**Лекція 9. Стандартизація розробки програмного забезпечення**

**Стандартизація розроблення ПЗ**

Стандартизація - це область, в якій постійно збільшується число учасників. Стандарти завжди грали особливу роль в телекомунікаційній галузі, оскільки вони є ключем до функціональної сумісності різних технологій: комп'ютерних систем, засобів зв'язку та передачі сигналів, програмного забезпечення та мультимедійних систем. У цій області основної технологічної проблемою є проблема порозуміння і взаємодії систем, джерелом якої служить використання виробниками таких технологій різного технологічного моделювання та стандартизації. Розробка технологій на основі загальних стандартів дозволить реалізувати принцип взаємної відкритості систем, який є найважливішою характеристикою інформаційного суспільства. Технологічно цей принцип вільного взаємодії та властивості відкритих систем називається *інтероперабельність,* також даний термін визначається і законодавчо. Інтероперабельність являє собою одне з головних властивостей відкритих систем і досягається за рахунок використання узгоджених наборів стандартів.

Розвиток будь-якої галузі економіки обов’язково супроводжується формалізацією використовуваних підходів та появою стандартів різного рівня. На ранніх етапах окремі підприємства формалізують внутрішні процеси, щоб забезпечити повторюваність результатів процесу або створення певного продукту. Для полегшення взаємодії підприємств та зручності споживачів розробляються ***галузеві стандарти.***

Розвиток кожного виду господарської діяльності приводить до потреби у державних засобах забезпечення якості продукції або процесу, тому розробляються та затверджуються ***державні стандарти***.

Для поліпшення умов співробітництва, розроблення загальнозрозумілих правил конкуренції на міжнародному ринку створюються об’єднання галузевих органів стандартизації, результатом діяльності яких є ***регіональні стандарти*** (діють у обмеженому переліку держав, які приєдналися) або ***міжнародні стандарти.***

Одним із перших стандартів, що мав істотний вплив на розвиток теорії проектування та розроблення інформаційних систем (ІС), був стандарт ***BSP (Business System Planning)***. Даний стандарт був розроблений компанією IBM у середині 70-х рр. ХХ ст. Процес BSP передбачав виділення в ході розроблення ІС таких кроків:

* отримання підтримки керівництва,
* визначення процесів підприємства,
* визначення класів даних,
* проведення інтерв’ю,
* обробка та організація результатів інтерв’ю.

Найважливіші кроки процесу BSP спостерігаються у більшості формальних методик.

**Стандарти на розроблення та супровід програмного забезпечення**

**На сьогодні діють такі стандарти, які регламентують процес розроблення ПЗ:**

* ***ГОСТ 34.601-90[[1]](#footnote-1)***– державний стандарт, що поширюється на автоматизовані системи і встановлює стадії та етапи їх створення. У стандарті міститься опис змісту робіт на кожному етапі.
* ***ISO/IEC 12207***. ***Systems and software engineering – Software Life Cycle Processes***– міжнародний стандарт на процеси розроблення та організацію життєвого циклу ПЗ. Поширюється на всі види замовленого ПЗ. Стандарт не містить опису фаз, стадій та етапів.
* ***Guide to the Software Engineering Body of Knowledge (SWEBOK)[[2]](#footnote-2)*** – Керівництво до зведення знань з програмної інженерії – галузевий стандарт Інституту інженерів з радіоелектроніки та електротехніки (IEEE), що систематизує основні види діяльності з програмної інженерії.

**Міжнародні стандарти ISO**

*ISO 12207* Systems and software engineering – Software Life Cycle Processes *- базовий стандарт* процесів життєвого циклу ПЗ, орієнтований на різні види ПО і типи проектів автоматизованих систем, в яких ПЗ є однією з складових частин. Стандарт визначає стратегію і загальний порядок в створенні та експлуатації ПЗ, він охоплює життєвий цикл від концептуалізації ідей до завершення проекту. Доцільність спільного використання стандартів на інформаційні системи і на ПО обумовлюється одним з положень ISO 12207, згідно з яким процеси, використовувані під час життєвого циклу ПЗ, повинні бути сумісні з процесами, використовуваними під час життєвого циклу автоматизованої системи.

