Порівняльна таблиця найбільш поширених методологій

| Назва методології | Сильні сторони | Слабкі сторони | Для якої галузі є доцільним |
| --- | --- | --- | --- |
| Waterfall | Структурований процес, так як є докладне документування кожного етапу  Дисциплінує, так як є чіткий план і послідовність етапів розробки та наявний строгий менеджмент  Чітке планування строків та витрат  Прозорість, що дозволяє прогнозувати бюджет та набирати команду | Відсутність гнучкості. Готове рішення замовник отримує тільки в кінці проєкту. В разі зміни чогось вся робота починається спочатку, що тягне за собой великі витрати  Тестування на останньому етапі. Це недолік, так як немає можливості повернутись на попередній етап і скоріше за все дефекти усуваються швидко і неякісно | Державні установи, так як є суворо складена звітність, графік та неможливість відхилятись від заданого спочатку плану |
| V-model | Сувора етапізація, так як розвиток і прогрес організовані і систематичні  Тестування починається з самого початку, тому неоднозначності виявляються з самого початку. | Не передбачено проведення аналізу ризиків, тому існує невизначеність та ризики  Не передбачає роботу паралельними подіями, через це ПЗ може стати застарілим і нікому не потрібним |  |
| Інкрементна та ітеративні моделі | Ця модель є більш гнучкою - менш затратною для зміни обсягу та вимог.  Її легше тестувати та налагоджувати протягом меншої ітерації.  У цій моделі клієнт може реагувати на кожну збірку.  Легше управляти ризиками, оскільки ризиковані частини визначаються та обробляються під час ітерації. | Потребує хорошого планування та дизайну.  Потребує чіткого і повного визначення всієї системи, перш ніж її можна буде розбити на частини і будувати поетапно. | Для розробки соціальних мереж, так як є зрозумілий загальний функціонал, але ще є можливість додавати різні функції і удосконалювати |
| Спіральна модель | Великий обсяг аналізу ризиків, а отже, покращується уникнення ризиків.  Суворий контроль затвердження та документації.  Додаткова функціональність може бути додана пізніше.  Програмне забезпечення створюється на ранній стадії життєвого циклу програмного забезпечення. | Використання моделі може бути дорогим.  Аналіз ризиків вимагає вузькоспеціалізованих знань.  Успіх проекту значною мірою залежить від етапу аналізу ризиків.  Не підходить для невеликих проектів. | Дослідження та військові програми |
| Agile | Задоволення потреб замовників шляхом швидкого та безперервного постачання корисного програмного забезпечення.  Акцент робиться на людях та взаємодії, а не на процесах та інструментах.  Постійна увага до технічної досконалості та гарного дизайну.  Регулярна адаптація до мінливих обставин.  Пізні зміни у вимогах не становлять перешкод | У випадку деяких програмних продуктів, особливо великих, важко оцінити зусилля, необхідні на початку життєвого циклу розробки програмного забезпечення.  Недостатньо уваги приділяється необхідному проектуванню та документації.  Немає розуміння кінцевого результату  Методологія чутлива до рівня команди проекту | Стартап, так як в цій розробці не завжди розумієш, що хочеш отримати в кінцевому значенні і можеш вносити зміни |

Чому з’явився Agile-маніфест? Які проблеми він мав вирішити та чи вдалося?

Довгий час ринок ІТ користувався класичними методологіями, такими як Waterfall і тощо. В класичних моделях є пріоритет послідовність, відсутність можливості повернутись на попередній етап розробки. Також, класичні методи дуже неповороткі і погано піддаються швидким змінам, що не є плюсом у швидкоплинному світі.

Agile-маніфест - радикально новий підхід для управління проектами. Він базується на таких фундаментальних принципах, як:

1. Люди та взаємодія важливіші, ніж процеси та інструменти.
2. Працююче ПЗ переважає над вичерпною документацією.
3. Співпраця з клієнтом важливіша, ніж узгодження контракту.
4. Реагування на зміни важливіше, ніж дотримання плану.

Ці принципи допомагають замовнику бути почутим командою розробників, змога вносити зміни. Також такий метод існує для швидкої розробки і швидкого вихода на ринок для отримання прибутку. Взаємодія між відділами команди сприяє більш швидкому реагуванню на зміни, на нові вимоги замовника, що зменшує витрати.

Гнучка методологія для сучасного світу має дуже багато сильних сторін, як для клієнта, так і для розробників, але все ж таки залишаються такі види проектів, де “жорстка” методологія досі актуальна. Наприклад, медицинська сфера, де якість результату набагато важливіша, ніж швидкість його отримання.

Можна вважати, що основні проблеми не гнучких моделей - це неможливість до змін. І так, Agile вирішив цю проблему, так як ця методологія є повністю прозорою і замовник присутній від самого початку і до кінця розробки і може вносити зміни. Чи може Agile повністю замінити всі методології розробки? Я вважаю, що ні. Для великих промислових підприємств, державних установ повинно бути присутнє чітке визначення інструментарію та функціоналу.

Завдання: Ти – засновник/ця стартапу і плануєш випустити на ринок мобільний застосунок для обміну світлинами котиків. Яку методологію ти обереш для процесу розробки і чому? Відповідь текстово обґрунтуй.

Я би обрала гнучку методологію Agile методом Scrum. За рахунок коротких ітерацій я зможу отримати готовий застосунок в доволі швидкий термін з можливістю змін і додавання нововведень.