

# Manual QA

Тестування в Agile

# Manual QA

## План уроку

- Agile цінності в реальному житті
- Вплив тестувальника на якість
- Активності до початку спринту
- Активності під час спринту
- Активності після спринту

Що було минулого разу

# Manual QA

## План уроку

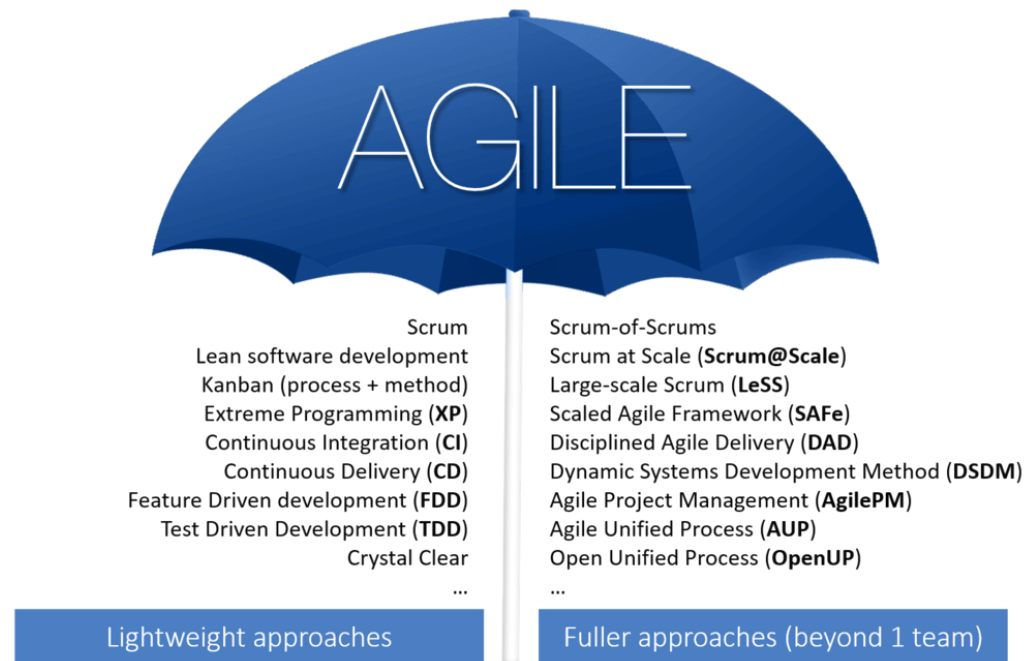
- **Jira workflow (робочий процес у Jira)** - це набір статусів і переходів, через які проходить задача протягом свого життєвого циклу, і зазвичай представляє собою процес у вашій організації.
- Статус задачі вказує на її поточне місце у робочому процесі проєкту. Статуси за замовчуванням створюються під час створення проєкту на основі одного зі стандартних шаблонів.
- У Jira є певна ієрархія задач, яка допомагає структурувати та зв'язати елементи між собою відповідно до зв'язків, взаємодій та приналежності тієї чи іншої задачі.
- Володіючи роллю Адміністратора Jira проєкту, ми можемо управляти та змінювати процес переходів у робочих процесах Jira і таким чином контролювати наступні аспекти поведінки переходу задачі з одного стану до іншого.
- **Zephyr** - це плагін в Jira, котрий дозволяє користувачам тестувати прямо всередині Jira. Коли Zephyr використовується з Jira, можна створити тест-кейс, переглянути в будь-якому проєкті Jira і виконати негайно або як частину циклу тестування, який може бути пов'язаний з іншими проблемами.

## Тестування в Agile

# Manual QA

## Що таке Agile?

Agile - серія підходів до розробки програмного забезпечення, орієнтованих на використання ітеративної розробки



# Manual QA

## Agile manifest (Принципи Agile)

- Люди та взаємодія важливіші за процеси та інструменти
- Працюючий продукт важливіший за вичерпну документацію
- Співпраця із замовником важливіша за погодження умов контракту
- Готовність до змін важливіша за слідування початковому плану

## Agile values в реальності

### Agile цінність

#### Люди та взаємодія

важливіші за процеси та інструменти

### Що значить

- Невелика суміщена команда, РО знаходиться разом із командою
- Крос-функціональна команда
- Команда, уповноважена приймати рішення

### Що відбувається в реальності

- Команда розподілена, різниця у часі може становити 10 годин
- РО з боку клієнта та важкодоступний для команди
- У команду входять розробники та тестувальники, які працюють на повну ставку, але дизайнери, розробники, автоматизація входять до інших команд.



## Agile values в реальності

### Agile цінність

Працюючий продукт  
важливіший за вичерпну документацію

### Що значить

- Готовий (тобто протестований) продукт наприкінці спринту

### Що відбувається в реальності

- Розробники пишуть код до останнього моменту спринту
- Тестування не може бути завершено у спринті
- Деякі типи тестування виконуються поза спринтами (регресійне, інтеграційне тощо).