Згідно ISO 12207, *система -* це об'єднання одного або декількох процесів, апаратних засобів, програмного забезпечення, обладнання та людей для забезпечення можливості задоволення певних потреб або цілей.

У стандарті ISO 12207 не передбачено будь-яких етапів (фаз або стадій) життєвого циклу інформаційної системи. Даний стандарт визначає лише ряд процесів, кожен процес, дія або завдання ініціюються і виконуються іншим процесом в міру необхідності, причому немає заздалегідь визначених послідовностей (природно, при збереженні логіки зв'язків по вихідним відомостям завдань і т. п.) .

В ISO 12207 усі процеси ЖЦ ПЗ розподілені на три групи

 Рис.1 Процеси ЖЦ ІС відповідно до стандарту ISO 12207

У стандарті ISO 12207 описані п'ять **основних процесів ЖЦ ПЗ**:

* *процес придбання* визначає дії підприємства-покупця, що здобуває інформаційну систему, програмний продукт або службу програмного забезпечення;
* *процес супроводу* визначає дії персоналу, який забезпечує супроводження програмного продукту, тобто управління модифікаціями програмного продукту, підтримку його поточного стану та функціональної придатності; сюди ж відносяться установка програмного виробу на обчислювальній системі і його видалення.
* *процес поставки* визначає дії підприємства-постачальника, яке постачає покупця системою, програмним продуктом або службою програмного забезпечення;
* *процес розробки* визначає дії підприємства-розробника, яке розробляє принцип побудови програмного виробу та програмний продукт;
* *процес функціонування* визначає дії підприємства-оператора, яке забезпечує обслуговування системи в цілому (а не тільки програмного забезпечення) в процесі її функціонування в інтересах користувачів. На відміну від дій, які визначаються розробником в. Інструкціях з експлуатації, визначаються дії оператора з консультування користувачів, отриманню зворотного зв'язку та ін., які він планує сам і бере на себе відповідні обов'язки;

    Крім основних, стандарт ISO 12207 обумовлює 8 допоміжних процесів, які є невід'ємною частиною всього життєвого циклу програмного виробу та забезпечують належну якість проекту програмного забезпечення.

***Допоміжні процеси*** призначені для підтримки виконання основних процесів, забезпечення якості проекту, організації верифікації та тестуванняПЗ. Це процеси: вирішення проблем; документування; управління конфігурацією; забезпечення якості; верифікації; атестації; спільної оцінки; аудиту.

***Організаційні процеси*** визначають дії та завдання замовників та розробників для керування процесами у ході проекту.Це процеси: управління; створення інфраструктури; удосконалення; навчання. Під процесом удосконалення розуміється не удосконалення інформаційної системи або програмного забезпечення, а поліпшення самих процесів придбання, розробки, забезпечення якості і т. д., реально здійснюваних в організації.  Процес адаптації визначає основні дії, необхідні для адаптації цього стандарту до умов конкретного проекту.

