

*Сафиуллин Тимур Валерьевич*

магистрант

*Серебрякова Татьяна Александровна*

канд. экон. наук, доцент

ФГБОУ ВО «Тихоокеанский государственный университет»

г. Хабаровск, Хабаровский край

## **ПЛЮСЫ И МИНУСЫ МЕТОДОЛОГИИ ИСПОЛЬЗОВАНИЯ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПОДХОДА ДЛЯ РАЗРАБОТКИ И ТЕСТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ**

*Аннотация:* в статье рассмотрена методология использования объектно-ориентированного подхода, выявлены основные различия со структурным подходом. Была проанализирована информация об использовании данного подхода и сделаны выводы об эффективности внедрения и использовании данной методологии. По факту работы были сделаны выводы в виде плюсов и минусов данной методологии.

*Ключевые слова:* объектно-ориентированное программирование, информационная система, структурный подход, декомпозиция, модель.

В современном мире нет предела совершенству. Будь то добавление новых функций в смартфон или же улучшение обороноспособности страны. В виду постоянной гонки за совершенством и опасаясь больших затрат при неудаче на помочь ученым, аналитикам и предпринимателям приходят системы анализа и проектирования систем, будь то информационные или любые другие. В проектировании информационных систем на текущий момент можно выделить два основных подхода: структурный и объектно-ориентированный.

Объектно-ориентированный подход основан на постоянном использовании моделей для разработки программной среды, на основе из ее целей, таких как обслуживание крупных корпоративных клиентов или же работа с клиентами предприятия, обработка информации для правильной работы сложных систем авиации, флота или же многотысячных мероприятий. При объектно-

ориентированном подходе используются формальные конструкции, представляющими их в программной системе. Сам термин и эквивалентные ему понятия начали появляться в 90х годах почти независимо друг от друга в областях связанных с компьютерами, программным обеспечением, базами данных, искусственным интеллектом и программировании. Тестирование представляет из себя процесс поиска ошибок или багов, путем контрольных запусков программы с заранее подготовленными входными данными и сравнения результатов на выходе. Тестирование никогда не гарантирует правильность работы программы, но дает возможность оценить работоспособность и дать оценку правильности.

Гради Буч считал, что открытость и легкость внесения изменений, благодаря устойчивым формам, на которых базируется конструкция объектно-ориентированного подхода, дает системе развиваться и изменяться постепенно и не приводит к полной ее переработке или замене, что гарантирует целостность конечного продукта.

Широкое применение объектно-ориентированного подхода произошло в 1990 году Аланом Кейем при разработке языка SmallTalk в котором использовались только объектно-ориентированные конструкции.

Преимущества объектно-ориентированного подхода:

– объектная декомпозиция дает возможность создавать программные системы меньшего размера путем использования общих механизмов, обеспечивающих необходимую экономию выразительных средств. Использование объектно-ориентированного подхода существенно повышает уровень унификации разработки и пригодность для повторного использования не только программного обеспечения, но и проектов, что в конце концов ведет к сборочному созданию программного обеспечения. Системы зачастую получаются более компактными, чем их не объектно-ориентированные эквиваленты, что означает не только уменьшение объема программного кода, но и удешевление проекта за счет использования предыдущих разработок;

- 
- объектная декомпозиция уменьшает риск создания сложных систем программного обеспечения, так как она предполагает эволюционный путь развития системы на базе относительно небольших подсистем. Процесс интеграции системы растягивается на все время разработки, а не превращается в единовременное событие;
  - объектная модель вполне естественна, поскольку в первую очередь ориентирована на человеческое восприятие мира, а не на компьютерную реализацию;
  - объектная модель позволяет в полной мере использовать выразительные возможности объектных и объектно-ориентированных языков программирования.

Минусы объектно-ориентированного подхода:

- разработка больших систем и их использование требует больших трудозатрат;
- сложность ведения документации, ввиду повышенных трудозатрат;
- сложность иерархии приводит к сложности определения какие поля и классы фактически относятся к данному объекту;
- код обращения очень велик, ввиду чего необходимо использовать несколько методов для его прочтения.

При использовании объектно-ориентированного подхода создаются модели, для воссоздания определенных формальных конструкций. Модель содержит в себе не все свойства и качества основного объекта, а лишь те, которые важны для разрабатываемой информационной системы. Тем самым модель меньше объекта, а значит и проще для восприятия и воссоздания, но при этом в модели сохраняются зависимости и понятия необходимые для разработки. Это упрощает разработку, тестирование и реализацию продукта.