## Agile values в реальності

### Agile цінність

#### Співпраця із замовником

важливіша за погодження умов контракту

### Що значить

- Команда визначає, що вона зобов'язується виконати наприкінці спринту.
- Вимоги змінюються, але терміни фіксовані

### Що відбувається в реальності

- Замовник нав'язує команді обсяг та терміни
- І змінює вимоги у рамках спринту
- І висуває не зовсім визначені вимоги

# Manual QA

## Agile values в реальності

### Agile цінність

#### Готовність до змін

важливіша за слідування початковому плану

#### Що значить

- Ретроспектива спринту – навчайтесь та вдосконалюйтесь, як стати ефективнішим
- Вітайте мінливі вимоги, навіть на пізніх стадіях розробки

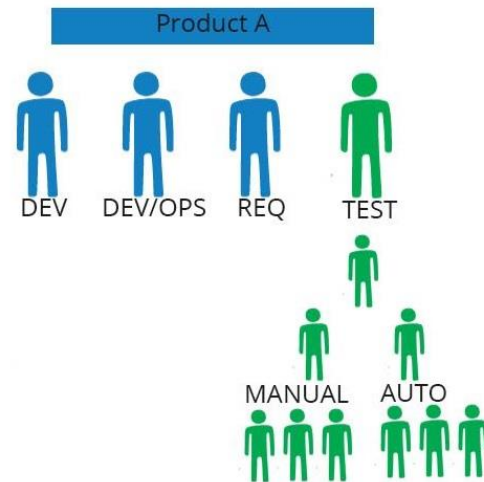
#### Що відбувається в реальності

- Команда не наважується відкрито говорити про проблеми
- Занадто багато тиску на термінах розробки, щоб встигнути на ретроспективи!
- Ми маємо ретроспективи, але нічого не змінюється!
- Вимоги постійно змінюються, тому що вони не готові до початку розробки

