**Практична робота №1-1. Ядро професійних знань SWEBOK як основа проектування програмного забезпечення.**

**Мета:** ознайомитися з основними документом SWEBOK.

**Методичні рекомендації до виконання практичної роботи**

Обрати тему для створення програмного проекту. Над цією темою ми будемо працювати в ході вивчення основ програмної інженерії та на наступному курсі конструювання програмного забезпечення.

**1.** Проаналізувати SWEBOK, визначити базові аспекти та мету їх застосування для створення Вашого власного проекту з області знань «Проектування ПЗ» за такими розділами:

* базові концепції проектування ПЗ (Software Design Basic Concepts)
* ключові питання проектування ПЗ (Key Issue in Software Design)
* структура і архітектура ПЗ (Software Structure and Architecture)
* аналіз і оцінка якості проектування ПЗ (Software Design Quality Analysis and Evaluation),
* нотації проектування ПЗ (Software Design Notations)
* стратегія і методи проектування ПЗ (Software Design Strategies and Methods).

**2.** Які ще положення з інших областей знань SWEBOK Вам будуть потрібні для Вашого проекту? Надайте коротке пояснення.

**3.** Оформити результати у вигляді документу / звіту. Звіт має містити:

номер і назву практичної роботи;

прізвище та номер групи студента;

назву проекту, його мету та визначення предметної області;

два блоки визначень та пояснень (відповідно до пп. 1,2).

*Предметна область* — це частина реального світу, що розглядається в межах певного контексту, тобто середовища, в якому існує об'єкт. Під контекстом можна розуміти область дослідження чи область, яка є об'єктом певної діяльності. Наприклад, для проекту створення сайту поліклініки предметну область можна визначити як медицина або медичні послуги, проекту застосунку оброблення звітності супермаркету - торгівля.

## Результати надсилати на електронну адресу викладача [t.i.lumpova@gmail.com](mailto:t.i.lumpova@gmail.com)

Файл повинен мати назву в такому форматі:

**ОРІ<Номер групи><Номер лекції / практичної / лабораторної [літера позначення типу роботи L – лекція, P – практична, R – лабораторна]<Прізвище англійською>**. Наприклад, **ОРІ3101R**buts.doc.

Не копіюйте фрагментів з різних інформаційних джерел, подумайте і викладіть свою точку зору. При наявності робіт -"близнюків" відповідь буде зараховуватися першому за часом надсилання.

**Строк виконання цієї роботи ІПЗ-31 - 16.09.2024**

**ІПЗ-32 - 16.09.2024**

**ІПЗ-33 - 16.09.2024**

**Контрольні питання і завдання**

1. Назвіть області знань SWEBOK інженерії розробки ПЗ.

2. Приведіть базові поняття SWEBOK.

3. Визначите цілі і завдання області знань – проектування ПЗ.

4. З якими стандартами узгоджуються ядра знань SWEBOK.

5. Який міжнародний стандарт визначає перелік і зміст процесів ЖЦ програмного продукту?

6. Які розділи ядра знань і стандарту найбільш необхідні при розробці програмних систем?

7. В чому різниця між програмним забезпеченням і програмним продуктом?

**Теоретичні відомості.**

За міжнародною концепцією CDIO (Conceive-Design- Implement-Operate) студентам надається освіта, яка формує інженерні основи, викладені в контексті життєвого циклу реальних систем, процесів і продуктів: «Задумай (Ідея)» - «Спроектуй (Проект)» - «Реалізуй (Реалізація)» - «Керуй (Управління, Експлуатація)». Відповідно, для реалізації такого підходу студенти мають на основі індивідуального проекту дослідження/розроблення реалізувати всі етапи життєвого циклу (далі – ЖЦ) проекту, виконуючі різні ролі на певних етапах.

