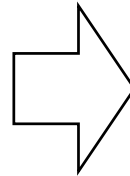

Планирование тестирования в тест-циклах

Уровни тестирования

КОМПОНЕНТНОЕ

- Автономное
 - отдельные модули
- Интеграционное
 - стыковка модулей

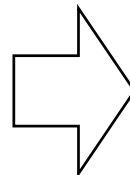


- база тестирования -- дизайн, внутренние интерфейсы
- выполняется разработчиками
- glass & "grey" box
- покрытие кода

речь идёт об этих уровнях

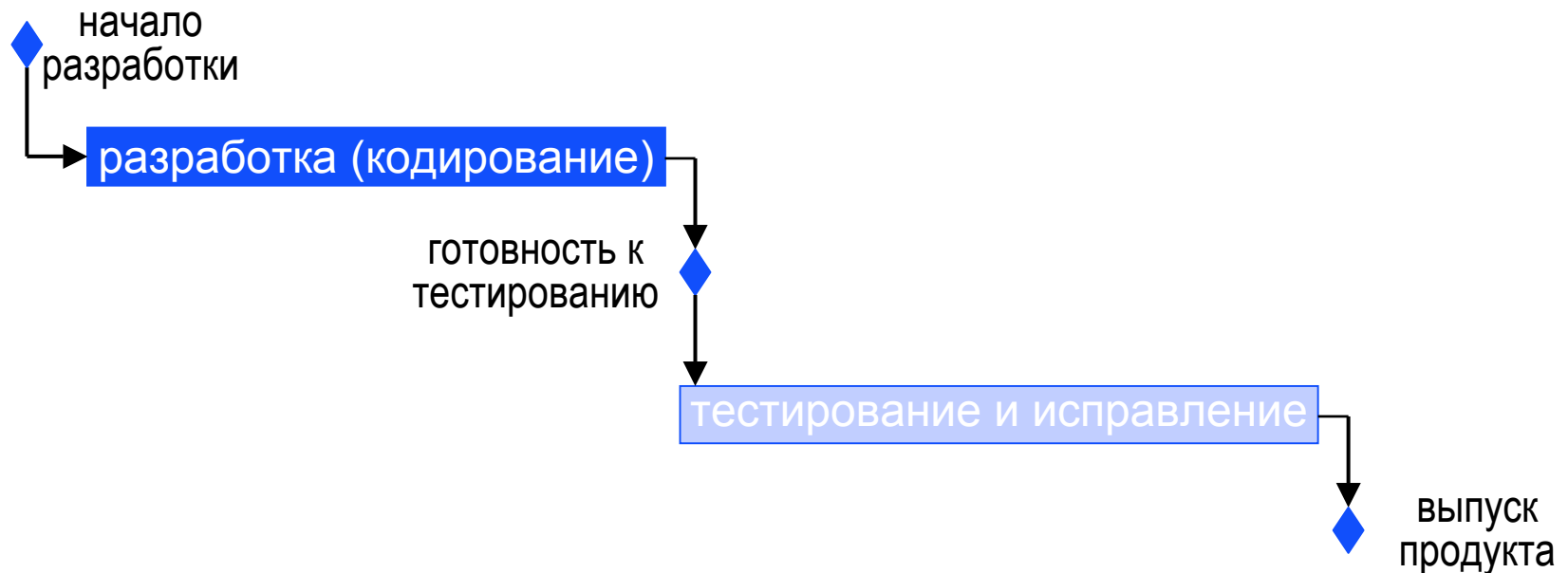
квалификационное

- «Подсистемное»
 - промежуточный выпуск
 - функциональный компонент
 - «инкремент»
- Системное
- Приемочное