# Manual QA

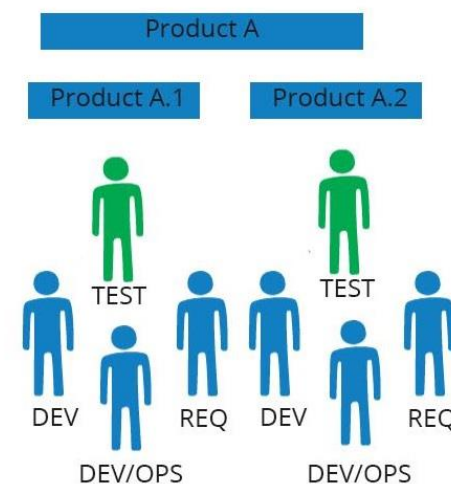
## Роль тестувальника в Agile

### Waterfall Team Org



Тестувальники **знаходять** відмінності між існуючими та необхідними умовами

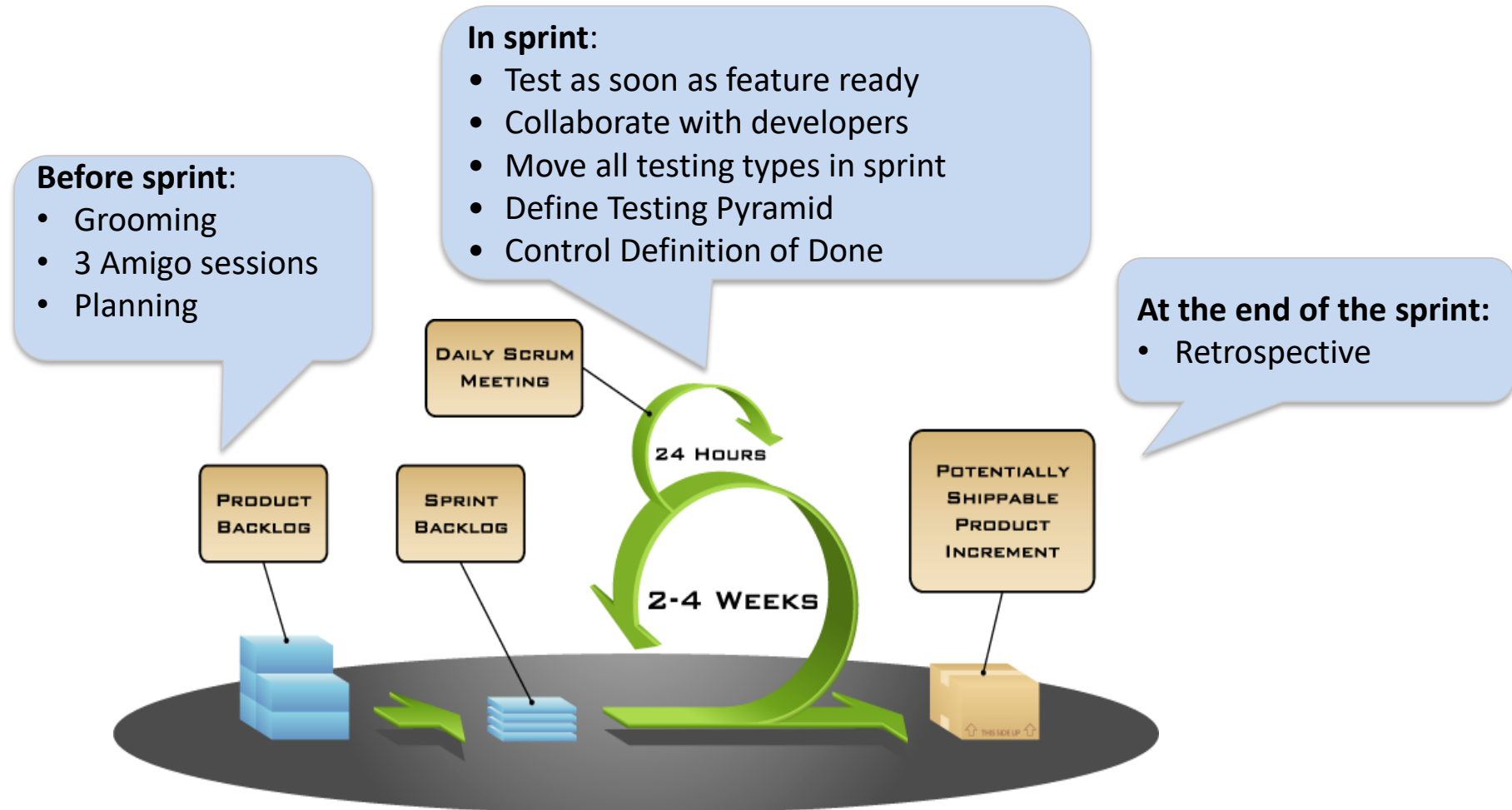
### Agile Team Org



Тестувальники **запобігають** ~~знаходять~~ відмінностям між існуючими та необхідними умовами

# Manual QA

## Де тестувальники можуть вплинути на якість?



## Активності до початку спринту

# Manual QA

## Grooming / Sprint Refinement

Коли	Перед спринтом, навіть за 1 тиждень до
Вхідні дані	Користувацькі історії, створені та описані РО
Хто бере участь	Вся команда, розробка та тестування, РО
Що роблять тестувальники перед grooming нарадою	<p>Аналізуйте користувацькі історії з точки зору:</p> <ul style="list-style-type: none"><li>• Яка інформація відсутня та завадить нам провести тестування?</li><li>• Що дивно/нелогічно/суперечить іншим вимогам?</li><li>• Я знаю, як це перевірити?</li><li>• Чого не вистачає для перевірки? (наприклад, дані)</li></ul>
На grooming нараді	<p>Ставте питання з користувацьких історій</p> <p>Уточнюйте відповіді РО</p> <p>Заплануйте, якщо необхідно, додаткові обговорення</p>

# Manual QA

## 3 Amigos session

Визначте 3 Amigos від кожної команди, щоб обговорити користувацькі історії:

- **Бізнес-аналітик** чи **РО** - Яку проблему ми намагаємося вирішити?
- **Розробник** – Як ми можемо створити рішення для вирішення цієї проблеми?
- **Тестер** - А як щодо цього, що може статися?

Коли це працює краще:

- Сеанси попереднього грумінгу
- Обговорення високорівневих вимог
- Коли можливості команди дуже обмежені і немає можливості запросити всю команду



# Manual QA

## Що може піти не так?

### Що може статися?

- Недостатньо історій існує/деталізовані в беклозі до грумінгу
  - РО не знає, яка мета завдання або деталі
  - РО не готовий відповісти на питання
  - РО не приходить на грумінг
- 
- Команда працює з кількома РО, і вони суперечать один одному
  - РО пізніше змінює думку
  - Вимоги змінюються на льоту

### Що можна зробити?

- Плануйте грумінг заздалегідь
  - Підготовка та обмін питаннями з РО перед сесією
  - Наполягайте на відхиленні незрозумілих історій на плануванні
- 
- Зберігайте відповіді РО у загальному джерелі
  - Записуйте відповіді
  - Вимірювати та інформувати про вплив змін

# Manual QA

## Planning

Коли	1-й день спринту
Вхідні дані	Користувацькі історії з уточненими вимогами (після грумінгу)
Хто бере участь	Вся команда, розробка та тестування, РО
Що роблять тестувальники	<ul style="list-style-type: none"><li>• Оцінюють тестування для кожної історії</li><li>• Обговорюють, чи не здається тестування неспіврозмірним розробці/занадто складним</li><li>• Можуть пропонувати історії з технічного боргу</li><li>• Можуть створювати задачі</li></ul>
Вихідні дані	Sprint backlog: оцінені історії, які команда зобов'язується закінчити до кінця спринту

# Manual QA

## Що може піти не так?

### Що може статися?

- Команда не може оцінити історію
- Клієнт наполягає на нижчих оцінках
- Члени команди мають великі відмінності між оцінками

### Що можна зробити?

- Переконайтеся, що історія відповідає критеріям INVEST
- Ще раз уточніть вимоги
- Розбийте історію на дрібніші шматочки
- Проконсультуйтеся з архітектором або будь-якими іншими експертами
- Обговоріть, що включено до оцінок членів команди
- Попросіть пояснити оцінки
- Надайте деталі оцінювання
- Поясніть ризики зниження оцінок – не буде жодних історій для здачі в кінці спринту, тому що команда не зможе виконати всі завдання, ризики якості – буде багато дефектів