**Особливості стандарту** ISO **12207**

* Стандарт ISO 12207 має динамічний характер, обумовлений способом визначення послідовності виконання процесів і завдань, при якому один процес при необхідності викликає інший або його частину. Такий характер дозволяє реалізувати будь-яку модель життєвого циклу. Відповідно до стандарту ISO 12207, модель життєвого циклу - це структура, яка містить процеси, дії і завдання, які здійснюються в ході розробки, функціонування та супроводження програмного продукту протягом усього життя системи, від визначення вимог до завершення її використання.
* Стандарт ISO 12207 забезпечує максимальний ступінь адаптивності. Безліч процесів і задач сконструйовано так, що можлива їх адаптація у відповідності з конкретними проектами інформаційних систем. Ця адаптація зводиться до виключення процесів, видів діяльності і завдань, які не застосовані в конкретному проекті.  Згідно ISO 12207, додавання унікальних або специфічних процесів, дій і завдань має бути обумовлено в контракті між сторонами. Причому «контракт» розуміється в самому широкому сенсі - від юридично оформленого документа до неформального угоди. Ця угода може бути визначене навіть єдиною стороною - як завдання, поставлене самому собі.
* Стандарт принципово не містить опису конкретних методів дій, а тим більше - заготовок рішень або документації. Він лише описує архітектуру процесів ЖЦ ПЗ , але не конкретизує в деталях, як реалізовувати або виконувати послуги і завдання, включені в процеси. Даний стандарт не вказує імена, формати або точний зміст одержуваної документації. Рішення такого типу приймаються сторонами, що використовують стандарт.
* Забезпечення якості різними процесами виконується з різною передбаченої ступенем організаційної незалежності контролюючої діяльності аж до обов'язкових вимог до повної незалежності перевіряти персоналу від будь-якої прямої відповідальності. Щодо перевірки об'єктів, то контроль цього виду передбачений на самих ранніх кроках розробки, починаючи з аналізу системних вимог шляхом їх перевірок на відповідність потребам придбання.
* Ступінь обов'язковості розглянутого стандарту така: після рішення організації про застосування ISO 12207 в якості умови торгових відносин вступає її відповідальність за зазначення мінімального набору необхідних процесів і завдань, які забезпечують узгодженість з цим стандартом.
* Стандарт містить гранично мало описів, спрямованих на проектування бази даних, що виправдано тим, що різні системи і різні прикладні комплекси програмного забезпечення можуть не тільки використовувати вельми специфічні типи баз даних, але і взагалі не використовувати базу даних.

***Цінність стандарту ISO 12207***полягаєу тому, що він містить набори завдань, характеристик якості, критеріїв оцінки і т. п., що дають всебічне охоплення проектних ситуацій. Наприклад, при виконанні аналізу вимог до системи передбачається, що:

* розглядається область застосування системи для визначення вимог, пропонованих до системи;
* специфікація вимог системи повинна описувати функції і можливості системи, області застосування системи, організаційні вимоги і вимоги користувача, безпека, захищеність, людські фактори, ергономіку, зв'язку, операції та вимоги супроводу; проектні обмеження та кваліфікаційні вимоги.

    Далі, при виконанні аналізу вимог до програмного забезпечення передбачено 11 класів характеристик якості, які використовуються пізніше при забезпеченні якості. При цьому розробник повинен встановити і документувати у вигляді вимог до програмного забезпечення наступні специфікації і характеристики:

* функціональні та можливі специфікації, включаючи виконання, фізичні характеристики та умови середовища експлуатації, при яких одиниця програмного забезпечення повинна бути виконана;
* зовнішні зв'язки (інтерфейси) з одиницею програмного забезпечення;
* вимоги кваліфікації;
* специфікації надійності, включаючи специфікації, пов'язані з методами функціонування та супроводу, впливу навколишнього середовища та ймовірністю травми персоналу;
* специфікації захищеності, включаючи специфікації, пов'язані з компрометацією точності інформації;
* людські фактори специфікацій з інженерної психології (ергономіці), включаючи пов'язані з ручним керуванням, взаємодією людини і устаткування, обмеженнями на персонал та областями, потребуючими в концентрованому людському уваги, які є чутливими до помилок людини і навчанню;
* визначення даних і вимог до бази даних;
* установочні та приймальні вимоги поставляється програмного продукту в місцях функціонування та супроводу (експлуатації);
* документацію користувача;
* робота користувача і вимоги виконання;
* вимоги сервісу користувача.

     Відповідно до стандарту IS012207, вимога кваліфікації - це набір критеріїв або умов (кваліфікаційні вимоги), які повинні бути задоволені для того, щоб кваліфікувати програмний продукт як такий, що задовольняє умовам та його специфікаціям і готовий для використання в навколишньому середовищі.

