
Управління проектами

ДЕНЬ 2



Програма та програмна система

Система та програмна система

Система є набором взаємодіючих або взаємозалежних компонентів, що утворюють єдине ціле або набором елементів (часто їх називають «компоненти») і відношень, які відрізняються від відношень самого набору або його елементів до інших елементів або наборів

Прикладами систем є, зокрема, автомобіль, автоматична пральна машина, мобільний телефон, комп'ютер тощо

Програмна система (software-intensive system) - це будь-яка система, в якій програмне забезпечення має суттєвий вплив на проектування, будівництво, впровадження і розвиток системи в цілому

Програма vs програмна система. Для кого створюється

Програма створюється для особистого використання або для використання знайомими.

Програмна система завжди створюється для замовника

Програма vs програмна система. Ким створюється

Програма створюється особисто **програмістом** або, можливо, за участю друзів чи знайомих

Програмна система обов'язково створюється **командою розробників**.

Програма vs програмна система. Як створюється

Створення програми, передбачає постановку задачі, розробку алгоритму, програмування та налагодження, використання за призначенням.

Розробка програмної системи вимагає інших видів діяльності.

Оскільки програмна система розробляється для замовника, то обов'язково присутні **вимоги** замовника до створюваної системи

Для створення системи необхідно здійснити її **проекткування**.

Створення програмної системи, звичайно, неможливе без розробки або **програмування**

Для того, щоб зменшити кількість наявних помилок та зробити програмні системи більш надійними, проводять **тестування**

Після того, як система побудована, її треба передати замовнику. Ця діяльність називається **впровадженням**

Діяльність розробників з підтримки створеної системи називається **супроводом**

Програма vs програмна система. Як створюється. 2

У розробці програмної системи важливо також врахувати й інші аспекти.

Це, зокрема, стосується **даних**.

Так само, для програмної системи важливою є наявність **документів**

Порівняння

	Програма	Програмна система
Для кого створюється		
	для себе або знайомих	для замовника
Ким створюється		
	особисто програмістом (можливо, за участю друзів та знайомих)	колективом розробників
Як створюється		
	постановка задачі	вимоги та їх аналіз
		проектування
	розробка алгоритму та програми, налагодження	програмування
		тестування
		впровадження
		супровід
		дані
		документи

Стандарти та методології

Стандарт - нормативний документ, заснований на консенсусі, прийнятий визнаним органом, що встановлює для загального і неодноразового використання правила, настанови або характеристики щодо діяльності чи її результатів, та спрямований на досягнення оптимального ступеня впорядкованості в певній сфері

- "Як має бути?"

Методологія - система певних правил, принципів і операцій, що застосовуються у тій чи іншій сфері діяльності

"Які кроки належить зробити, щоб було так, як треба?"

Стандарт SWEBOOK

SWEBOOK - Software Engineering Body of Knowledge, або Зведення Знань з Програмної Інженерії

SWEBOOK описує 10 областей знань:

- Software requirements – програмні вимоги
- Software design – проектування програмного забезпечення (дизайн, архітектура)
- Software construction – конструювання програмного забезпечення
- Software testing - тестування програмного забезпечення
- Software maintenance – супровід (підтримка) програмного забезпечення
- Software configuration management – управління конфігураціями
- Software engineering management – управління в програмній інженерії
- Software engineering process – процеси програмної інженерії
- Software engineering tools and methods – інструменти та методи
- Software quality – якість програмного забезпечення