## Активності під час спринту

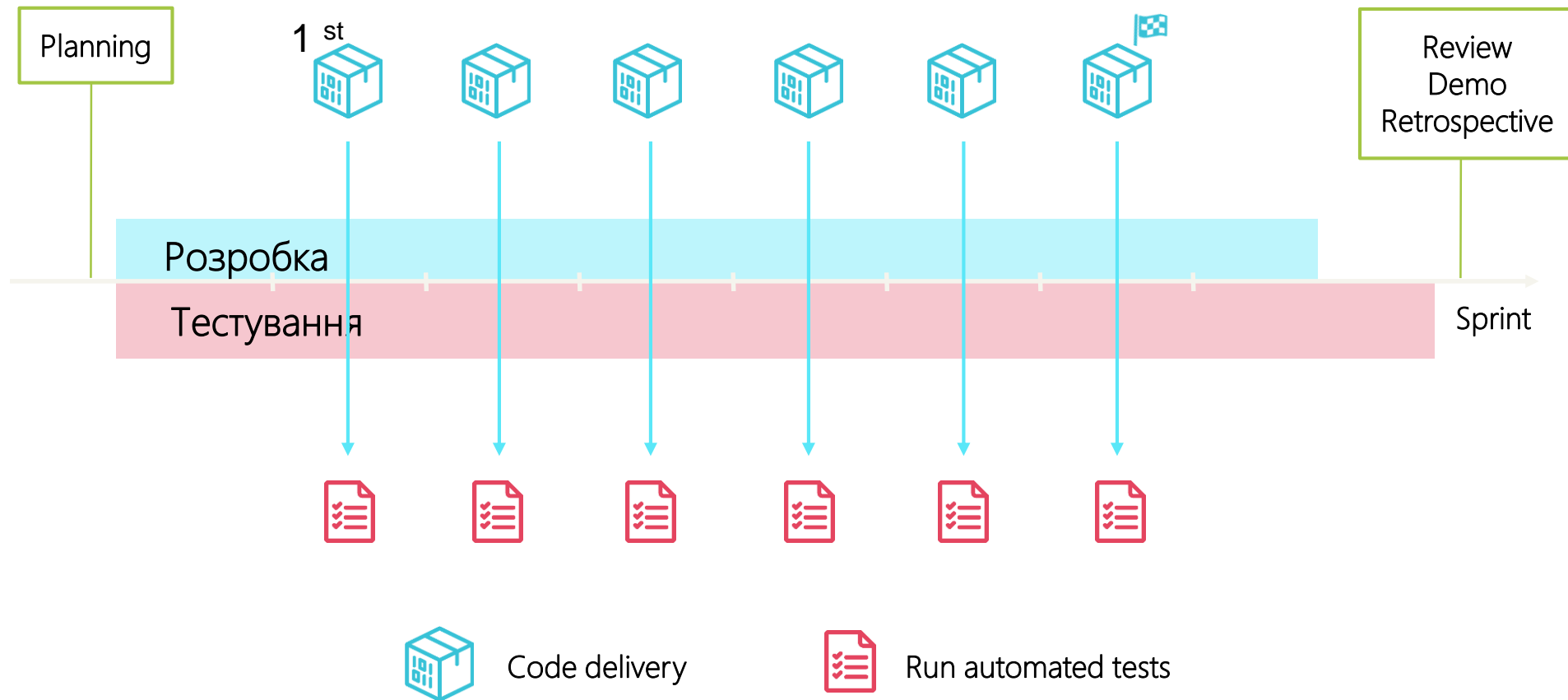
Мета тестувальника – надавати моментальний зворотний зв'язок розробникам

## Як можна виявити дефекти раніше чи запобігти їм?

- Отримайте чіткі вимоги після grooming / 3 amigos сесії
- Спільна робота з розробниками:
  - Поясніть, що ви збираєтеся тестувати, покажіть свої чек-аркуші / тест-кейси
  - Уточніть разом будь-які питання, що стосуються користувацьких історій
  - Наполягайте на використанні кращих методик розробки: модульні тести, код ревью, стандарти кодування і т. д.
  - Наполягайте на налаштуванні безперервного розгортання для тестування - не витрачайте час на очікування нових збірок
- Не відкладайте «спеціальний» тип тестування до кінця релізу