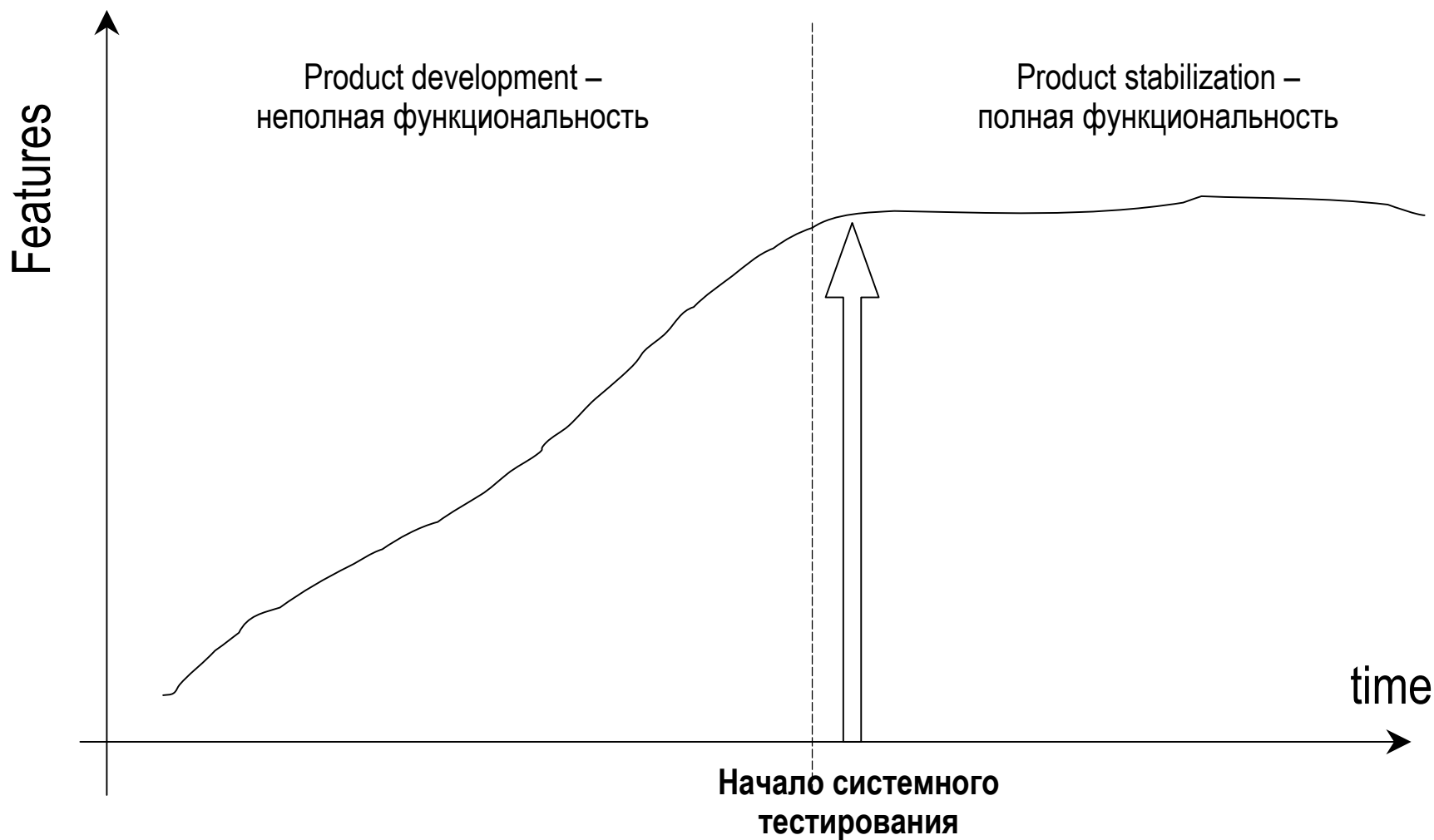
- база тестирования -- внешние требования
- Выполняется тестерами
- black box
- покрытие требований

Типичный фрагмент план-графика (waterfall)



