

Сквозной структурный контроль

Это совокупность технологических операций контроля.

Цель – как можно раньше обнаружить ошибки.

Термин **«сквозной»** отражает **выполнение контроля на всех этапах** разработки.

Термин **«структурный»** означает **наличие четких рекомендаций** по выполнению контролирующих операций **на каждом этапе**.

Условия и порядок проведения ССК:

- Применяется при нисходящей разработке.
- Выполняется на специальных контрольных сессиях.
- Материалы для очередной сессии **выдаются** участникам **заранее**.
- Проводит сессию руководитель проекта или главный программист.

Выявлена закономерность

Чем чаще сессии, тем:

- ✓ меньше ошибок;
- ✓ упрощается процесс тестирования;
- ✓ ускоряется процесс разработки.

Приглашаются:

- Разработчики
- Специалисты по предметной области
- Люди от заказчика
- Другие эксперты

Этапы	Что проверяется
1. Определение требований	План проекта, сроки, область применения продукта
2. Определение спецификации	Спецификация на систему, постановка задачи, тесты контроля.
3. Проектирование	Функциональные спецификации. Например, схемы иерархии. Вручную по частям проверяют алгоритмы на конкретных наборах данных. («Белый ящик»)
4. Реализация	План реализации и тестирования модулей. Набор тестов. Эффективность взаимодействия модулей. Коды отдельных модулей. («Черный ящик»+структурный контроль)
5. Эксплуатация	Окончательно документация. Тесты в целом. («Оценочное тестирование»)

Основная задача — убедиться в **правильности концептуальных решений.**

- ✓ Для всех этапов **целесообразно иметь списки наиболее часто встречающихся ошибок.**
(Списки формируют по литературным источникам и исходя из опыта предыдущих разработок.)
- ✓ Все найденные ошибки фиксируют в специальном документе, **но сразу не исправляют их.**

Выводы:

Кроме **раннего обнаружения ошибок** **сквозной структурный контроль обеспечивает своевременную и качественную документацию по проекту.**

Оценочное тестирование

(тестирование системы в целом)

1. Тестирование удобства использования и эксплуатации.

- ❖ Определение степени прагматики, обучения, понятности и привлекательности для пользователей разрабатываемого продукта.

Например, определяется:

- **Сколько времени и шагов** понадобится пользователю для решения **основных задач** приложения.
- **Сколько ошибок** сделал пользователь во время работы с приложением?
- **Насколько быстрее** проводится **повторное выполнение операций** после перерыва в сравнении с **новым пользователем** и др.

- ❖ Тестирование интерфейса человек-машина. Учитываются психологические факторы, проверяется удобен ли интерфейс.

Например,

проверяется удобство навигации (для Web-сайтов).

В частности:

- могут быть реализованы все функции, **но пользоваться неудобно (например, выполняются лишние шаги);**
- Или расположение элементов интерфейса **нелогично**, программа быстро **вызывает утомление** и т.д.

2. Тестирование на предельных объемах.

Например, для систем, попадающих в класс БД, проверяют на больших объемах данных.

3. Тестирование на предельных нагрузках.

Проверяют возможность обработки пикового объема данных в течение заданного интервала времени.

Например, для интернет-приложений это может быть:

- количество посетителей за интервал времени;
- количество операций, одновременно выполняемых на платформе веб-сайта.

4. Тестирование защиты.

Проверяют, например, защиту от несанкционированного доступа к информации и др.

5. Тестирование производительности.

Определяют пропускную способность при заданной конфигурации и нагрузке.

6. Тестирование требований к памяти.

Определяют реальные потребности в оперативной и внешней памяти.

7. Тестирование конфигурации оборудования.

Проверка работоспособности программного продукта на разном оборудовании.

8. Тестирование совместимости.

Проверка корректной работы в определенном окружении.

Это может быть связано:

- с аппаратной платформой;
- с устройствами (сетевые устройства. внешние (принтеры, CD/DVD-приводы, веб-камеры и пр.);
- с программной средой функционирования (ОС (Unix, Windows, MacOS); Базы данных (Oracle, MS SQL, MySQL); Системное программное обеспечение (веб-сервер, файрволл, антивирус); Браузеры (Internet Explorer, Firefox, Opera, Chrome, Safari) и др.
- совместимость между версиями и др.

9. Тестирование удобства и правильности установки (Инсталлятора)

Инсталлятор включает **три действия**:

- ✓ установка
- ✓ обновление
- ✓ удаление.

Тестирование усложняется, если установка одной и той же программы осуществляется **на разные платформы**.

При установке проверяется корректность списка файлов в инсталляционном пакете

(например, должны отсутствовать проектные файлы, не включенные в инсталляционный пакет, т.е. не должны попасть на диск пользователя).

При установке проверяется регистрация приложения в ОС (регистрация расширений, права доступа пользователя и др.).

При обновлении, например, проверяется **обратная совместимость** создаваемых данных (сохранность и корректная работа созданных до обновления данных, возможность корректной работы старых версий приложения с данными, созданными в новых версиях и др.).

При удалении (Деинсталляции) проверяется **корректность удаления приложения** (удаление из системного реестра, удаление физических файлов приложения и др.).

10. Тестирование надежности.

Используют математические модели.

(Можно проверить свойство 2 часа простоя за 30 лет работы).

Степень надежности характеризуется вероятностью работы программного продукта без отказа в течение определенного периода времени.

Существует 4 основные составляющие функциональной надежности программ:

безотказность - свойство программы выполнять свои функции во время эксплуатации;

работоспособность - свойство программы корректно (так как ожидает пользователь) работать весь заданный период эксплуатации;

безопасность - свойство программы быть не опасной для людей и окружающих систем;

защищенность - свойство программы противостоять случайным или умышленным вторжениям в нее.

11. Тестирование восстановления.

Например, проверка восстановления БД после сбоев оборудования и программы. Рекомендуется тестировать восстановление из резервных копий регулярно.

12. Тестирование удобства обслуживания.

Проверка средств обслуживания, включенных в программное обеспечение.

13. Тестирование документации.

Проверка документации, например, если документация содержит примеры, то их все необходимо попробовать, а также инструкции.

14. Тестирование процедуры - проверка ручных процессов, предполагаемых в системе (АСУ).