І РІВЕНЬ

**1.**



**2.** Валідація - це процес перевірки продукту з точки зору кінцевого користувача.

Верифікація - це процес перевірки продукту на відповідність вимогам замовника.

ІІ РІВЕНЬ

**2.**

| **Тип компанії** | **Плюси** | **Мінуси** |
| --- | --- | --- |
| *Product companies* | 1. Стабільная робота 2. Вітається творчій підхід до роботи 3. Можливість відчувати себе частиною продукту, отже більш значущим. | 1. У початківців майже немає шансів 2. Залежність від успіху продукту на ринку 3. Можлива однотипність, якщо компанія тривалий час займається розвитком одного й того ж продукту. |
| *Outsource companies* | 1. Найкращий варіант для початківців 2. Постійний різновид проектів, продуктів. 3. Розвиток Hard skills, так як можна сьогодні працювати над веб.сайтом, завтра над грою, післязавтра над моб.додатком 4. Будь-які організаційні питання з Замовником вирішуватиме компанія, отже звичайному працівнику менший клопіт. | 1. Якість продукту може бути не на першому місці, головне швидкість 2. Будь-які рамки та вимоги встановлюються Замовником, а не компанією. 3. При відсутності заказів - відсутній й прибуток. Виходить певна залежність від замовлень. 4. Зачасту є додатковий тиск на співробітників. |

**3.** Одного разу донька захотіла самостійно зшити гамак-тунель для свого домашнього улюбленця - пацючка. Вона довго обирала матеріал, щоб був м’ягенький, вимірювала розміри тканини, робила часті стібки, щоб шов був надійним, обрала найліпше кріплення. По всім параметрам, що були задані “Продукт” пройшов *верифікацію*. Але улюбленець-пацючок не схотів у ньому відпочивати. На практиці виявилось, що йому було замало місця всередині. Можна зробити висновок, що *валідацію* “Продукт” не пройшов.

ІІІ РІВЕНЬ

**2.** Одним із семи принципів тестування є ***кластерізація дефектів.*** Важливість цього принципу полягає в тому, що якщо ми бачимо в одному модулі багато багів нам обов’язково треба копати глибше. Так ми знайдемо більше багів, детальніше проаналізуємо ризикований модуль.

Ще одним принципом є ***раннє тестування***. Ще його називають “зсувом вліво”. Мається на увазі зсув вліво у схемі циклу розробки ПЗ, найкраще на етапі *вимог*. Дотримання цього принципу значно скоротить час на тестування та не менш важливі кошти.

Моїм досвідом є взаємодія з недостатньо протестованим продуктом. А саме з Хмарним медичним інформаційним сервісом Health 24.

Мною як медичним працівником, а не тестувальником було знайдено десятки багів і передано їх через службу support команді розробників. Наприклад цілий модуль, що стосувався вводу інформації щодо проведених щеплень дитині містив в собі величезну кількість багів. Сюди чудово відноситься принцип кластерізації дефектів. А щодо раннього тестування то я можу тільки уявити які фінансові втрати мала компанія, що розробила цей продукт, бо усі баги мною знайдені були після релізу.