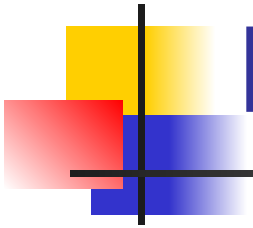
Анализ стандартных методов тестирования.

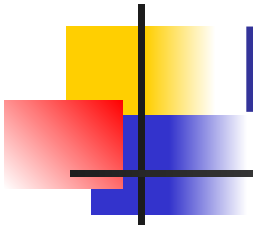
Применимость к разработке игр.

Шишенин Александр Apeiron lead-tester



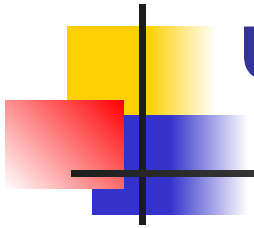
Несколько аксиом

- Тестирование способно выявить только наличие ошибки, а не их отсутствие.
- Удачный тест, это который нашел ошибку.



Еще аксиомы

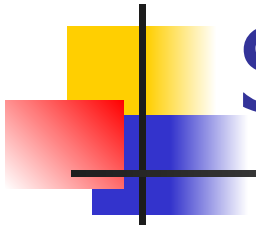
- Только полное тестирование всех вариантов способно обеспечить отсутствие ошибок.
  - Обычно невозможно.
- Старайтесь проверять систему целиком, как можно раньше.
- Сначала проверяйте старую функциональность, а потом уже новую.
- Проверки типичных ситуаций важнее граничных случаев.



Что есть тестирование?

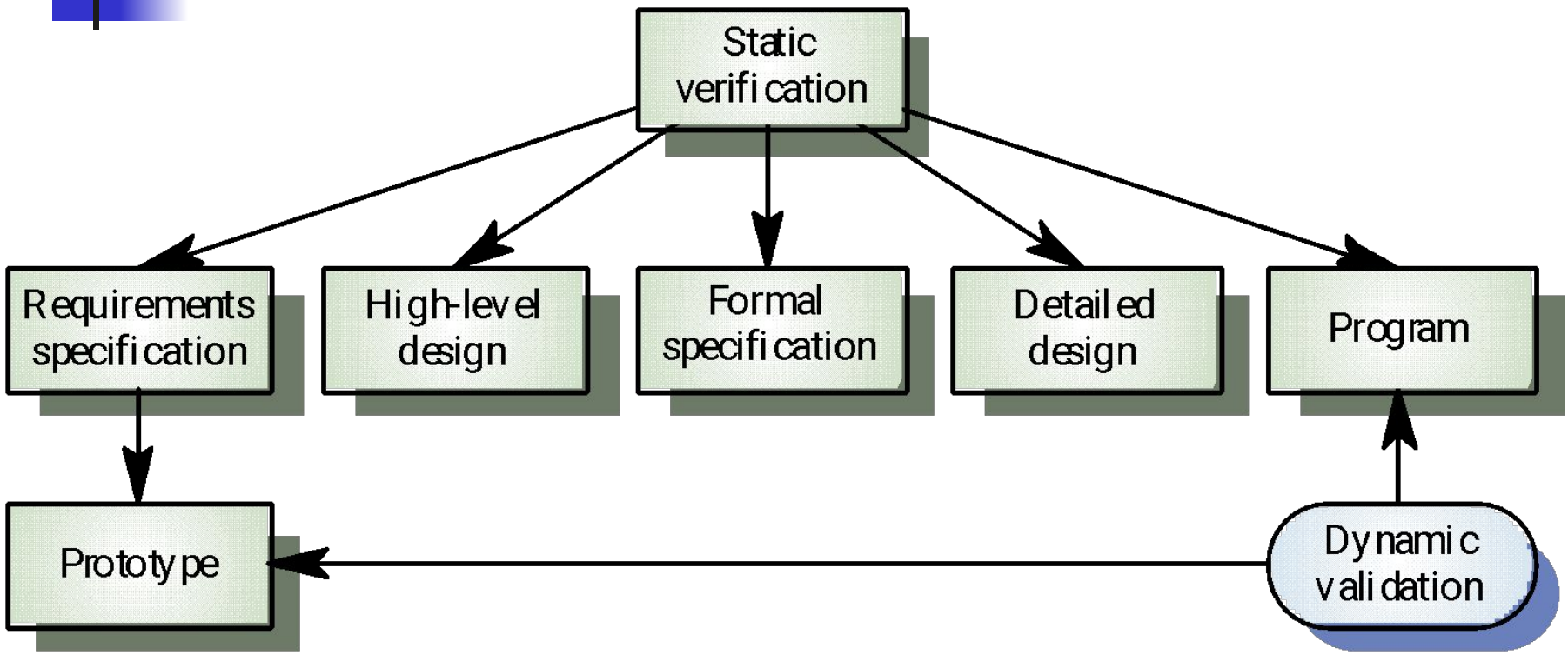
- Verification:
  - “Мы правильно создаем проект?”
  - Соответствие спецификации.
- Validation:
  - "Мы создаем правильный проект?”
  - Соответствие ожиданиям и нуждам пользователей.