Объектно-ориентированный подход постоянно добавляет специфические сложности в технологию тестирования. Часть системы может выполнять задачи не обращаясь ко всем ресурсам системы, что позволяет зациклывать работу программы и появлению ошибок уже в рабочем продукте. При всем при этом,

тестировать подпрограмму или элемент практически не возможно, ибо важна взаимосвязь элементов в системе, а значит выявление ошибок в подпрограммах в разы сложнее чем в самой программе.

В виду того, что объектно-ориентированный подход не гарантирует правильность работы программы, то тестирование является основным показателем качественности и работоспособности системы в целом. А поскольку изменения в подсистемы можно вносить без прямого воздействия на всю систему в целом, это значит что уровень и частота тестирования должна быть выше, чем в структурно созданных программах. Преимущества объектно-ориентированного подхода в возможности повторного использования, приводит к необходимости более тщательного тестирования. В виду этого одна задача не сможет протестировать всю систему целиком, по этому были введены новые аспекты тестирования.

Инкапсуляция создает проблему видимости данных, так как они доступны только через операции. В процессе тестирования это может создать определенные проблемы с выводом значений.

Наследование ставит вопросы о повторении тестирования наследуемых операций. Допустим операция А принадлежит базовому классу, и она протестирована. Операция В принадлежит производному классу и вызывает операцию А. Должна ли операция А тестироваться повторно?

Полиморфизм приводит к неоднозначности с вызовом операций, которая может быть разрешена только на этапе выполнения. Следовательно заранее построить и спланировать набор тестов становится просто невозможным.

Объектно-ориентированная технология привносит свои особенности в процесс тестирования систем. Формулируется несколько вопросов, которые необходимо разрешить для успешного проведения тестирования:

1. Какая часть унаследованных свойств должна заново тестироваться.
2. Когда и как можно проверять информацию о состоянии класса.
3. Как можно проверить поведение системы, зависящее от состояния, когда отсутствует единый механизм управление состояниями в программе.

---

#### 4. Как следует тестировать интеграцию классов и какие стратегии тестирования применять.

В текущий момент основной проблемой объектно-ориентированного подхода является сложность создания системы и уменьшение быстродействия при ее использовании. С первой проблемой фактически борются пользователи, которые сами или с помощью компаний вносят изменения в существующий программы или же создают интерфейсы для упрощения создания кодов, что ускоряет процесс создания продукта. А с быстродействием борются компании производящие софт. Еще пару лет назад скорость передачи данных превышающая 1МБ или обработка более миллиона операций в секунду считалась рекордной. Сейчас же без проводов, каждый пользователь средненького смартфона имеет доступ к производительным мощностям превосходящие любые, даже самые смелые ожидания десятилетней давности.

Для тех кто уже пользуется или только собирается воспользоваться объектно-ориентированным подходом, стоит учесть все плюсы и минусы этого подхода. А так же подумать о создании информационной системы которая бы удовлетворяла всем имеющимся потребностям. В текущий момент существует десяток постоянно поддерживаемых и обновляющиеся программ для создания информационной системы, что позволит не создавать с нуля, а воспользоваться уже готовыми продуктами, которые за годы на рынке продемонстрировали себя. Решение подобных вопросов выполняется путем разработки новых подходов и модернизации старых, специально для тестирования объектно-ориентированных систем.

В следствии всего вышесказанного можно сделать вывод что использование объектно-ориентированного подхода к разработке информационной системы уменьшает сложность системы, в виду отказа от не важных характеристик, повышает надежность в виду наличия первоначальных взаимосвязей, позволяет вносить изменения и корректиды в отдельные компоненты или подсистемы, не затрагивая остальные части системы и обеспечивает возможность неоднократ-

ного использования отдельных компонентов информационной системы, не обращаясь ко всей системе на прямую.

Постоянное использование объектно-ориентированного подхода позволяет разработать надежные, структурированные, достаточно просто модифицируемые информационные системы. Этим обусловлен высокий интерес программистов и разработчиков к объектно-ориентированному подходу.

### ***Список литературы***

1. Гради Буч. Объектно-ориентированный анализ и проектирование с примерами приложений. – 3 издание, пер с англ. – М.: ООО «И.Д. Вильямс», 2010. – 720 с.
2. Григорьев Л.Ю. Менеджмент по нотам: Технология построения эффективных компаний. – М.: Альбина Паблишерс, 2010. – 692 с.