# Manual QA

## Ідеальний таймлайн тестування у спринті

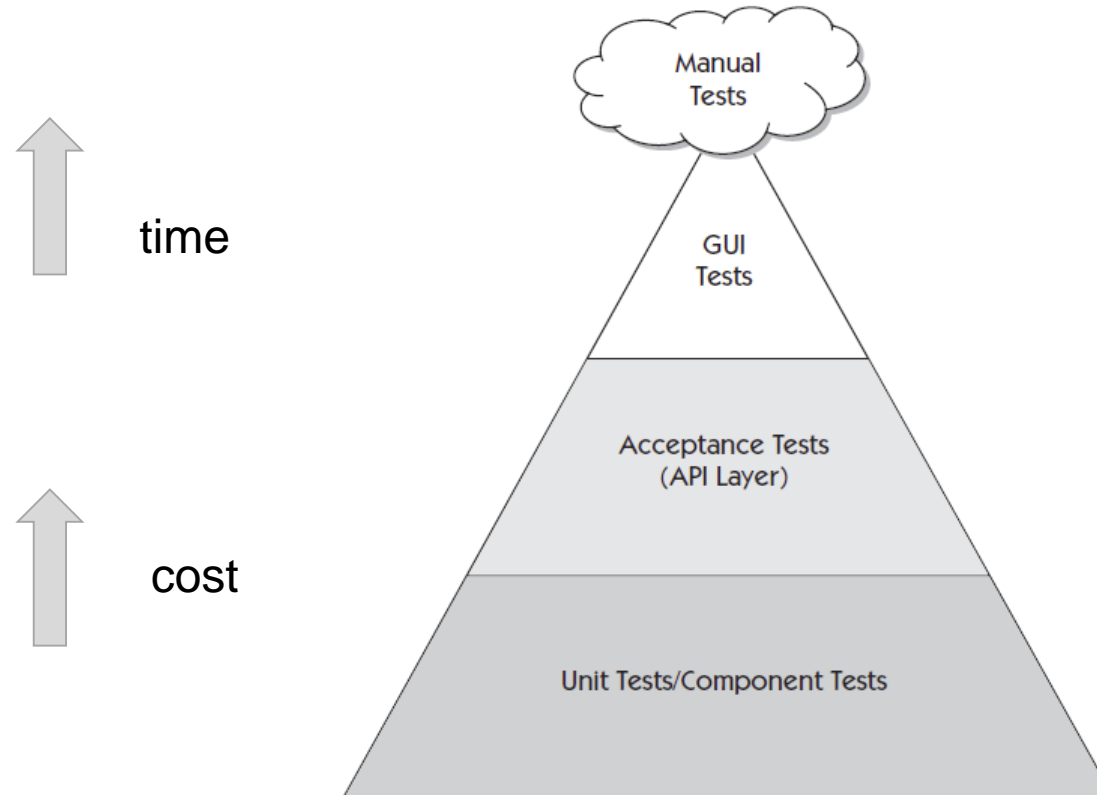


## Щоб побудувати ідеальний таймлайн, треба

- Визначення піраміди тестування
- Інтеграція автоматичних тестів у конвеєр CI/CD
- Виконуйте часто:
  - Модульні тести - після кожного коміту
  - Димові тести - після розгортання збірки (деплою)
  - Регресійні тести - вночі (на вимогу або за розкладом)

# Manual QA

Щоб побудувати ідеальний таймлайн, треба





## Що може статися під час спринту?

### Що може статися?

- Тестове середовище не працює
- Немає тестових даних
- Залежності від інших команд/систем

### Що можна зробити?

- Якщо це відбувається часто, налаштуйте резервне тестове середовище
- Якщо це викликано проблемами з розгортанням збірки => CI/CD, автоматизація тестів, модульних тестів
- Обговоріть це з головним розробником/менеджером. Загальний час, який ми втрачаємо, є найсильнішим аргументом
- Як ми можемо отримати якомога більш «справжні» дані? (реплікація, анонімізація, генерація)
- Розробіть макети і заглушки (mocks and stubs)
- Узгоджуйте терміни з іншими командами

## Що може статися під час спринту?

### Що може статися?

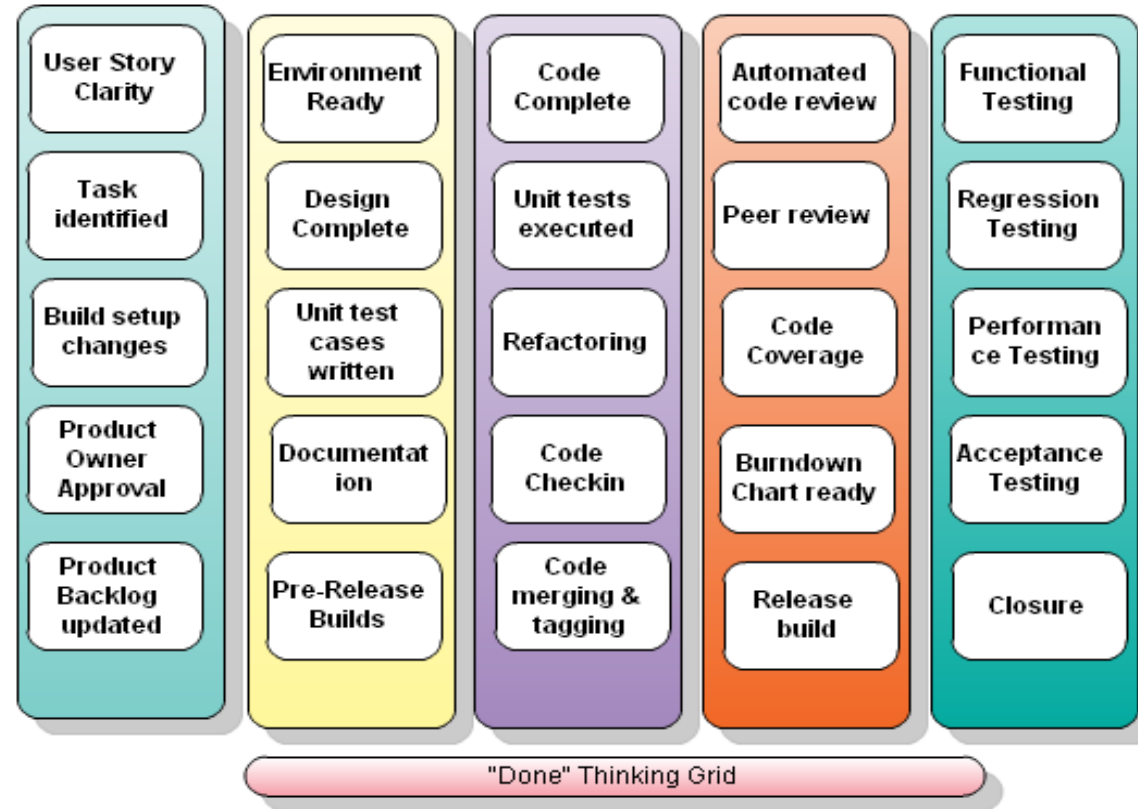
- Розробники займаються розробкою до найостаннішого дня
- Багато дефектів у функціоналі

