Лекция 4. Место верификации в жизненном цикле ПО

Хотя общую структуру жизненного цикла произвольной программной системы определить невозможно, существует несколько наиболее часто используемых способов организации различных видов деятельности в рамках жизненного цикла. Их называют моделями жизненного цикла ПО. Чаще всего работы организуются либо в соответствии с каскадной (или водопадной) моделью [1,2] (см. Рис. 1, слева), либо в рамках одной из многочисленных разновидностей итеративной модель жизненного цикла, впервые описанной в 1970 году [2] (Рис. 1, справа).

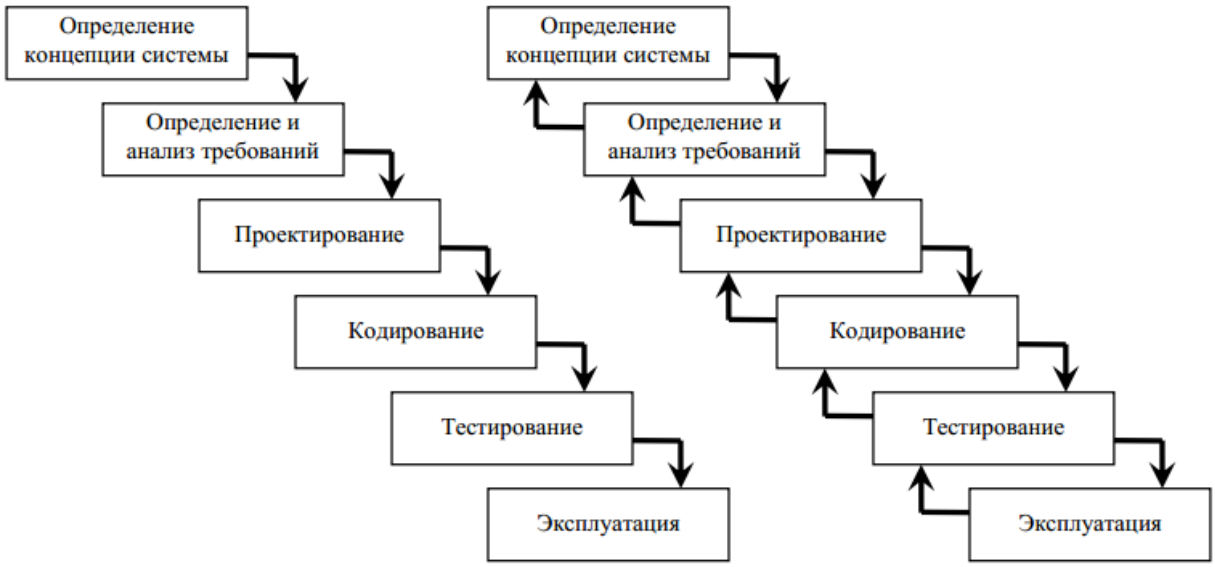

Рисунок 1. Схема каскадной и итеративной модели жизненного цикла ПО

Каскадная модель хорошо работает в тех случаях, когда требования к создаваемой системе удается полностью выявить и зафиксировать в начале проекта (что на практике случается не часто), и результаты всех выполняемых действий проходят тщательный анализ на внутреннюю корректность и соответствие исходным данным. В противном случае обнаруживаемые впоследствии ошибки и недоработки в результатах предыдущих шагов существенно затрудняют продвижение проекта и снижают его управляемость.

Таким образом, в рамках каскадной модели верификация имеет должна выполняться в рамках всех видов деятельности для проверки корректности их результатов, и именно она в первую очередь обеспечивает успешное движение к

конечной цели. Один из видов верификации — тестирование — даже выделяется в отдельный этап проекта.

В рамках итеративной модели отдельные виды деятельности уже не привязаны к этапам проекта и могут выполняться в разнообразных комбинациях. Итеративная модель позволяет быстро реагировать на изменения требований, но требует большего умения от руководителя проекта. В ее рамках различные методы верификации также имеют важнейшее значение, поскольку только с их помощью можно получить оценку качества результатов проекта, как конечных, так и промежуточных. Именно оценка качества служит основной информацией для оценки продвижения к целям проекта, планирования следующих итераций, принятия решений о прекращении проекта или передаче его результатов заказчику.