Вимоги та їх аналіз

Вимоги

Вимога - це умова чи можливість (capability) якій повинна відповідати система

Вимоги прийнято поділяти на функціональні та нефункціональні



КАК ЗАКАЗЧИК ОБЪЯСНИЛ,
ЧТО ЕМУ НУЖНО



КАК МЕНЕДЖЕР ПРОЕКТА
ЭТО ПОНЯЛ



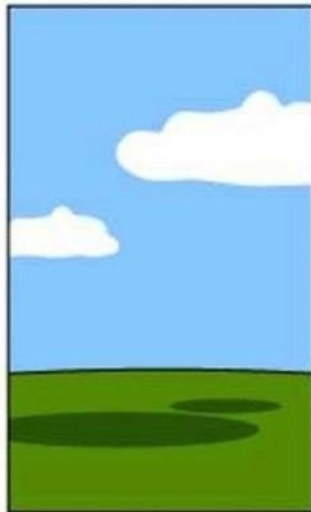
КАК ЭТО
СПРОЕКТИРОВАЛИ



КАК ПРОГРАММИСТ
ЭТО НАПИСАЛ



КАК ОТДЕЛ ПРОДАЖ
ЭТО ПРЕЗЕНТОВАЛ



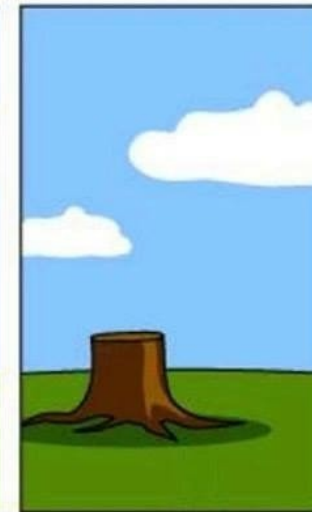
КАК ВЫГЛЯДИТ
ДОКУМЕНТАЦИЯ



ИНСТРУКЦИЯ ПО
УСТАНОВКЕ



ЗА ЧТО ЗАПЛАТИЛ
КЛИЕНТ



КАК ПРОХОДИТ
ПОДДЕРЖКА ПРОДУКТА



ЧТО ЗАКАЗЧИК
ДЕЙСТВИТЕЛЬНО ХОТЕЛ

Функціональні вимоги

Функціональні вимоги регламентують функціонування або поведінку системи (behavioral requirements).

Функціональні вимоги відповідають на запитання «що повинна робити система» в тих чи інших ситуаціях.

Функціональні вимоги визначають основний «фронт робіт» Розробника, і встановлюють цілі, завдання та сервіси, що надаються системою Замовнику.

Функціональні вимоги записуються, як правило, за допомогою розпорядчих правил: ***«система повинна дозволяти комірникові формувати прибуткові та видаткові накладні»***.

Нефункціональні вимоги

Нефункціональні вимоги, регламентують внутрішні і зовнішні умови або атрибути функціонування системи

Виділяють наступні основні групи нефункціональних вимог:

- Зовнішні інтерфейси (External Interfaces),
- Атрибути якості (Quality Attributes),
- Обмеження (Constraints).

Зовнішні інтерфейси

Зовнішні інтерфейси – це взаємозв'язки системи з зовнішнім світом.

Не варто зводити зовнішні інтерфейси тільки до інтерфейсу користувача.

До них також включають зв'язки із зовнішніми системами. Наприклад:

«Інтерфейс користувача Системи повинен бути витриманий у корпоративних кольорах Замовника.»

або

«Система бухгалтерського обліку повинна отримувати дані про співробітників з кадрової системи. Дані отримують шляхом імпорту з періодом один раз на тиждень.»

Атрибути якості

Атрибути якості встановлюють пріоритети у розробці системи з точки зору якісних характеристик

Обмеження

Обмеження вказують на умови, які накладають на систему.

Наприклад:

«Загальна кількість користувачів системи - до 50 000.

Кількість користувачів, які працюють з системою одночасно, - до 3 000.»

або

***«Система дозволяє будувати тільки стандартні звіти, перелік яких, наведено у ...
Система не будує нестандартні звіти.»***

Основні атрибути якості

Застосовуваність (Usability)

Надійність (Reliability)

Продуктивність (Performance)

Підтримуваність (Supportability).

Основні атрибути якості.2

Застосовуваність (Usability – часто дають просто трансліт юзабіліті) – це зручність системи до застосування. Прикладом таких вимог може бути:

«Будь-яка функція системи повинна бути доступною за допомогою не більше, ніж трьох кліків мишею»

Основні атрибути якості.3

Надійність – це здатність системи працювати безвідмовно певний проміжок часу.

Надійність вказують у годинах безвідмовної роботи, або, частіше, у відсотках безвідмовної роботи по відношенню до відмов.

Наприклад «три дев'ятки» - 99,9%. Або «п'ять дев'яток» - 99,999%.

Розраховують надійність, виходячи з графіку роботи системи.

- Якщо система повинна працювати тільки у робочий час (8x5, або 8 годин на тиждень 5 днів тижня), то відмова системи у неробочий час не враховується. Якщо ж система повинна працювати цілодобово (24x7), то враховуються всі відмови. Не вважаються також відмовами планові зупинки системи для її обслуговування. Для розрахунку надійності ділять загальний допустимий час відмов (на рік) на загальний час роботи системи та віднімають результат від 100%. Наприклад, при цілодобовій роботі системи надійність - 99,999% означає, що загальний час непрацездатності системи не повинен перевищувати приблизно 40 хвилин на рік.