### Що можна зробити?

- Домовтеся про день, коли перші розроблені історії будуть доставлені на тестування
- Підштовхуйте розробників до плавного внесення змін
- Домовтеся про кількість днів або годин до закінчення спринту, коли всі історії будуть доставлені
- Скорочення capacity команди на кількість ненабраних story points
- Модульні тести, авто тести на CI/CD, які виконуються перед розгортанням збірки
- Локальне тестування
- Визначте основну причину помилки: проблеми з навколишнім середовищем, проблеми з вимогами, проблеми з кодом (модульні тести проходять?)

# Manual QA

## Definition of Done



## Активності після спринту

## Ретроспектива. Що й Навіщо?

Ретроспектива - це зустріч, де команда обговорює, що може бути змінено , що може зробити наступний спринт більш цінним.

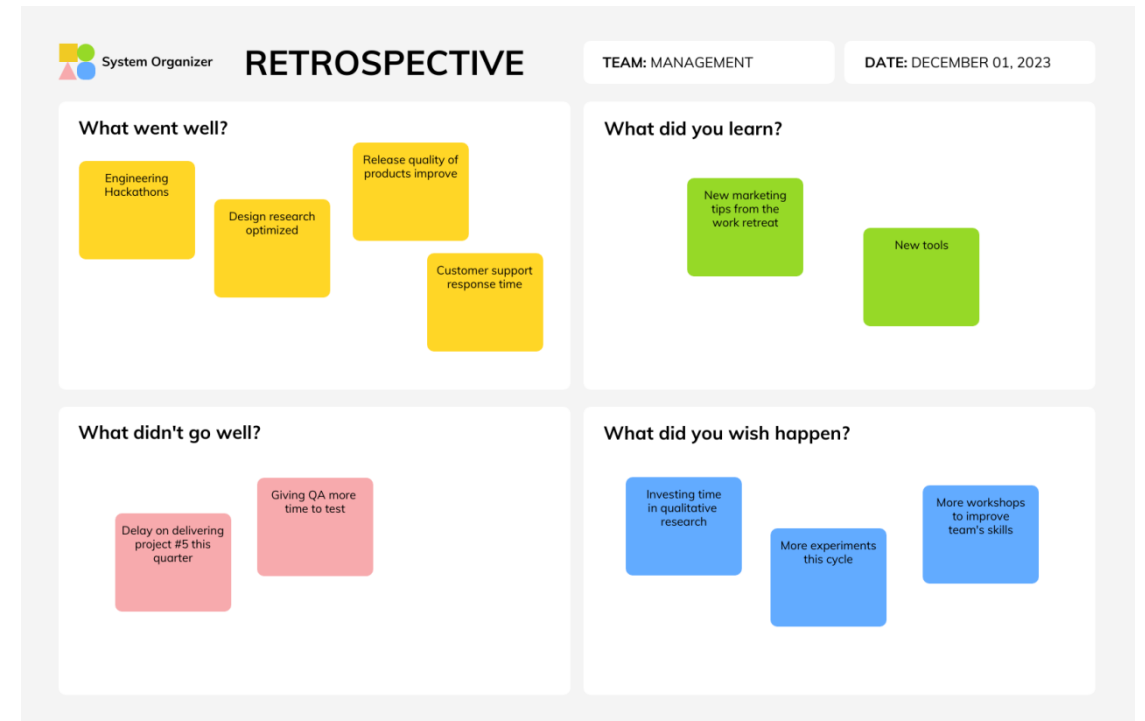
Чому ретроспектива важлива?

- Допомога у виявленні проблем команди і області для покращень
- Найкращий спосіб вирішення проблем або покращення
- Мотивує команду: команда відчуває себе в силах змінити процес так, як вони хочуть
- Дозвольте сказати «спасибі» членам команди і поділитися найкращими практиками всередині команди
- Тімбілдінг

