

Инспектирование программ как метод оценивания качества



Инспектирование программ – это просмотр и проверка программ с целью обнаружения в них ошибок.

Идея формализованного процесса проверки программ была сформулирована корпорацией IBM в 1970-х годах. В настоящее время данный метод верификации получил широкое распространение. На его базе разработано множество других методов, но все они основываются на базовой идее метода инспектирования, согласно которому группа специалистов выполняет тщательный построчный просмотр и анализ исходного кода программы.

Инспектирование программ не требует их исполнения, поэтому данный метод можно использовать до завершения полной реализации программ.

Во время инспектирования проверяется исходное представление системы. Это может быть модель системы, спецификация или программа, написанная на языке высокого уровня. Для обнаружения ошибок используется знание разрабатываемой системы и семантика ее исходного представления. Каждую ошибку можно рассматривать отдельно, не обращая внимания на то, как она влияет на поведение системы.

Главное отличие инспектирования от других методов оценивания качества программ состоит в том, что его **цель – обнаружение дефектов**, а не исследование общих проблем проекта. Дефектами являются либо ошибки в исходном коде, либо несоответствия программы стандартам.

Процесс инспектирования – формализованный. В нем принимает участие небольшая группа людей (обычно не более, чем четыре человека).

У каждого в группе есть своя роль. Обязательно должны присутствовать: автор, рецензент, инспектор, координатор. Рецензент «озвучивает» программный код, инспектор проверяет его, координатор отвечает за организацию процесса.

Для начала процесса инспектирования программы необходимы следующие условия:

- наличие точной спецификации кода (без полной спецификации невозможно обнаружить дефекты в проверяемом программном компоненте);
- члены инспекционной группы должны хорошо знать стандарты разработки;
- в распоряжении группы должна быть синтаксически корректная последняя версия программы (нет смысла рассматривать код, который «почти завершён»).

Сам процесс инспектирования должен быть относительно коротким (не более двух часов) и сосредоточенным только на выявлении дефектов, аномалий и несоответствий стандартам. Инспекционная группа не должна предлагать способы исправления дефектов или рекомендовать какие-либо изменения в других программных компонентах.

После инспектирования автор изменяет программу, исправляя обнаруженные ошибки. На этапе доработки координатор принимает решение о том, необходимо ли повторное инспектирование. Если повторное инспектирование не требуется, все обнаруженные дефекты фиксируются документально.

Доказано, что инспектирование является эффективным методом обнаружения ошибок. Также немаловажно, что инспектирование значительно дешевле экстенсивного тестирования программ. В экспериментах, сравнивалась эффективность инспектирования и тестирования.

Инспектирование программного кода оказалось более эффективным и менее дорогостоящим, чем тестирование. Более 60% ошибок в программах можно обнаружить с помощью неформального исследования (инспектирования) программ.

Процесс инспектирования также может оценить качественные характеристики систем: соответствие стандартам, переносимость и удобство сопровождения.

Конечно, инспектирование не может полностью заменить тестирование. Инспектирование лучше использовать как начальный процесс верификации для обнаружения большей части программных дефектов.

Путем инспектирования проверяют соответствие ПО ее спецификации, однако таким способом нельзя проверить динамическое поведение системы. Более того, нерационально инспектировать законченные системы, собранные из нескольких подсистем. На этом уровне возможно только тестирование. Тестирование также необходимо для оценки надежности и производительности, проверки пользовательского интерфейса и соответствия системы требованиям заказчика.

Инспектирование на соответствие стандартам кодирования

Стандарт оформления кода (стандарт кодирования, стиль программирования) — набор правил и соглашений, используемых при написании исходного кода на некотором языке программирования.

Стандарт оформления кода обычно принимается и используется некоторой группой разработчиков программного обеспечения для единообразного оформления совместно используемого кода.

Стандарт сильно зависит от используемого языка программирования.

Обычно, стандарт оформления кода описывает:

- способы выбора названий и используемый регистр символов для имён переменных и других идентификаторов:
- стиль отступов при оформлении логических блоков ;
- способ расстановки скобок, ограничивающих логические блоки;
- использование пробелов при оформлении логических и арифметических выражений;
- стиль комментариев и использование документирующих комментариев.

Образцом для стандарта кодирования может стать набор соглашений, принятых в какой-либо распространенной печатной работе по языку, широко применяемая библиотека или API.

Реже разработчик языка выпускает подробные рекомендации по кодированию на нём. Выпущены, например, стандарты кодирования на C# от Microsoft и на Java от Sun.

Предложенная разработчиком или принятая в общеизвестных источниках манера кодирования в большей или меньшей степени дополняется и уточняется в корпоративных стандартах.

Основные принципы распространённых стандартов кодирования в последнее время оказывают влияние на синтаксис вновь создаваемых языков программирования. В некоторых из них соглашения, ранее применявшиеся только в стандартах кодирования, стали обязательными элементами синтаксиса.

В некоторых современных языках ([Python](#)) отступы влияют на логику исполнения (т.е. блоки кода выделяются не ключевыми словами, а размером отступов), в других ([Ruby](#)) — частью языка стали соглашения о регистре букв и префиксах для типов, констант, переменных.

В результате, если ранее недисциплинированный программист мог игнорировать стандарты кодирования из личных соображений, то теперь, при работе на новых языках, соблюдение стандартов в определённой мере контролируется транслятором.

Соблюдение стандарта кодирования дает программисту массу преимуществ. Код становится более однородным, более читабельным, более простым для восприятия. Упрощаются поиск ошибок в коде, модификация, сопровождение и повторное использование кода.

Стандарты кодирования можно сравнить с правилами движения. Как грамотные водители соблюдают правила движения, так и грамотные программисты следуют стандарту кодирования. Каждый из нас знает, что если правила систематически нарушать, то рано или поздно неприятностей не избежать. Именно тех неприятных неприятностей, из-за которых это правило и было придумано.

Рекомендации по стилю написания программ на C++

Самостоятельно найти в Интернет