     Хоча стандарт не вказує конкретної моделі життєвого циклу або методу розробки, він визначає, що сторони-учасники при використанні стандарту відповідальні за наступне:

* вибір моделі життєвого циклу для розроблювального проекту;
* адаптацію процесів і задач стандарту до цієї моделі;
* вибір та застосування методів розробки програмного забезпечення;
* виконання дій і завдань, придатних для проекту програмного забезпечення.

В стандарті ISO міститься 12207 найбільш широкий набір процесів, дій і завдань, що охоплює більшість можливих ситуацій при максимальній адаптованості. Він містить мінімум обмежень і конкретних рекомендацій. При використанні ISO 12207 детальні визначення процесів, форм документів тощо доцільно виносити в різні функціональні стандарти, відомчі нормативні документи або фірмові методики, які можуть бути використані або не використані в кожному конкретному проекті.

Для підтримки практичного використання стандарту ISO 12207 розроблені такі технологічні документи: Керівництво для ISO/IEC 12207 (ISO/IEC TR 24748-3:2011 Systems and software engineering - Life cycle management - Part 3: Guide to the application of ISO/IEC 12207 (Software life cycle processes)) та Керівництво з використання ISO/IEC 12207 в керуванні проектами (ISO/IEC TR 16326:2009 Systems and software engineering - Life cycle processes - Project management).

У 2002 р. був опублікований стандарт на процеси життєвого циклу систем ISO/IEC 15288 Systems and software engineering - System life cycle processes, у розробленні якого брали участь фахівці різних галузей: системної інженерії, програмування, управління якістю, людськими ресурсами, безпекою та ін. Даний документ враховує практичний досвід створення систем в урядових, комерційних, військових та академічних організаціях і може бути застосований для широкого класу систем, але його основне призначення – підтримка створення комп'ютеризованих систем. На цей час діє версія стандарту 2008 р. У стандарті ISO/IEC 15288:2008 у структурі ЖЦ виділені групи процесів за видами діяльності (рис. 2).



Рисунок 2 – Процеси ЖЦ систем відповідно до стандарту ISO/IEC 15288

Стандарти ISO/IEC 12207 та ISO/IEC 15288 мають єдину термінологію і розроблені таким чином, щоб могли використовуватись одночасно у проекті.

У процесі промислового розроблення ПЗ обов’язково використовуються стандарти якості серії ISO 9000. Серія ISO 9000 (управління якістю) містить у собі такі стандарти:

* ISO 9000-1. Керування якістю і гарантії якості. Частина 1. Посібник з вибору й використання.
* ISO 9000-2. Керування якістю й гарантії якості. Частина 2. Загальний посібник із застосування стандартів ISO 9001, ISO 9002 і ISO 9003.
* ISO 9000-3. Керування якістю й гарантії якості. Частина 3. Посібник із застосування стандарту ISO 9001 при розробленні, установці й супроводі ПЗ.
* ISO 9000-4. Керування якістю й гарантії якості. Частина 4. Посібник з керування надійністю програм.
* Основний стандарт ISO 9001:2009 задає модель системи якості для процесів проектування, розроблення, виробництва, установки й обслуговування (продукту, системи, послуги).

**Стандарти організації IEEE**

У 1963 р. у результаті злиття Інституту радіотехніків (Institute of Radio Engineers, IRE) і Американського інституту інженерів-електриків (American Institute of Electrical Engineers, AIEE) була створена міжнародна некомерційна асоціація технічних фахівців, світовий лідер у галузі розроблення стандартів з радіоелектроніки та електротехніки Інститут інженерів з радіоелектроніки та електротехніки IEEE (Institute of Electrical and Electronics Engineers). Дана міжнародна організація об’єднує понад 400 тис. фахівців із 170 країн. IEEE здійснює інформаційну і матеріальну підтримку фахівців для організації та розвитку наукової діяльності в електротехніці, електроніці, комп'ютерній техніці та інформатиці.