1. Задачи верификации в рамках жизненного цикла ПО

Все используемые на практике модели жизненного цикла по схеме организации работ являются разновидностями либо каскадной, либо итеративной модели, поэтому независимо от процесса разработки ПО верификация играет в нем ключевую роль, решая следующие задачи.

- Выявление дефектов (ошибок, недоработок, неполноты и пр.) различных артефактов разработки ПО (требований, проектных решений, документации или кода), что позволяет устранять их и поставлять пользователям и заказчикам более правильное и надежное ПО.
- Выявление наиболее критичных и наиболее подверженных ошибкам частей создаваемой или сопровождаемой системы.
- Контроль и оценка качества ПО во всех его аспектах.
- Предоставление всем заинтересованным лицам (руководителям, заказчикам, пользователям и пр.) информации текущем состоянии проекта.
- Предоставление руководству проекта и разработчикам информации для планирования дальнейших работ, а также для принятия решений о продолжении проекта, его прекращении или передаче результатов заказчику.

2. Верификация и другие процессы разработки и сопровождения ПО

Процессом жизненного цикла ПО называется группа видов деятельности, выполняемых для решения определенного набора связанных задач по разработке или сопровождению ПО. На сегодняшний день нет четко определенного общего списка процессов, который не вызывал бы возражений у тех или иных исследователей и практиков.

Международные стандарты ISO 12207 [6], IEEE 1074 [3], ISO 15288 [4],ISO 15504 [5] используют несколько отличающиеся системы процессов. По ISO 12207 к верификации имеют отношение 5 процессов: обеспечение качества (quality assurance), собственно верификация, валидация, совместные экспертизы (joint review) и аудит (audit). Тестирование целиком отнесено к валидации. Кроме того, выделен процесс разрешения проблем (problem resolution), для которого верификация и валидация поставляют входные данные(те самые проблемы).

IEEE 1074 выделяет только один связанный с верификацией процесс — группу деятельностей по оценке (evaluation), которая включает экспертизы (review), аудиты, прослеживание требований и тестирование. Еще несколько видов деятельности, которые можно отнести к верификации и валидации, разбросаны по другим процессам — сбор и анализ метрик, анализ осуществимости, определение потребностей в улучшении ПО, валидация программы обучения.

ISO 15288 считает отдельными процессами управление качеством, оценивание, верификацию и валидацию.

В ISO 15504 (SPICE) в качестве процессов выделены совместные экспертизы и аудиты (один процесс), управление качеством, обеспечение качества и экспертизы (review). Тестирование считается частью других процессов — реализации и интеграции ПО и интеграции программно-аппаратной системы в целом.

С технической точки зрения верификация и валидация являются неотъемлемыми элементами деятельности по обеспечению качества. С одной стороны, эта деятельность должна обеспечить формирование критериев качества, использование при разработке доказавших свою эффективность технологий, определение и контроль процедур выполнения отдельных операций, точность, согласованность и полноту при описании требований, проектных решений,

пользовательской документации, формулировку требований и проектных решений на необходимом уровне абстракции. С другой стороны, в рамках обеспечения качества с помощью верификации и валидации необходимо оценивать текущие характеристики качества ПО и отдельных артефактов процесса разработки и сопоставлять их с критериями и правилами, определенными в рамках системы обеспечения качества проекта и организации в целом.

Деятельность по управлению качеством отличается от обеспечения качества, по-видимому, только большим акцентом на административных процедурах.

Экспертизы и аудит, в свою очередь, являются методами проведения верификации и валидации, такими же, как тестирование, оценка архитектуры на основе сценариев или проверка моделей. В стандартах они рассматриваются как отдельные процессы, скорее всего, потому что применимы к произвольным артефактам жизненного цикла в рамках любого вида деятельности, а также часто используются для оценки процессов и организационных видов деятельности в проекте, в отличие от большинства других методов верификации.

