

## Beet Seed

1. Склади порівняльну таблицю найбільш поширених методологій:

No.	Назва методології	Сильні сторони	Слабкі сторони	Для якої галузі є доцільною
1	Scrum	1. Відкритість до змін 2. Спонукає до тісної взаємодії між учасниками команди 3. Великий акцент на прозорість роботи	1. Потребує часу щоб члени команди зрозуміли методологію 2. Для великих проектів в чистому - не дуже підходить (тоді вже в ход йде Nexus) 3. Досить популярний - і іноді клієнти хочуть принципово роботу по Скраму, навіть коли він не підходить	Стартапи, різні діджитал бізнеси. Не обов'язково навіть розробка
2	Waterfall	1. Добре підходить якщо проект жорстко зарегульований 2. Є достатньо часу щоб пропрацювати дизайн рішення, та підготувати усі спеки 3. При розробці - менше часу уходить на уточнення та питання, тк немає змін	1. Зміни - дуже болюче явище 2. Потребує багато часу щоб надати готовий результат (та випустити на ринок) 3. Зворотній зв'язок як результат теж буде вже після того як рішення готове	Зарегульовані галузі - наприклад авіація, військові проекти, космчні, ітд

## Beet Sprout

1. Напиши розгорнуті відповіді (0,5 - 1 сторінки тексту) на такі два питання + 2. Які проблеми він мав вирішити і чи це вдалося?

До Agile майже все робилося по Waterfall-y. І доволі частим явищем були: пропущені дедлайни. Або над рішенням працювали роками, але воно так і не було доведено до робочого стану (улюблений приклад Scrum майстерів - Expeditionary Combat Support System).

Причин цього було чимало, я б виділив основним те - що звичаний план дій тоді був, все детально обдумати, задокументувати - а потім довго розробляти поки рішення не буде готове.

Тут виникають основні проблеми:

1. Зворотній зв'язок виникає тільки під кінець. Тобто може виявитися що в результаті рішення дуже відрізняється від того що очікувалося (зі сторони клієнтів)
2. Другий момент - фідбек від юзерів. Тобто він теж з'являється коли рішення вийшло на ринок. І з цим може багато проблем. Можливо у рішення поганий UX і воно взагалі не зрозуміле. Вірогідність стикнутися з багами у користувацьких також більша ніж з Agile. Ну і останнє, юзери можуть на ділі використовувати рішення/фічі не так як планувалося при плануванні. (традиційна історія багатьох технологій та винаходів, той же інтернет, чи мікрохвильовки)

3. Зміни. В бізнесі це стандартне явище, може бути як і ринком, так і державою, або просто бажанням клієнтів. Але за waterfall-м виникає ряд проблем. Треба переробляти документацію, оновляти розроблений функціонал. І в ряді ситуацій легше почати з нуля - що дуже неефективно.

Чи вирішив він ці проблеми? Загалом так, зараз підхід та процеси краще ніж в 90-х. Але нічого в світі не ідеально - і Agile породив свою низку проблем. Але то вже окрема історія

## Mighty Beet

1. 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 б скоріше обрав Kanban, або Scrumban.

Причини:

1. Стартап нішовий. Треба зрозуміти чи зайде ідея на ринку.
2. Я б почав з MVP, щоб зрозуміти в залежності від реакцію користувачів та ринку - що треба додати/оновити/прибрати.

Для цього Kanban дасть достатньо гнучкості порівнянно з Waterfall.

Можна було б обрати Scrum, але:

1. Буду відвертий, я його не дуже люблю, тому що сам не стикався з нормальним скрамом на практиці (Scrumban only).
2. У Скрама серед деяких практикантів - підхід та ідеалогія на рівні з релігією - і мені це теж не до вподоби. Методологія має просто працювати. А не бути об'єктом фетішу.