Розробка програмного забезпечення завжди досить складна процедура. Спростити цей процес допомогає міжнародний стандарт на системи управління якістю ISO 9001. Настанови щодо застосування ISO 9001:2008 при розробці програмного забезпечення надані в стандарті I**SO/IEC/IEEE 90003 «Розробка програмного забезпечення. Настанови щодо застосування ISO 9001:2008 при розробці програмних продуктів».** Стандарт призначений для розробки, поставки та обслуговування програмного забезпечення. Стандарт був розроблений спільно з Міжнародною електротехнічною комісією (IEC) та Інститутом інженерів електротехніки та електроніки (IEEE

**Керівництво до зведення знань із програмної інженерії (SWEBOK, 2004)** містить опис 10 галузей знань**:**

* Software requirements – програмні вимоги;
* Software design – дизайн (архітектура);
* Software construction – конструювання ПЗ;
* Software testing – тестування;
* Software maintenance – експлуатація (підтримка) ПЗ;
* Software configuration management – управління конфігураціями;
* Software engineering management – управління у програмній інженерії;
* Software engineering process – процеси програмної інженерії;
* Software engineering tools and methods – інструменти та методи;
* Software quality – якість ПЗ.

Для кожної галузі SWEBOK містить опис ключових елементів у вигляді підобластей (subareas). Для кожної підобласті наведена декомпозиція у вигляді списку тем (topics) із їх описом.

**Стандарт зрілості компанії-розробника ПЗ CMM**

Говорячи про стандартизацію процесів підприємства потрібно розглянути модель зрілості технологічних процесів організації Capability Maturity Model (CMM), розроблену Інститутом інженерів ПЗ (Software Engineering Institute, SEI) та корпорацією Mitre під керівництвом Уоттса Хамфри (Watts Humphrey).

Методологія CMM розроблялася й розвивалася в США як засіб, що дозволяє вибирати кращих виробників ПЗ для виконання держзамовлень у першу чергу міністерства оборони. Для цього були розроблені критерії оцінки зрілості ключових процесів компанії та визначений набір дій, необхідних для їхнього подальшого вдосконалювання. У підсумку методологія виявилася надзвичайно корисною для більшості компаній, що прагнуть якісно поліпшити існуючі процеси проектування, розроблення, тестування програмних засобів і звести керування ними до зрозумілих і легко реалізованих алгоритмів і технологій, описаних у єдиному стандарті.

У подальшому ця модель переросла у методологію підвищення якості процесів підприємства Capability Maturity Model Integration (CMMI). Застосування CMMI дозволяє поставити розроблення ПЗ на промислову основу, підвищити керованість ключових процесів і виробничу культуру в цілому, гарантувати якісну роботу й виконання проектів у строк.

Основою для створення CMM стало базове положення про те, що фундаментальна проблема "кризи" процесу розроблення якісного ПЗ полягає не у відсутності нових методів і засобів розроблення, а в нездатності компанії організувати технологічні процеси й керувати ними.

Для оцінки ступеня готовності підприємства розробляти якісний програмний продукт СММ використовує ключове поняття зрілість організації (Maturity).

***Незрілою*** вважається організація, у якій:

* відсутнє довгострокове й проектне планування;
* процес розроблення ПЗ і його ключові моменти не ідентифіковані, реалізація процесу залежить від поточних умов, конкретних менеджерів і виконавців;
* методи і процедури не стандартизовані й не документовані;
* результат не визначений реальними критеріями, які встановлюються за запланованими показниками із застосуванням стандартних технологій і розроблених метрик;
* процес вироблення рішення відбувається стихійно, на грані мистецтва.

У цьому випадку велика ймовірність появи несподіваних проблем, перевищення бюджету або невиконання строків здачі проекту. У такій компанії, як правило, менеджери й розробники не керують процесами - вони змушені займатися поточними та спонтанними проблемами.