3. Верификация различных артефактов жизненного цикла ПО

Артефакты жизненного цикла ПО можно разделить на технические и организационные. К техническим артефактам относятся описание требований (техническое задание), описание проектных решений (эскизный и технический проекты), исходный код (текст программы), документация пользователей администраторов (рабочая документация), сама работающая система. администраторов (рабочая документация), сама работающая система. Техническими также являются вспомогательные артефакты для проведения верификации и валидации — формальные модели требований и проектных решений, наборы тестов и компоненты тестового окружения, модели поведения реального окружения системы.

Организационными артефактами являются структура работ, разнообразные проектные планы (план-график работ, план конфигурационного управления, план обеспечения качества, план обхода и преодоления рисков, планы проверок и испытаний и пр.),описания системы качества, описания процессов и процедур

выполнения отдельных работ. Верификация может и должна проводиться для всех видов артефактов, создаваемых при разработке и сопровождении программных систем.

1. При верификации организационных документов и процессов проверяется, насколько выбранные формы организации, планы и методы выполнения работ соответствуют задачам, решаемым в рамках проекта, и ограничениям по сроками бюджету, то есть, что с помощью выбранных методов и технологий проект действительно можно выполнить в рамках контракта. Проверяется также, что команда проекта в достаточной степени владеет используемым технологиями разработки, или же что запланированы необходимые мероприятия по обучению. В дальнейшем рассматриваются, в основном, методы верификации, нацеленные на оценку качества технических, а не организационных артефактов процесса разработки.
2. При верификации описания требований одной из первых задач верификации является оценка осуществимости требований с помощью технологий, взятых на вооружение в проекте и в рамках выделенных на проект ресурсов. Проверяются также характеристики требований, указанные в стандартах IEEE 830 [31] и IEEE 1233 [32], а именно следующие.
    - Однозначность. Требования должны однозначно, недвусмысленно выражать нужные ограничения.
    - Непротиворечивость или согласованность. Различные требования не должны противоречить друг другу или основным законам предметной области.
    - Внутренняя полнота. Требования должны описывать поведения системы во всех возможных в контексте ее работы ситуациях. Все значимые законы предметной области и нормы действующих в ней стандартов должны быть учтены в требованиях как ограничения на работу системы.
    - Минимальность. Требования не должны быть сводимы друг к другу на основе формальной логики и основных законов предметной области.
    - Проверяемость. В каждой затрагиваемой требованием ситуации должен быть способ однозначно установить, выполнено оно или нарушено.

- Систематичность. Требования должны быть представлены в рамках единой системы, с четким указанием связей между ними, с уникальными идентификаторами и набором определенных атрибутов: приоритетом, риском внесения изменений, критичностью для пользователей и пр.

Кроме этого, требования должны адекватно и достаточно полно отражать нужды и потребности пользователей и других заинтересованных лиц. Требования должны затрагивать все существенные для пользователей аспекты качества системы: помимо функциональных требований, должны быть адекватно отражены требования к производительности, надежности, удобству использования, переносимости и удобству сопровождения. Для проверки адекватности и полноты отражения реальных потребностей пользователей необходимо проводить валидацию.

3. При верификации проектных решений проверяются следующие свойства.
    - Все проектные решения связаны с требованиями и действительно нацелены на их реализацию. Все требования нашли отражение в проектных решениях.
    - Проектные документы точно и полно формулируют принятые решения, отдельные их элементы не противоречат друг другу.
    - При оформлении проектных документов учтены все правила корректности составления документов такого типа на соответствующих языках. Если используются графические нотации, такие как DFD, ERD или UML, то все диаграммы составлены с соблюдением всех правил и ограничений этих языков.
    - Для проектных решений, связанных с критически важными требованиями к системе, например, по ее безопасности и защищенности, необходимо с помощью максимально строгих методов установить их корректность, т.е. то, что они действительно реализуют соответствующие требования во всех возможных в контексте работы системы ситуациях
4. При верификации исходного кода системы проверяют указанные ниже характеристики.