# Manual QA

## Дошка ретроспективи

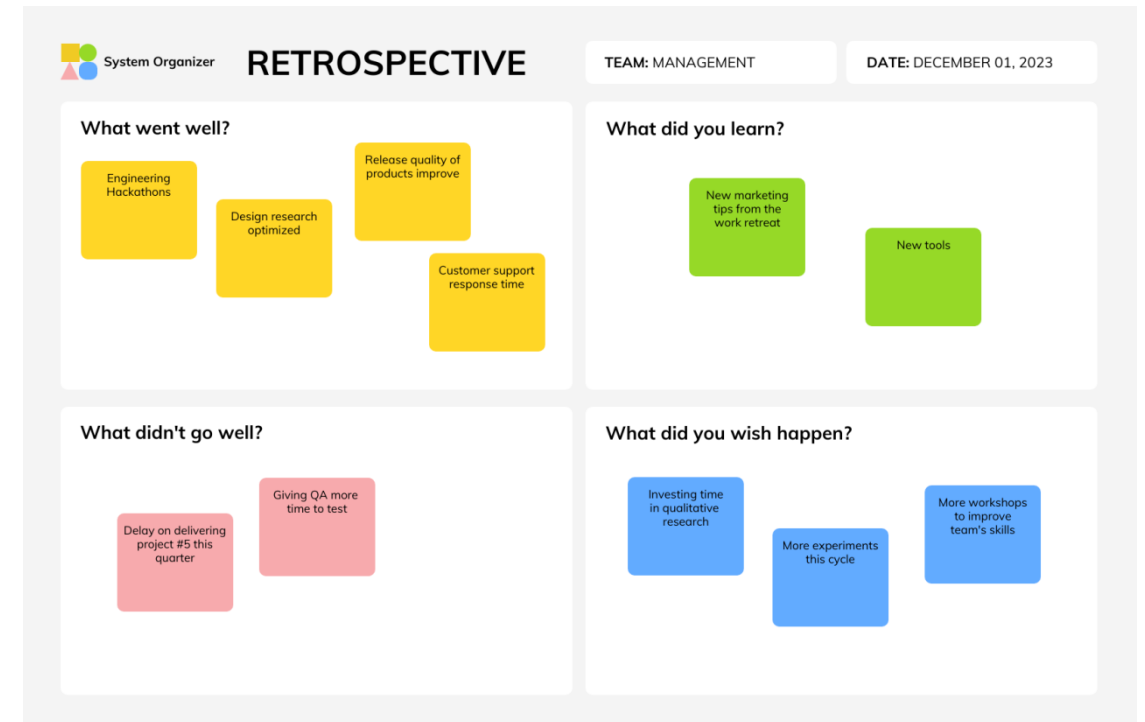
- Пишіть на стікерах:
  - що було добре
  - що потрібно припинити
  - що поліпшити
- Додайте на дошку
- Уточніть усі пункти
- Голосуйте за пункти, які мають бути покращені насамперед



# Manual QA

## Як провести ретроспективу ефективно?

- Не звинувачуйте інших
  - Говоріть про факти, а не про людей
- Зосередьтеся не на проблемах, зосередьтеся на рішенні
- Підготуйтеся
  - Заздалегідь проаналізуйте спринт (чи все було доставлено вчасно? Чи все було зроблено вчасно? Чи все було зроблено?)
  - Не тільки підіймайте проблеми, але й пропонуйте рішення
- Використовуйте факти та цифри
  - Наприклад, доставка була із затримкою на X днів



## Ретроспектива. Що, якщо?

### Що може статися?

- Команда соромиться відкрито говорити про проблеми
- Занадто багато тиску на термінах розробки, немає часу на ретроспективи!
- У нас є ретроспективи, але нічого не змінюється