**Основні ознаки *зрілої* організації:**

* у компанії є чітко визначені й документовані процедури керування вимогами, планування проектної діяльності, керування конфігурацією, створення й тестування програмних продуктів, відпрацьовані механізми керування проектами;
* ці процедури постійно уточнюються й удосконалюються;
* оцінки часу, складності й вартості робіт ґрунтуються на накопиченому досвіді, розроблених метриках і кількісних показниках, що робить їх достатньо точними;
* актуалізовано зовнішні й створені внутрішні стандарти на ключові процеси й процедури;
* існують обов'язкові для всіх правила оформлення методологічної програмної й користувальницької документації;

**•** технології незначно змінюються від проекту до проекту на основі стабільних і перевірених підходів і методик;

* максимально використовуються напрацьовані у попередніх проектах організаційний і виробничий досвід, програмні модулі, бібліотеки програмних засобів;
* активно апробуються і впроваджуються нові технології, виробляється оцінка їхньої ефективності.

СММ визначає п'ять рівнів технологічної зрілості компанії (рис. 3), за якими замовники можуть оцінювати потенційних претендентів на підпис контракту, а розробники – удосконалювати процеси створення ПЗ.

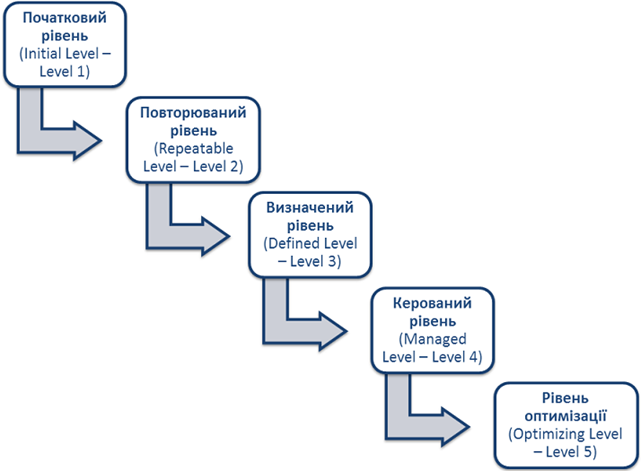


Рисунок 3. Рівні зрілості компанії за СММ

Кожний із рівнів, крім першого, складається з декількох ключових областей процесу (Key Process Area), що містять цілі (Goal), зобов'язання щодо виконання (Commitment to Perform), можливість виконання (Ability to Perform), виконувані дії (Activity Performed), їхній вимір і аналіз (Measurement and Analysis) та перевірку впровадження (Verifying Implementation). Таким чином, СММ фактично є комплексом вимог до ключових параметрів ефективного стандартного процесу розроблення ПЗ та засобом його постійного поліпшення. Виконання цих вимог збільшує ймовірність досягнення підприємством поставлених цілей у сфері якості.

СММ визначає такий мінімальний набір вимог: реалізувати 18 ключових областей процесу розроблення ПЗ, що містять 52 цілі, 28 зобов'язань компанії, 70 можливостей виконання (гарантій компанії) і 150 ключових практик.

У результаті аудиту та атестації компанії присвоюється певний рівень, що після наступних аудитів може підвищуватися або знижуватися. Кожний наступний рівень в обов'язковому порядку містить у собі всі ключові характеристики попередніх. У зв'язку з цим сертифікація компанії щодо одного з рівнів припускає безумовне виконання всіх вимог більш низьких рівнів.

До ***переваг*** ***моделі CMM*** належить те, що вона орієнтована на організації, які займаються розробленням програмного забезпечення. У даній моделі вдалося більш детально визначити вимоги, специфічні для процесів, пов'язаних з розробленням ПЗ. Із цієї причини в CMM наведені не тільки вимоги до процесів організації, але й приклади реалізації таких вимог.