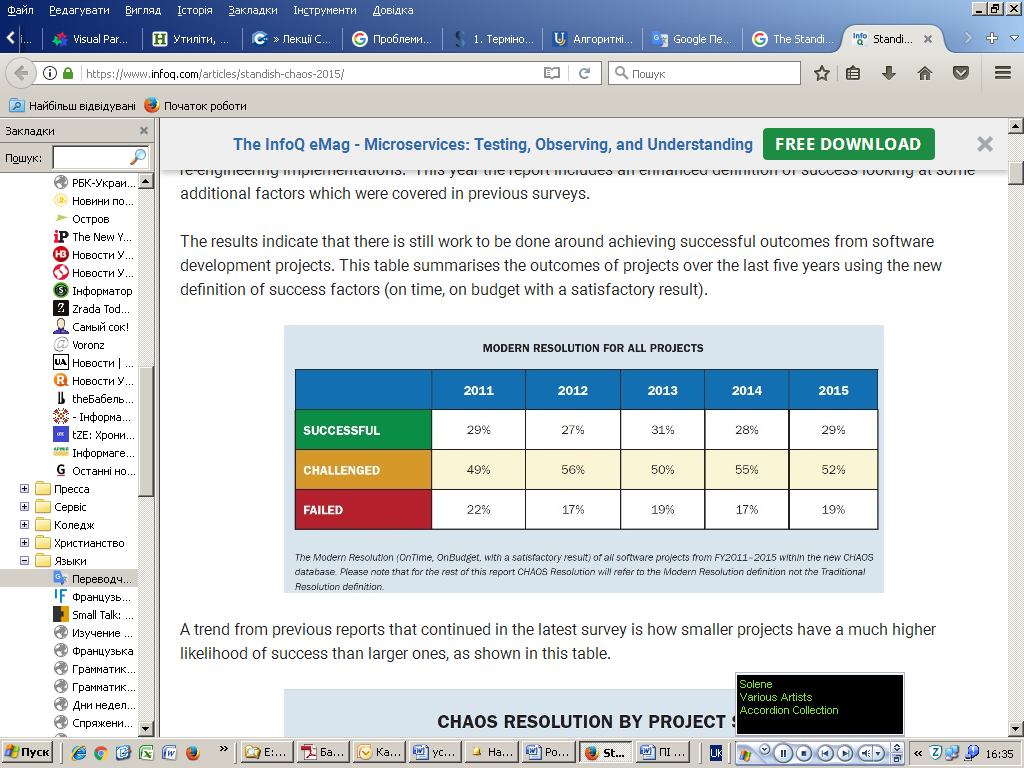
IT-проекти – проекти в області інформаційних технологій, для частини з яких метою яких є розробка програмного забезпечення (далі – ПЗ) або програмного продукту (далі – ПП). Замість словосполучення "програмне забезпечення" часто використовують інше - "програмний продукт". Ці поняття близькі за змістом, але не тотожні. ПЗ, в загальному випадку, є частиною інформаційної системи, наприклад – САПР. До складу таких систем крім програмного, входять і інші види забезпечень – інформаційне, математичне, технічне, методичне, тощо. ПЗ не є завершеним рішенням, це лише одна з частин системи. ПП – це закінчене рішення. Одна з головних властивостей ПП – його здатність бути проданим (і купленим). Можливість продавання ПП – основа бізнесу по розробці ПЗ.

Відповідно до визначення стандарту ISO/IEC 12207 ПЗ (Software) – набір комп'ютерних програм, процедур і пов'язаної з ними документації та даних. Тобто програмний продукт – це програма зі всією супутньою документацією, програма, яку можна продати, або витягувати з нею фінансову вигоду іншим чином. Відповідно, важливу роль грає настанова / керівництво користувача, що надає можливість покупцю ефективно користуватися продуктом. Для успішного просування продукту необхідним (але не достатнім) є виконання таких умов:

1. Продукт повинен виходити на ринок: належної якості; вчасно; бути цікавим потенційним користувачам.
2. Витрати повинні відповідати початковому бюджету.