- Когда вся функциональность реализована, начинается системное тестирование

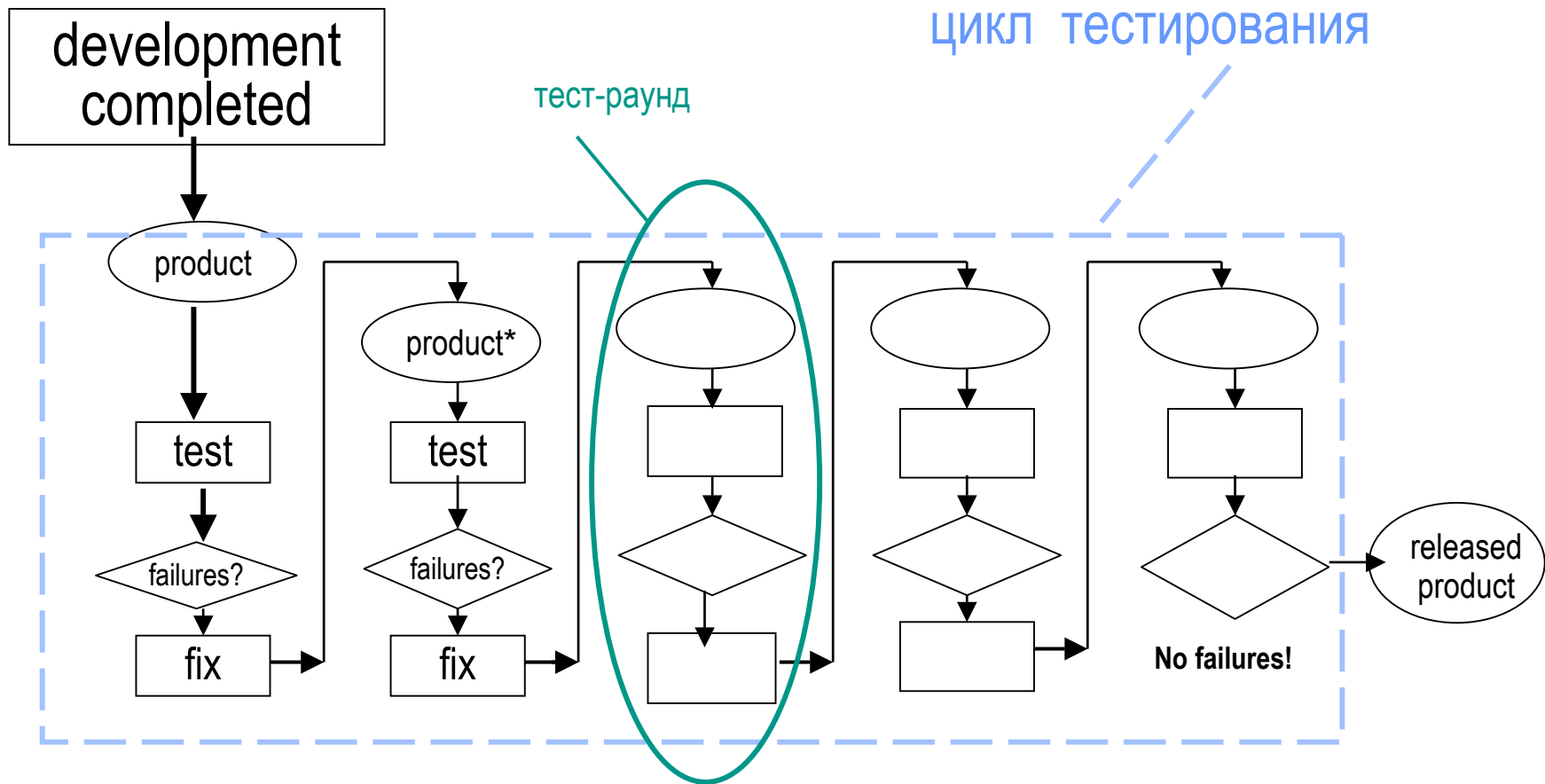
Системное тестирование в другом разрезе



Характеристики системного тестирования

- В ходе СТ обнаруживаются, исправляются и проверяются дефекты и несоответствия системы
- Кроме исправлений, никаких изменений в системе не предполагается: функциональность в ходе тестирования не меняется
- СТ состоит из «прогонов» (тест-раундов), в каждом из которых тестируется очередная собранная версия системы (build)
- СТ завершается, когда система (очередная версия) имеет требуемый уровень качества/готовности
- Уровень готовности (в терминах числа и степени серьёзности дефектов) обычно определяется до начала СТ

