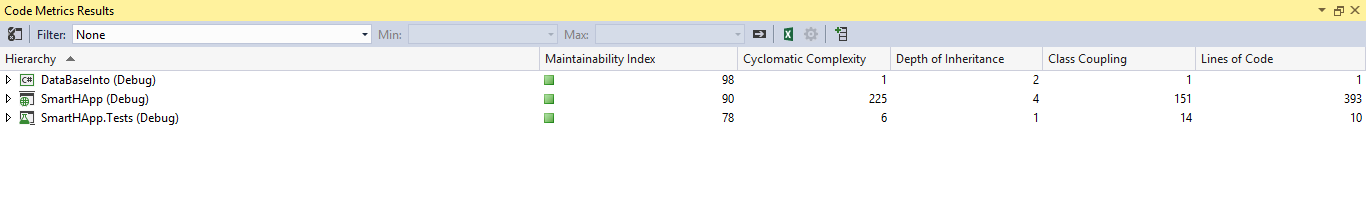
Основні переваги в швидкості створення. У відповідь на вимоги замовник швидко отримує прототип інтерфейсу і одразу може узгодити вимоги, до того як написаний робочий код системи. На цьому етапі дуже низька вартість змін вимог.

Еволюційне прототипування має на меті послідовно створювати макети системи які будуть з кожним кроком ближчі до реального продукту.

Сильною стороною цього підходу є те, що на кожному кроці ми, фактично, маємо робочу систему, хоч і не з повною функціональністю. Цей підхід часто обирається у випадках, коли вимоги відомі не повністю, а уточнюються на кожному наступному етапі.

Оцінка якості коду за допомогою інструменту оцінки. ( як варіант - Sonar Qube)

Оцінку якості було проведено стандартними механізмами Visual Studio:



**Висновок:**Отже версіонування є одним із головних характеристик якісної розробки так як це допомагає розробляти нові продукт-версії таким чином щоб не втратити зроблену роботу, це дозволяє експериментувати і шукати оптимальну відповідність вимогам нашого програмного забезпеченення.