***Основний недолік CMM*** полягає в тому, що модель не авторизована як стандарт ні міжнародними, ні національними органами зі стандартизації. Втім, CMM давно стала промисловим стандартом. До недоліків моделі також необхідно віднести більші зовнішні накладні витрати на приведення процесів компанії у відповідність до моделі СММ, ніж при використанні моделей Міжнародного стандарту ISO 9000.

Більшість західних стандартів є не обов’язковими, проте "добровільність" стандартів в розвинених країнах має не той зміст, який вкладають наші розробники. У західному розумінні "добровільність" розуміється як необхідність і навіть обов'язок постачальника добровільно покладати на себе відповідальність за виконання і навіть перевищення вимог, викладених у добровільних національних або галузевих стандартах на продукцію, процес або послугу. Кожен учасник цивілізованого ринку знає, що без виконання вимог чинних добровільних стандартів, розроблених при безпосередньому добровільну участь постачальників продукції або послуги, неможливі не тільки успішна діяльність, але й саме існування організації. За логікою законодавства, добровільно приймаючи стандарт, суб'єкт певної діяльності тим самим демонструє споживачеві безпеку свого виробництва, що дає йому певну перевагу в конкуренції.

У той же час, дотримуючись добровільності стандартизації в галузі апаратно-програмного комплексу, можна припустити, що на ринку з'являться і існують технології, що містять помилки, за допомогою яких можливо отримати інформацію третім особам. Відповідно в разі застосування програмного забезпечення і відсутності стандартів з програмного забезпечення, виникає питання про відповідальність за шкоду, заподіяну третім особам, причому умова про працездатність програми повинно бути істотною умовою договору.

Звідси випливає, що для того, щоб забезпечити нормальний обіг програмного забезпечення, необхідно підтвердити, що розроблене програмне забезпечення пройшов якийсь тестування. У той же час успішне проходження тестування ще не означає, що в програмі немає помилок, у тому числі перешкоджають її нормальному використанню.

1. ***Контрольні запитання*.**
2. Чим на Вашу думку викликана потреба в стандартизації ПЗ?
3. Які базові стандарти на розробку ПЗ Ви знаєте?
4. Як розподілені процеси ЖЦ ПЗ в стандарті ISO 12207?
5. В чому особливості стандарту ISO 12207?
6. В чому цінність стандарту ISO 12207**?**
7. Як визначається відповідно до стандарту IS0 12207 вимога кваліфікації?
8. Що поєднує стандарти ISO/IEC 12207 та ISO/IEC 15288?
9. Яке призначення стандартів серії ISO 9000?
10. Як на Вашу думку впливає стандарт СММ на якість програмного забезпечення?

***Рекомендована література***

1. Лавріщева К.М. Програмна інженерія. Електронний підручник: http://csc.knu.ua/uk/library/books/lavrishcheva-6.pdf
2. Лавріщева К.М. Програмна інженерія. – Підручник.–К.:Академперіодика, 2008.–415с.
3. И. Соммервиль. Инженерия программного обеспечения,  
    6 изд. – И.д. "Вильямс", 2002.
4. Проектування інформаційних систем: Посібник // За редакцією Пономаренка В.С. – К.: Видавничий центр "Академія". 2002. ­ 488 с. URL: <http://www.dut.edu.ua/uploads/l_874_10304054.pdf>.
5. ДСТУ ISO/IEC 15288:2005. Інформаційні технології. Процеси життєвого циклу системи (ISO/IEC 15288:2002, IDT) - К.: Держстандарт України, 2005. - К.: Держстандарт України, 2004.
6. ДСТУ ISO/IEC/IEEE 12207:2018 Інформаційні технології. Процеси життєвого циклу програмного забезпечення - К.: Держстандарт України, 2018.

***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.

1. ГОСТ 34.601-90 Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания. – М., 1991. [↑](#footnote-ref-1)
2. <https://studfile.net/preview/3652697/> неофіційний переклад російськлю [↑](#footnote-ref-2)