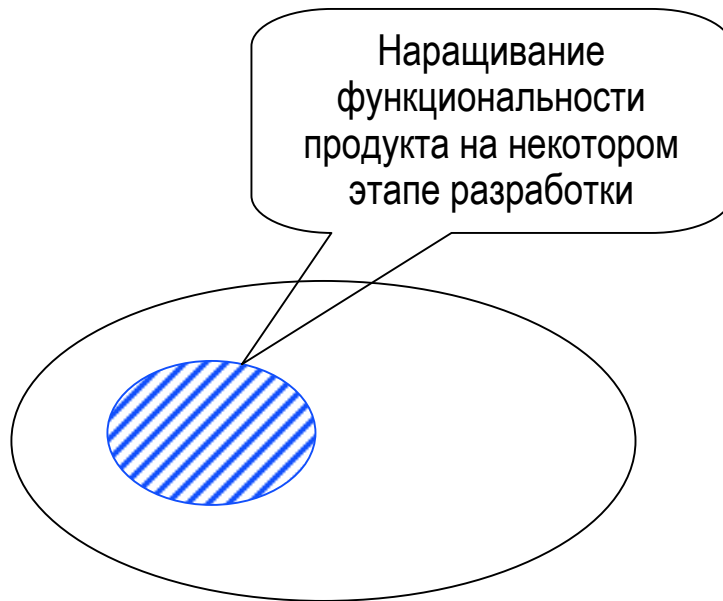
Общая схема системного тестирования



тест-раунд = раунд тестирования и исправления дефектов

Характер изменений в продукте на разных фазах

Неполная функциональность

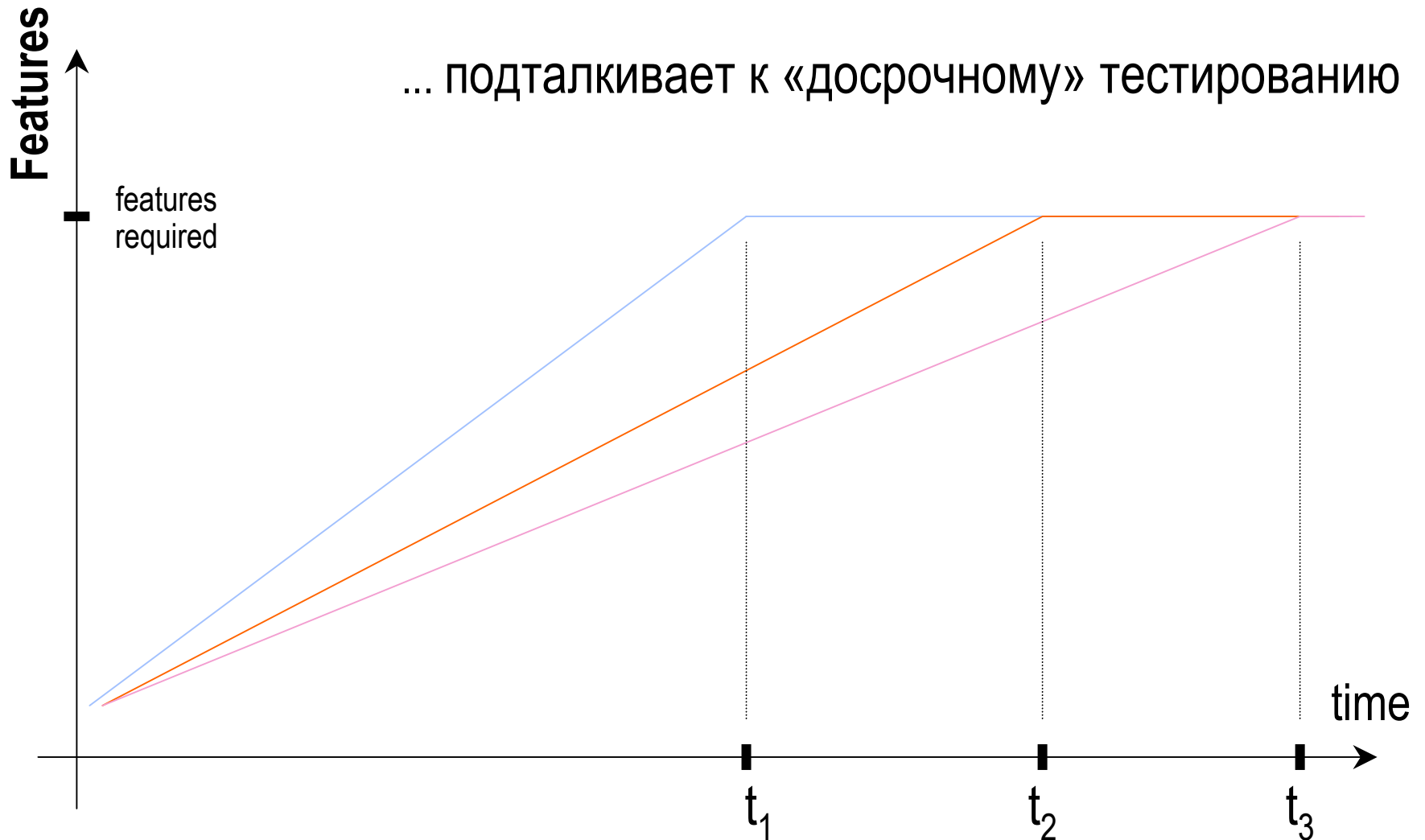


Полная функциональность

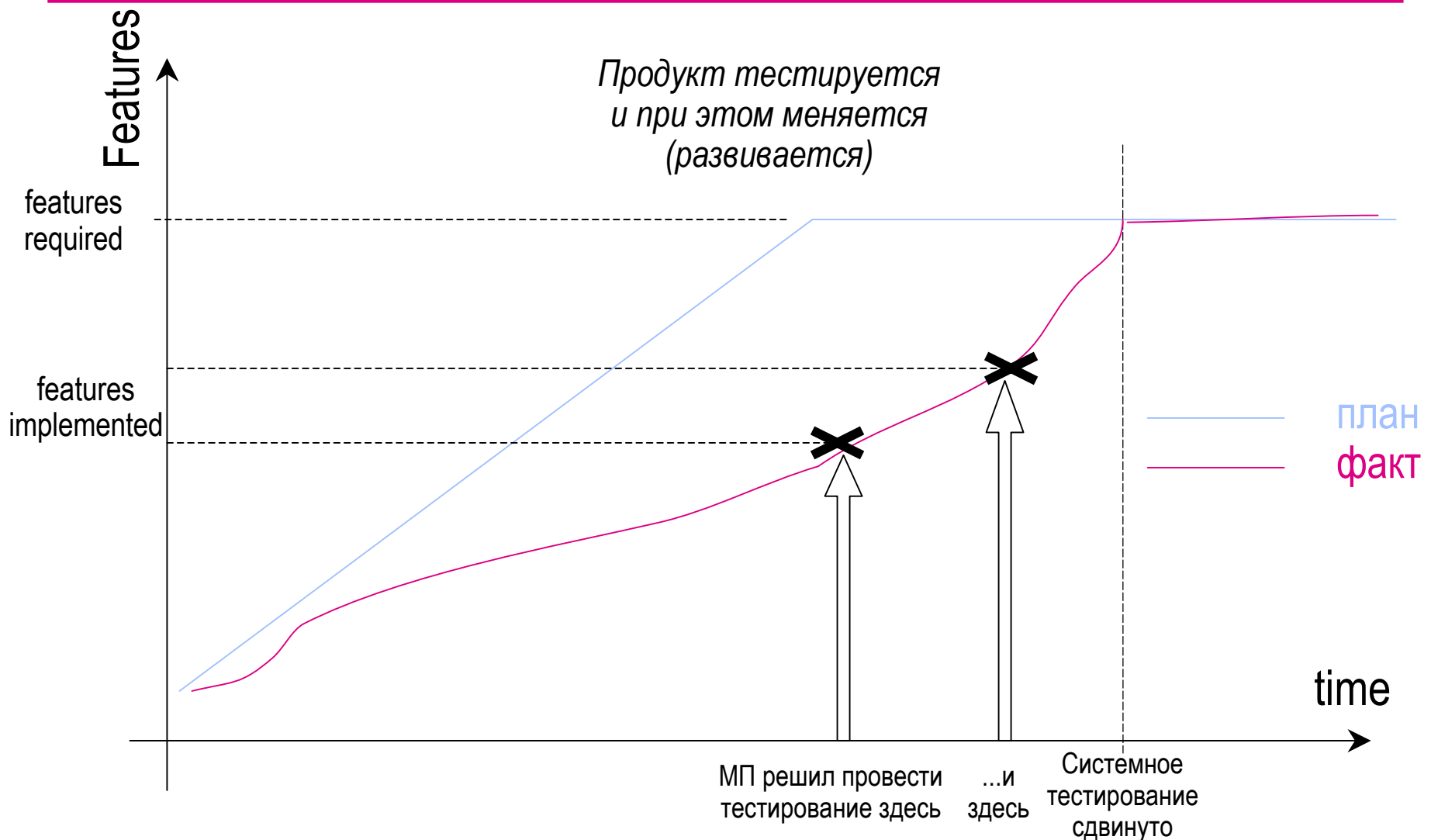