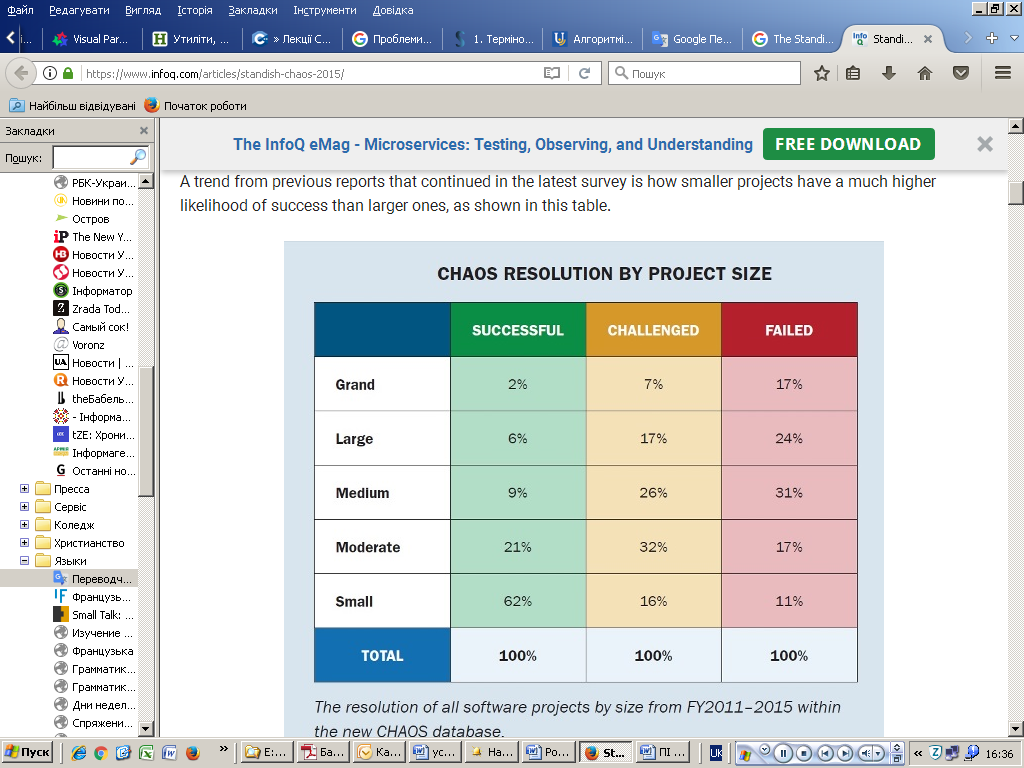
Занадто часто ці вимоги не виконуються. Визначимо ступень успішності проектів:

* Провалені: закінчилися невдачею - мета взагалі не була досягнута.
* Проблемні: закінчилися створенням продукту, але перевищили бюджет або (и) не уклалися в час або (и) мають лише часткову функціональність.
* Успішні: закінчилися створенням продукту, уклалися до бюджету і час. Вся планована функціональність реалізована.



За даними The Standish Group International в звіті Extreme Chaos (https://www.infoq.com/articles/standish-chaos-2015/ ) за період з 2011 по 2015 р.р. в середньому 28% проектів були успішними, серед них великих лише 2%.

Свіжі відомості https://hennyportman.files.wordpress.com/2021/01/project-success-qrc-standish-group-chaos-report-2020.pdf



Наведені діаграми говорять про те, що із зростанням розміру проекту шанси на його успіх катастрофічно падають.

Найбільш істотними причинами проекту невдачі є такі.

1. Нереалістичні тимчасові рамки.

Правильно оцінити час, необхідний для виконання проекту, - складне завдання, колосальне значення в ньому має грамотний облік ризиків.

2. Недолік кількості виконавців. Часто Причина - переоцінювання можливості своїх співробітників, або недооцінювання складності завдання, що з‘ясовується в ході розробки.

3. Розмиті межі проекту. Одна з найбільш серйозних причин невдачі проекту - нечітко сформульовані цілі, неодноразово змінні в ході розробки. Причина – не були чітко сформульовані вимоги до системи відповідно до побажань користувача

4. Недостатність засобів. Відомо дві крайнощі при плануванні бюджету: надмірне роздування (підхід песиміста) і надмірне зменшення (підхід оптиміста). Використання першого підходу найчастіше приводить до втрати проекту (дорожче ніж у конкурентів). Другий підхід часто застосовується або через надмірний оптимізм менеджменту, або в рекламних цілях, щоб за всяку ціну виграти проект. За демпінгові заходи завжди доводиться розплачуватися. Якісно реалізувати проект за виділені гроші виявляється просто неможливим. При провалі проекту репутація розробника може впасти до не відновлюваного рівня. Розумним є оцінювати бюджет реально з деякою перестраховкою на випадок непередбачених ситуацій (захворів ключовий співробітник, вийшло з ладу дороге устаткування...).