Static and dynamic verification

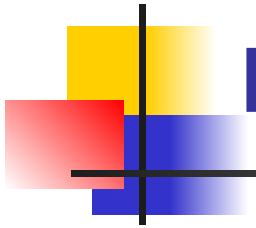


- Статические. *Software inspections.*
  - Проверки документации, кода и пр.
- Динамические. *Software testing.*
  - Проверка поведения программы.

Static and dynamic V&V



Проверка путей выполнения программы.

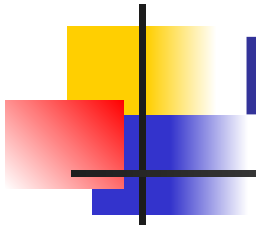


- Цель – создать набор тестов, на которых каждая инструкция будет выполнена хоть один раз.
- Начальным данным для построения таких тестов является исходный код.
- Условные инструкции являются вершинами этого графа.
- Проверить все комбинации путей часто невозможно. Делайте выборку.

Тестирование сценариями.

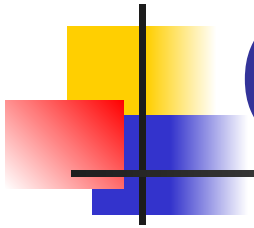
- Надо определить случаи использования.
- Найти потоки взаимодействия.
- Моделирование интересующих ситуаций при помощи скриптов или при помощи мульта.

Интеграционное тестирование.

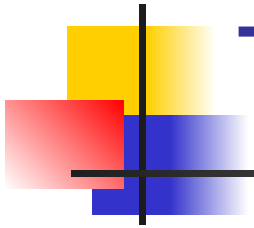


- У нас есть набор модулей, которые работают корректно. Надо собрать воедино.
- Основная трудность – локализация ошибки.
- Инкрементальная интеграция облегчает локализацию ошибки.
- Сверху вниз или снизу вверх. Надежность и стоимость.

Тестирование интерфейсов. (не пользовательских)

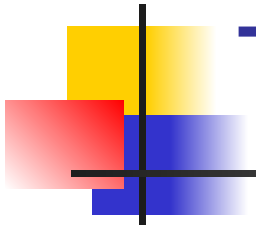


- Проверка информации общей или передаваемой между подсистемами.
- Характерна для ООП.



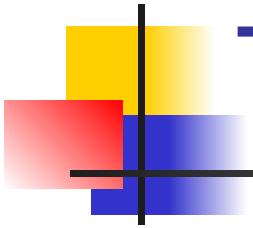
Типы интерфейсов.

- Параметры передаваемые между процедурами.
- Общая память.
- Процедурные интерфейсы. Наборы процедур предоставляемые другой подсистеме.
- Передача сообщений.



Типичные ошибки.

- Неправильно использование интерфейса.
  - Например неправильный порядок параметров.
- Неправильное понимание. При вызове мы подразумеваем другое поведение.
  - Например по событию смерти глав героя можно сразу проиграть ролик и выйти в меню, а можно и нет.



Тестирование ресурсов.

- Когда начинать?
- Как координировать работу?
- Создание читов, обходов.
- Локализация ошибки.
  - Ошибка в ресурсе или в экспортере?

Сборка версий.

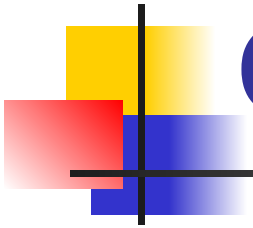
- Финальная сборка.
  - Выбираем компоненты, которые будут включены.
- Автоматические средства сборки.
  - Скрипты и прочее.
- Сбор необходимых ресурсов.
  - Как ничего не забыть.

Стресс тесты.

- Перегрузка системы сверх заданных границ.
  - Может показать узкие места (bottle neck).
- Стресс тесты - приводящие к падениям системы.
  - Желательно добиться сохранения как можно большего количества информации или локализовать падение. Оно не должно быть катастрофическим.
- Характерно для сетевых приложений.
  - Перегрузка сети.

Инспекции кода. Причины возникновения.

- Никакое разумное тестирование не может дать гарантии отсутствия ошибок.
- “Дебажить” приходится всегда.
- Есть потенциальные ошибки, которые лучше тоже уничтожать.

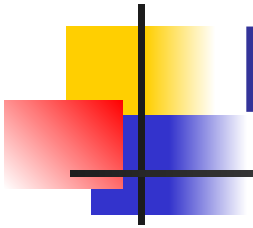


Суть метода.

- Проверить можно все, что можно читать.
- Не требует запуска системы.
- Много разных дефектов можно найти за 1 раз.
- Обучение персонала.
- Списки стандартных ошибок.

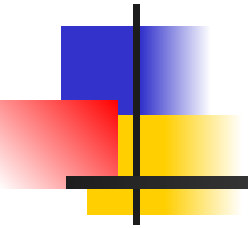
Необходимые условия.

- Четкая спецификация.
- Члены команды должны быть знакомы со стандартами организации.
- Синтаксически корректный код.
- Подготовленный checklist.
- Менеджмент должен понимать, что это увеличит стоимость разработки на ранних этапах.
- Менеджмент не должен использовать результаты для карающих мероприятий.



Порядок проведения

- Сначала всем выдают материалы, которые необходимо проверить.
- Каждый готовится по отдельности, отмечает неточности.
- Сама инспекция с выявлением дефектов.
- Автор исправляет недостатки.
- Повторная инспекция.



Вопросы.