- Все элементы кода связаны с проектными решениями и требованиями и корректно реализуют соответствующие проектные решения.
- Код написан в соответствии с синтаксическими и семантическими правилами выбранных языков программирования, а также с принятыми в организации и данном проекте стандартами оформления текстов программ (coding rules, coding conventions). Выполнены требования к удобству сопровождения кода, в коде отсутствуют неясные места, все его элементы можно протестировать с помощью сценариев возможной работы системы.
- В исходном коде отсутствуют пути выполнения, достижимые в условиях работы системы и приводящие к ее сбоям, зацикливаниям или тупиковым ситуациям, разрушению процессов и данных проверяемой системы или объемлющей, исключительным ситуациям, непредусмотренным в требованиях и проектных решениях, и пр. Во всех возможных в контексте работы системы сценариях выполнения кода принятые проектные решения и требования соблюдаются, и нет элементов кода, выполняющих непредусмотренные требованиями действия. Эти правила на практике невозможно проверить полностью, но при верификации стремятся как можно более достоверно подтвердить его. При возрастании критичности требований, связанных с компонентами и элементами кода, требуется более строгое подтверждение, и используются более строгие и трудоемкие методы

5. Верификация самой работающей системы или ее компонентов, которые можно выполнять независимо, призвана проверить следующее.
    - Система или ее компоненты действительно способны работать в том кружении, в котором они нужны пользователям, или же в рамках достаточно точной имитации этого окружения.
    - Поведение системы или ее компонентов на возможных сценариях их использования соответствует требованиям по всем измеримым характеристикам. Это, снова, невозможно проверить полностью. Однако, для наиболее критичных требований и сценариев использования применяются более строгие и полные методы

проверки соответствия. Часто проверяется также соответствие поведения системы и ее компонентов реальным нуждам пользователей — это уже является валидацией.

6. При верификации пользовательской документации проверяется следующее.
    - Документация содержит полное, точное и непротиворечивое описание поведения системы.
    - Описанное в документации поведение соответствует реальному поведению системы.

Верификации также должны подвергаться тестовые планы или планы других мероприятий по верификации, а также тесты или материалы, подготовленные для проведения верификации других артефактов, например различные формальные модели. В этих случаях проверяются такие характеристики.

- Подготовленные планы соответствуют основным рискам проекта и уделяют различным его артефактам ровно такое внимание, которое требуется, исходя из их зрелости и важности для проекта.
- Методы верификации, которые планируется применять, действительно способны дать лучшие результаты (с точки зрения обнаружения ошибок и получения достоверных оценок качества, отнесенных к затратам) в намеченных для них областях.
- Подготовленные материалы (тесты, списки возможных ошибок для инспекций, формальные модели требований или окружения системы и пр.) соответствуют контексту использования системы, требованиям к проверяемым артефактам и связанным с ними проектным решениям и могут быть использованы в качестве входных данных для выбранных методов проведения верификации.

Литература

[1] H. D. Benington. Production of Large Computer Programs. Proceedings of the ONR Symposium on Advanced Program Methods for Digital Computers, June 1956, pp. 15-27.

Переиздана в Annals of the History of Computing, October 1983, pp. 350-361.

[2] W. W. Royce. Managing the Development of Large Software Systems. Proceedings of IEEE WESCON, pp. 1-9, August 1970.

Переиздана в Proceedings of the 9th International Software Engineering Conference, Computer Society Press, pp. 328-338, 1987.

[3] IEEE 1074-2006 Standard for Developing a Software Project Life Cycle Processes. IEEE, 2006.

[4] ISO/IEC 15288 Systems engineering — System life cycle processes. Geneva, Switzerland: ISO, 2002.

[5] ISO/IEC 15504-1 Information technology — Process assessment, Part 1: Concepts and vocabulary. Geneva, Switzerland: ISO, 2004.

[6] ISO/IEC 12207 Systems and software engineering — Software life cycle processes.Geneva, Switzerland: ISO, 2008.