5. Недостатність кваліфікованих кадрів. Це - одна з істотних проблем ІТ галузі. Технології розвиваються з такою швидкістю, що професіонали вимушені весь час оновлювати свої знання. Відносна новизна самої області IT, з одного боку, впровадження інформаційних технологій, що стає повсюдним, у всі сфери людської діяльності, з іншою, а, означає, все зростаючий попит на фахівців ведуть до істотного браку кваліфікованих кадрів.

Для того, щоб підвищити об'єми ринку, необхідно підвищити показник успішності проектів, що вимагає від ІТ- галузі не тільки нових технологій, але і значної кількості грамотних фахівцях, здатних ці технології застосовувати. Саме це привело світову комп'ютерну спільноту до необхідності систематизації накопичених знань і фіксації загальні з них у вигляді ядер знань (Body of Knowledge – BOK) для різних областей інформатики. Для створення ядра знань ПЗ був створений міжнародний комітет при американському об'єднанні комп'ютерних фахівців ACM (Association for Computing Machinery) і інституті інженерів по електроніці і електротехніці IEEE Computer Society. До комітету увійшли фахівці світового рівня в області інформатики і розробки ПЗ, які внесли свій досвід і знання, а також систематизували накопичені різнорідні знання і визначили (1999г., 2001г., 2004г.) ядро професійних знань SWEBOK (Software Engineering Body Knowledge) програмної інженерії, як основи проектування ПЗ. Ядро включає суму знань, розподілену по 10 спеціалізованим областям, які відображають окремі процеси проектування ЖЦ ПЗ і методи їх підтримки. Таким чином, документ SWEBOK (Software Engineering Body of Knowledge), підготовлений співтовариством IEEE Computer Society, став одним із базових проектів зі систематизації накопичених знань в області програмної інженерії.

Призначення SWEBOK полягає в об’єднанні знань з розробки ПЗ і їх структуруванні. Він покликаний вирішувати такі завдання при підготовці фахівців у галузі ПЗ: визначити необхідний набір знань і рекомендовані практики; визначити етичні та професійні стандарти; визначити навчальну програму для студентів, аспірантів, що продовжують навчання.

Визначається, що програмного інженерія- це технологія побудови комп’ютерних програмних систем на інженерній основі за методами, засобами та інструментами програмування, сучасними стандартами процесів ЖЦ, менеджменту та керування якістю. Особливістю виробництва нових систем є технологія їх проектування від аналізу предметної області до утворення коду для виконання на комп’ютерах. Проектування у програмній інженерії – це конструювання комп'ютерних систем методами та засобами програмування за такими загальними кроками:

- опис вимог;

- опис специфікацій системи;

- проектування системи;

- конструювання (реалізація) системи;

- тестування, оцінка надійності та якості системи.

Ядро знань SWEBOK і регламентовані процеси стандарту ISO/IEC 12207 узгоджені між собою. Вони утворюють практичний базис інженерії виробництва програмного продукту якзастосування систематичного, дисциплінованого та вимірюваного підходу до розроблення, експлуатації і супроводження ПЗ із застосуванням інженерних методів до розроблення ПЗ.

Інженерію програмного продукту розглядають з двох точок зору:

- як інженерну діяльність, у якій інженери різних категорій виконують роботи в рамках проекту, використовуючи відповідні теоретичні методи і засоби, що рекомендовані у ядрі знань SWEBOK, а також стандарти процесів проектування цільових об’єктів за вибраними методами;

- як систему керування проектом, якістю і ризиками з використанням правил і положень стандартів ЖЦ, якості та менеджменту проекту.

Ядро знань SWEBOK - стислий опис концептуальних основ інженерії програмного продукту. Структурно ділиться на 10 розділів (knowledge areas), які умовно можна розкласти за двома категоріями: проектування продукту й інженерна діяльність.

Створення продукту - це методи і засоби розробки (формування вимог, проектування, конструювання, тестування, супровід).

Інженерна діяльність - методи керування проектом, конфігурацією і якістю та базовим процесом організації-розробника.

Загальне призначення інженерії програмного продукту - практичне виготовлення комп’ютерних програм, систем і інструментів із застосуванням теоретичних та інженерних методів.

Завантажити SWEBOK можна за посиланням

https://oiipdf.com/swebok-v3-0-guide-to-the-software-engineering-body-of-knowledge

Є також переклад, але він із великим нагромадженням коментарів перекладача

https://studfile.net/preview/3652697/page:2/