Начало системного тестирования

Поздняя стабилизация продукта



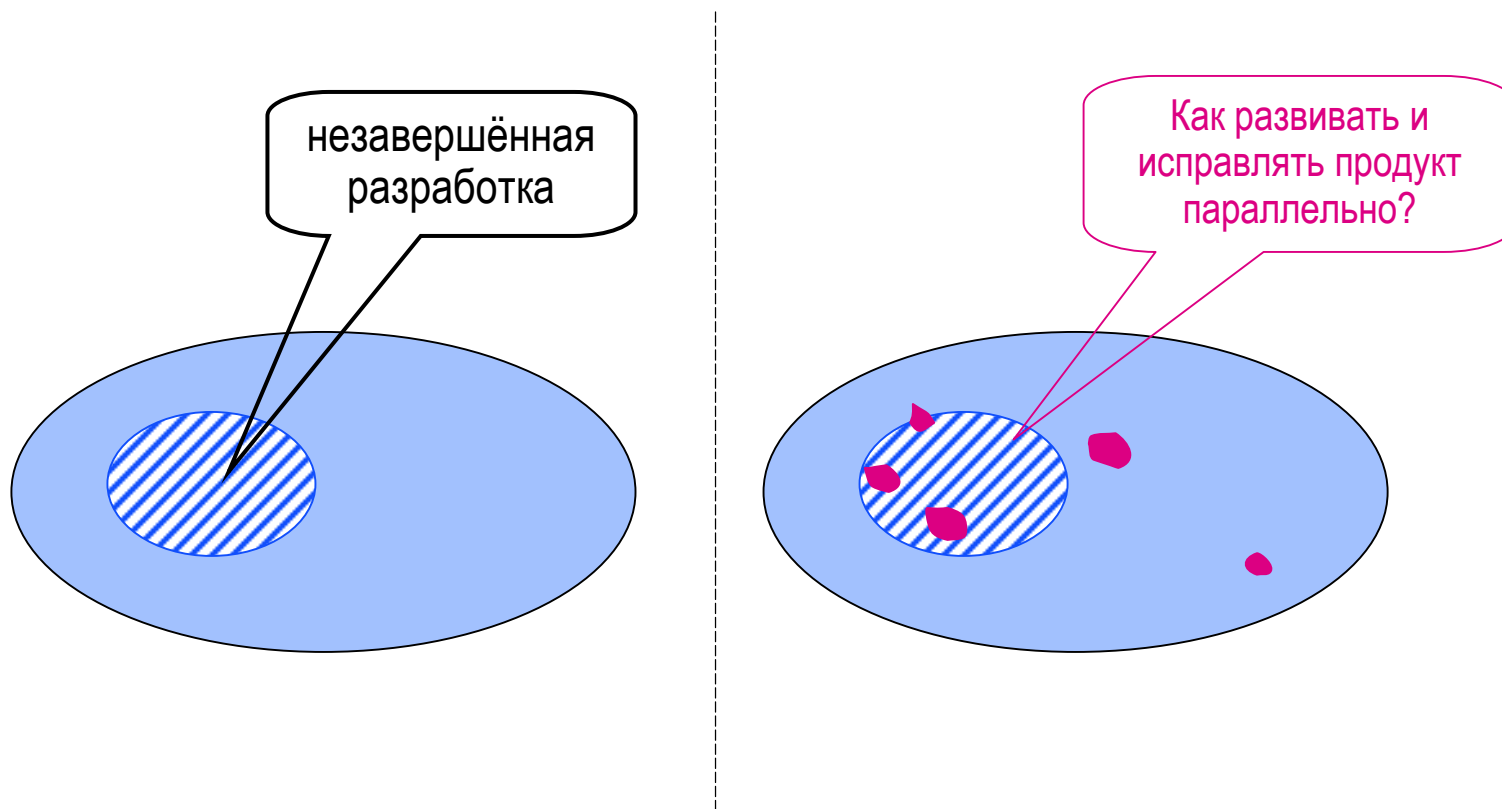
Планировали как лучше, а получилось...



«Досрочное» тестирование ставит массу вопросов

- **Каковы рамки тестирования?**
 - Обычная практика: определяются «на ходу» (нет в плане-графике); хаотичность
- **Сколько раундов проводится?**
 - Обычная практика: рамки меняются с каждым раундом, раунды не обязательно связаны друг с другом
- **Как исправляются и проверяются дефекты? Достигаются ли цели по качеству продукта?**
 - Обычная практика: дефекты откладываются (не исправляются систематически)
 - Цели не ставятся (нет в плане)
 - Нет достоверной статистики