### Що можна зробити?

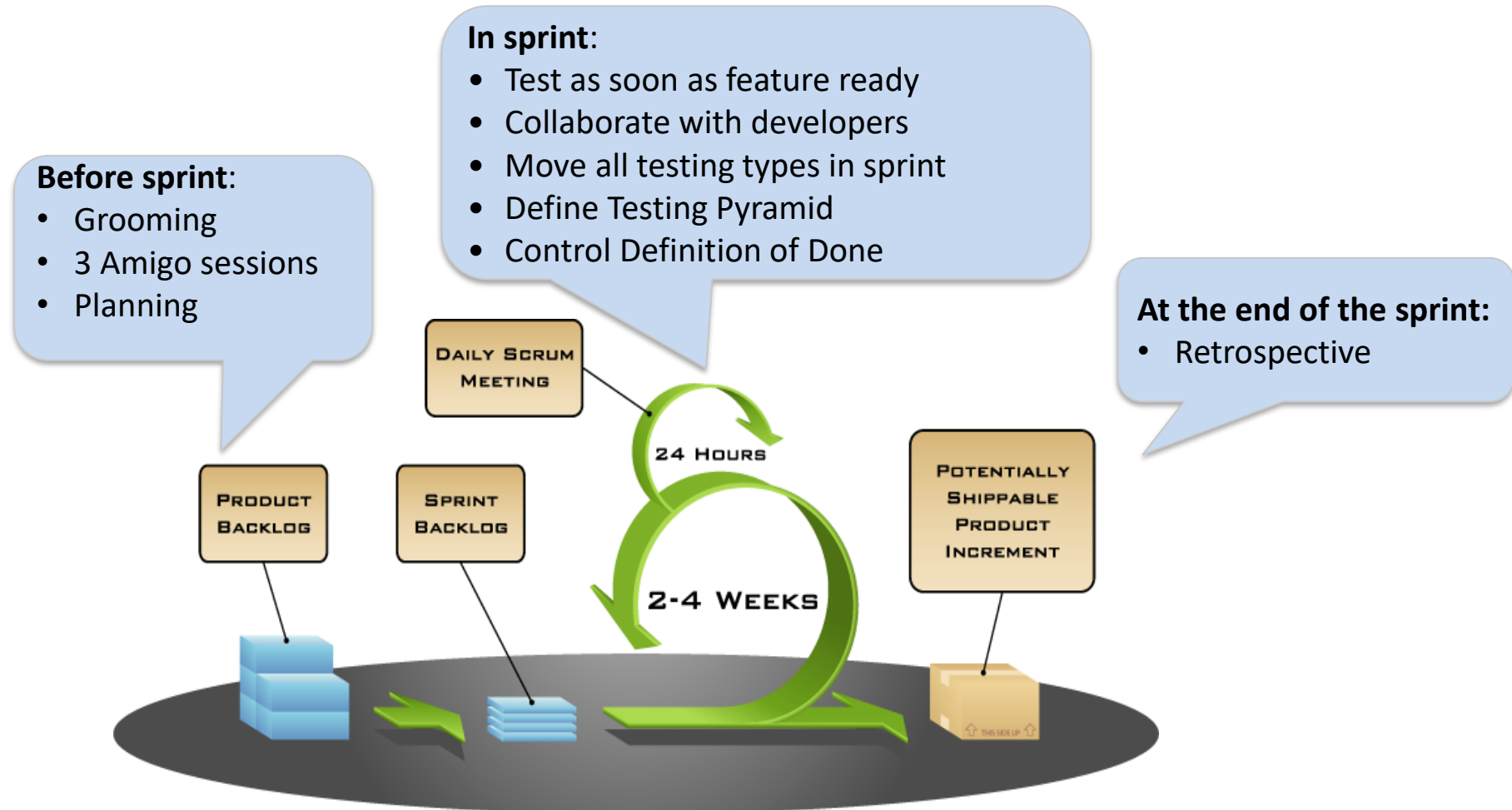
- Нагадайте команді про мету ретро: знайти рішення, а не винних
- Організуйте ретро так, щоб у кожного була можливість висловити/написати свою думку
- Це перше питання для обговорення на зустрічі!
- Подумайте про те, що можна зробити разом
- Під час ретро команда голосує за дії, які мають бути зроблені у наступному спринті. Включіть їх до беклогу
- Для кожної дії – власник та часова шкала (навіть невеликий крок у правильному напрямку)
- Почніть ретро з огляду дій, які мають бути зроблені з попереднього ретро



Що ми сьогодні вчили

# Manual QA

## Де тестувальники можуть вплинути на якість?



# Manual QA

## Підсумок. Ваш вплив на якість

У реальному світі	Ваша реакція
<ul style="list-style-type: none"><li>Команда розподілена, різниця часових поясів може становити 10 годин</li><li>РО з боку клієнта та важкодоступний для команди</li><li>До команди входять розробники, тестувальники, дизайнери, DevOps</li></ul>	<ul style="list-style-type: none"><li>Заздалегідь налаштуйте всі необхідні зустрічі команди / просто збережіть часовий інтервал для можливих питань</li><li>Використовуйте веб-камери для комунікації з командою на зборах</li><li>Погодьте доступний час із РО для питань/зустрічей команди</li></ul>
<ul style="list-style-type: none"><li>Розробники пишуть код до останнього моменту спринту</li><li>Тестування не може бути завершено у спринті</li><li>Деякі типи тестування виконуються поза спринтами (регресія, інтеграція, продуктивність)</li></ul>	<ul style="list-style-type: none"><li>Домовитися з розробниками про перші/останні дні постачання</li><li>Доставляйте функціонал ітеративно</li><li>Працюйте спільно з розробниками</li><li>Не постачайте непротестовані фічі</li><li>Спробуйте перемістити всі типи тестування у спринт</li></ul>

# Manual QA

## Підсумок. Ваш вплив на якість

У реальному світі	Ваша реакція
<ul style="list-style-type: none"><li>• Клієнт тисне на команду та терміни</li><li>• Вимоги змінюються у спринті</li><li>• Вимоги чітко не визначені</li></ul>	<ul style="list-style-type: none"><li>• Покажіть швидкість команди та оцінки роботи</li><li>• Поясніть ризики зниження оцінок</li><li>• Під час планування збережіть деякі можливості команди для таких термінових запитів</li><li>• Погодьтеся взяти їх у спринт замість якихось інших історій</li></ul>
<ul style="list-style-type: none"><li>• Команда соромиться відкрито говорити про проблеми</li><li>• Занадто багато тиску на термінах розробки, немає часу на ретроспективи!</li><li>• Ми маємо ретроспективи, але нічого не змінюється!</li></ul>	<ul style="list-style-type: none"><li>• Нагадайте команді про цілі ретро.</li><li>• Організуйте ретро так, щоб у кожного була можливість висловити/написати свою думку.</li><li>• Під час ретро команда голосує за дії, які мають бути виконані у наступному спринті.</li><li>• Якщо ці дії занадто великі, розділіть їх на маленькі кроки та визначте, які кроки будуть виконані у наступному спринті.</li><li>• Почніть ретро з огляду дій, які повинні бути зроблені з попереднього ретро</li></ul>

# Manual QA

## Підсумки

Що одне, найголовніше, ви дізнались сьогодні?

# Manual QA

Дякую за увагу! До нових зустрічей!

# Інформаційний відеосервіс для розробників програмного забезпечення