Одна из проблем хаотичного тестирования



Начало тестирования
продукта с неполной
функциональностью

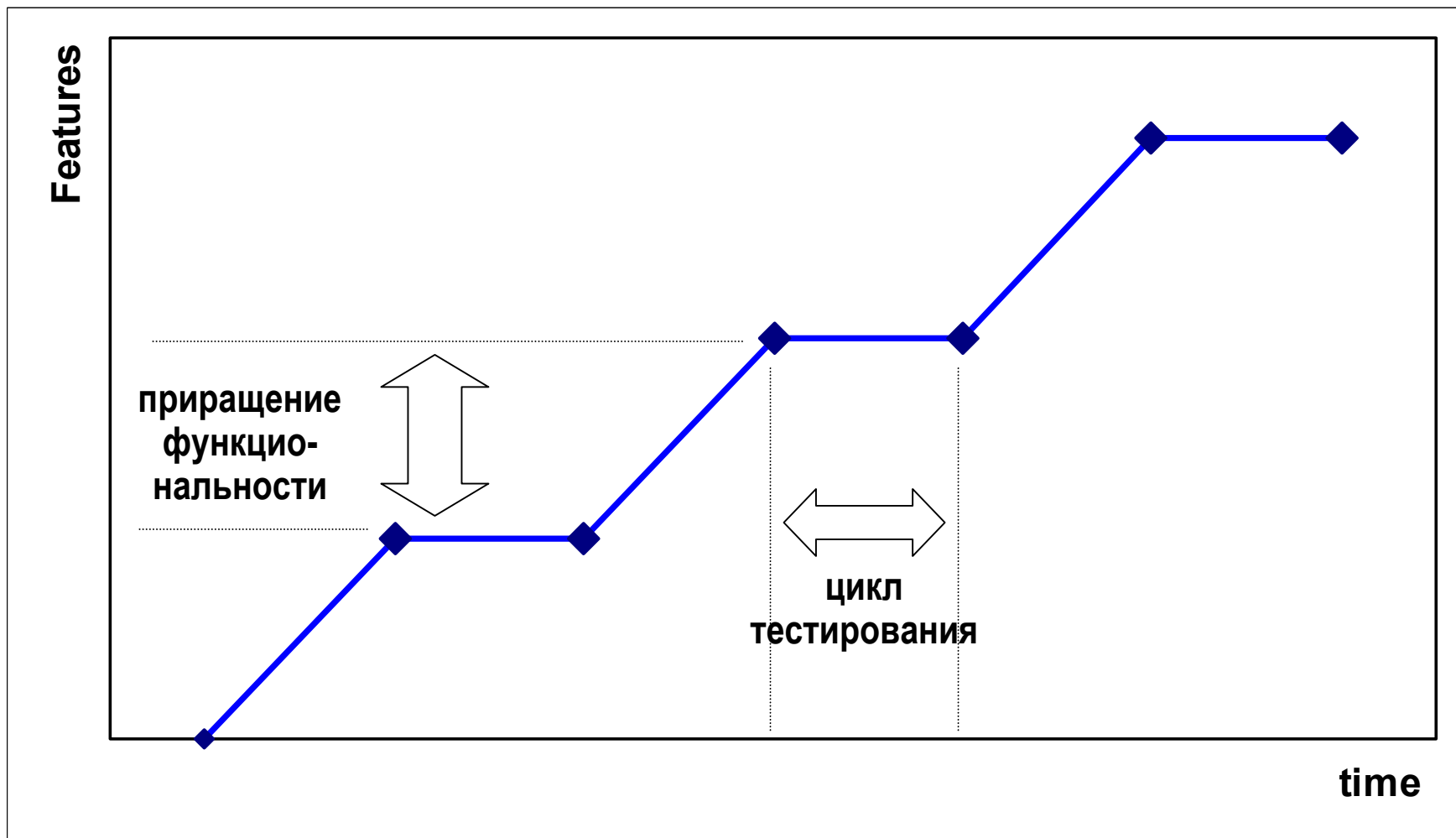
Риски при хаотичном тестировании

- если область, в которой тестировался продукт, в дальнейшем была существенно изменена или удалена, то результаты тестирования (в этой области) бесполезны, а усилия тестеров напрасны (бюджет потрачен впустую!);
- если область продукта была изменена раньше, чем исправлен найденный в ней дефект, то дефект может «спрятаться» и неожиданно всплыть позже
- могут быть внесены новые дефекты в те области продукта, где их по результатам тестирования не было (нужно регрессионное тестирование?)
- область тестирования меняется: ЧТО тестируем?

Следствие:

НЕОПРЕДЕЛЁННЫЙ СТАТУС (КАЧЕСТВО) ПРОДУКТА

Циклы тестирования «внутри» разработки



Характеристики цикла тестирования

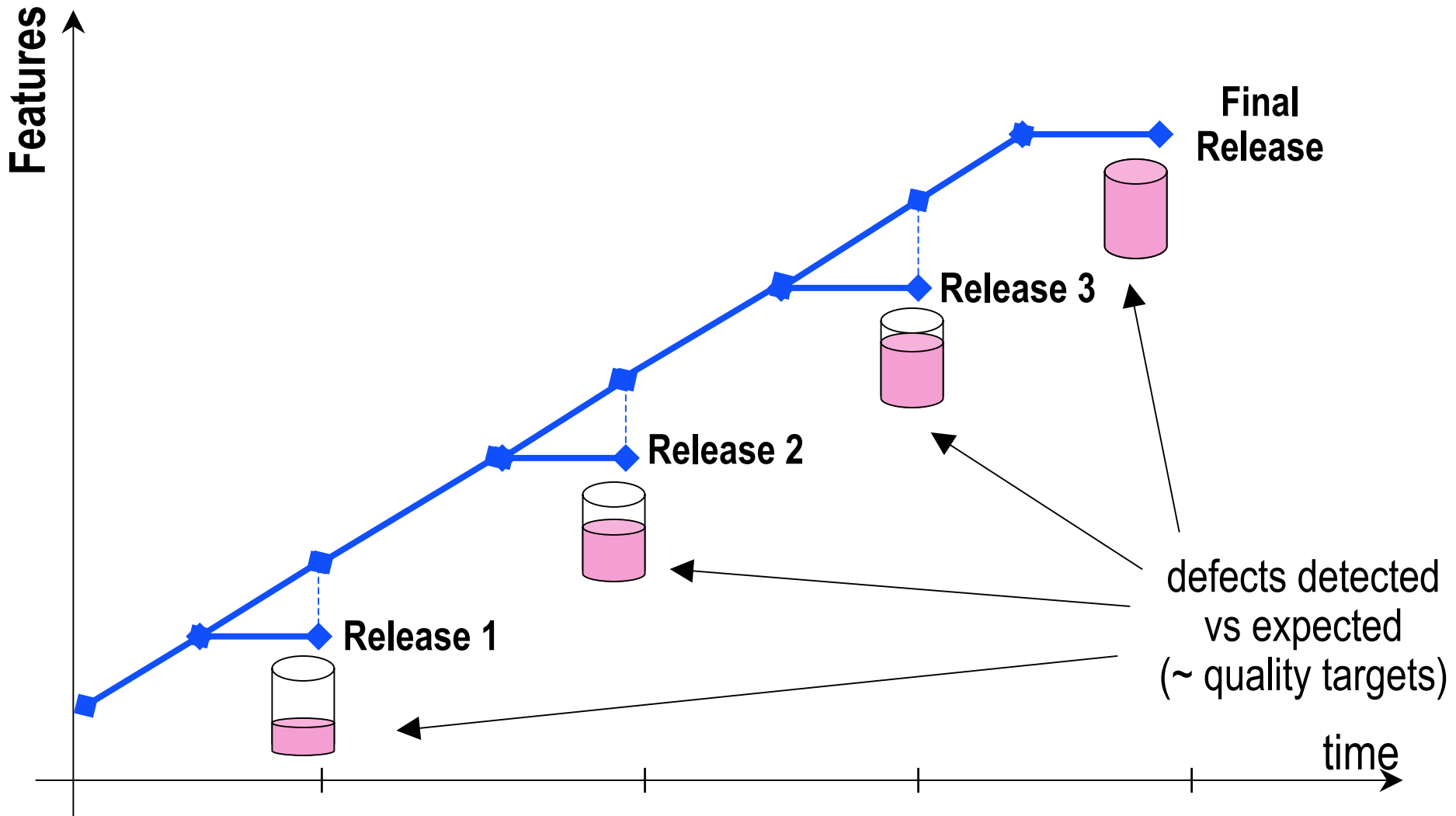
- К началу тест-цикла завершена реализация промежуточного продукта, он содержит всю функциональность в запланированных рамках; эта функциональность составляет область тестирования
- Дефекты и несоответствия обнаруживаются, исправляются и проверяются *в области тестирования*. Кроме исправлений, никаких *изменений в области тестирования* не предполагается
- Тест-цикл даёт «здоровые» метрики для определения уровня качества/готовности продукта в области тестирования (тренды обнаружения и исправления дефектов)
- Критерии готовности определяются до начала тест-цикла («цели по качеству»). Тест-цикл завершается, когда продукт (очередная версия) имеет требуемый уровень готовности в *области тестирования*

Что даёт тест-цикл

В результате выполнения цикла тестирования мы

- имеем объект измерений -- продукт, взятый в определённый момент разработки
 - мы «фиксируем» состояние продукта (baseline); всегда можем собрать его в этом состоянии
- получаем достоверную информацию о качестве продукта
 - в виде совокупности измерений данного объекта
 - выполнить эти измерения можно только в формате тест-цикла
- знаем, достигли ли нужного нам на данный момент уровня качества продукта
 - мы знаем, в какой мере зафиксированное состояние продукта (подсистема, ядро, функциональный слой) свободно от дефектов (в какой мере достигнуты критерии)
- снижаем риски разработки
 - развиваем продукт от достигнутого состояния («руберж»), делая поправки в зависимости от достигнутого

Quality targets and test cycles



Риски, на снижение которых направлены тест-циклы

- **Недостоверность информации об объекте тестирования**
 - Неопределённость объекта/области тестирования: меняются рамки
 - Изменения в объекте (не связанные с исправлениями), даже когда он определён
 - Критерии оценки продукта отсутствуют
- **Недостижение нужного качества продукта в проектные сроки**
 - Следствие предыдущего: не знаем, с какой скоростью движемся => рискуем не успеть

Тест-циклы: тестирование на всём протяжении проекта

