

ЖИЗНЕННЫЙ ЦИКЛ ИСХОДНОГО КОДА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Фадеев С.Г.¹

¹ФГБОУ ВПО «ЧГУ им. И.Н. Ульянова», Чебоксары, Россия (428015, Чебоксары, пр-т Московский, д. 15), e-mail: writetofadeev@gmail.com

Рассмотрены стадии жизненного цикла исходного кода по аналогии со стадиями жизненного цикла программного обеспечения: формирование требований, проектирование, реализация, тестирование, внедрения, эксплуатация и сопровождение. Каждая стадия жизненного цикла исходного кода описана в сравнении соответствующей стадией жизненного цикла программного обеспечения. Приведено соотношение времени стадий между собой. В описании стадии формирования требований к исходному коду проведено деление требований на общие и специфические. Общие требования к исходному коду не являются уникальными и поэтому к моменту начала работы над программным проектом являются уже сформированными и для них не требуется отдельная стадия проектирования. Специфические требования к исходному коду меняются от проекта к проекту и для них требуется стадия проектирования. Во многом стадии жизненного цикла исходного кода схожи со стадиями жизненного цикла программного обеспечения, что делает возможным взаимообогащения обеих технологий.

Ключевые слова: исходный код, жизненный цикл, программное обеспечение.

LIFE CYCLE OF SOFTWARE SOURCE CODE

Fadeev S.G.¹

¹Chuvash State University n.a. I.N. Ulyanov, Cheboksary, Russia (428015, Cheboksary, Moscow prospect, 112), e-mail: writetofadeev@gmail.com

We consider the stages of life cycle of source code by analogy with the stages of software life cycle: requirements definition, design, implementation, testing, deployment, operation and maintenance. Each stage of life cycle of source code is described in comparison with the corresponding stage of software life cycle. The time relationship between stages is given. In the description of the stage of requirements definition to source code division of requirements into general and specific is carried out. General requirements to source code are not unique and therefore at the beginning of work on a software project are already formed and they do not require a separate design stage. Specific requirements to source code change from project to project and require the design stage. In many respects the stages of life cycle of source code are similar to the stages of software life cycle, making it possible to mutual enrichments of both technologies.

Keywords: source code, life cycle, software

Программное обеспечение (ПО) – совокупность программ системы обработки информации и программных документов, необходимых для их эксплуатации [1].

Исходный код - компьютерная программа в текстовом виде на каком-либо языке программирования [1].

ПО, как правило, поставляется не в виде исходного, а в виде бинарного кода, являющегося результатом компиляции исходного кода. Таким образом, исходный код является лишь промежуточным продуктом (случаи, когда исходный код является конечным продуктом, в данной статье не рассматриваются). По этой причине исходный код не интересует ни заказчика, ни будущих пользователей ПО, т.к. сам по себе не выполняет для них никаких полезных функций. Полезные функции выполняет ПО, полученное путем компиляции исходного кода.

Одна из моделей жизненного цикла ПО - каскадная модель - рассматривает следующие

стадии, следующие друг за другом:

1. Формирование требований;
2. Проектирование;
3. Реализация;
4. Тестирование;
5. Внедрение;
6. Эксплуатация и сопровождение.

Исходный код, как правило, создается на стадии реализации ПО и продолжает существовать до вывода ПО из эксплуатации. По аналогии со стадиями жизненного цикла ПО можно рассмотреть аналогичные стадии жизненного цикла исходного кода.

Соотношение времени стадий жизненного цикла ПО и стадий его исходного кода приведены на рис. 1:

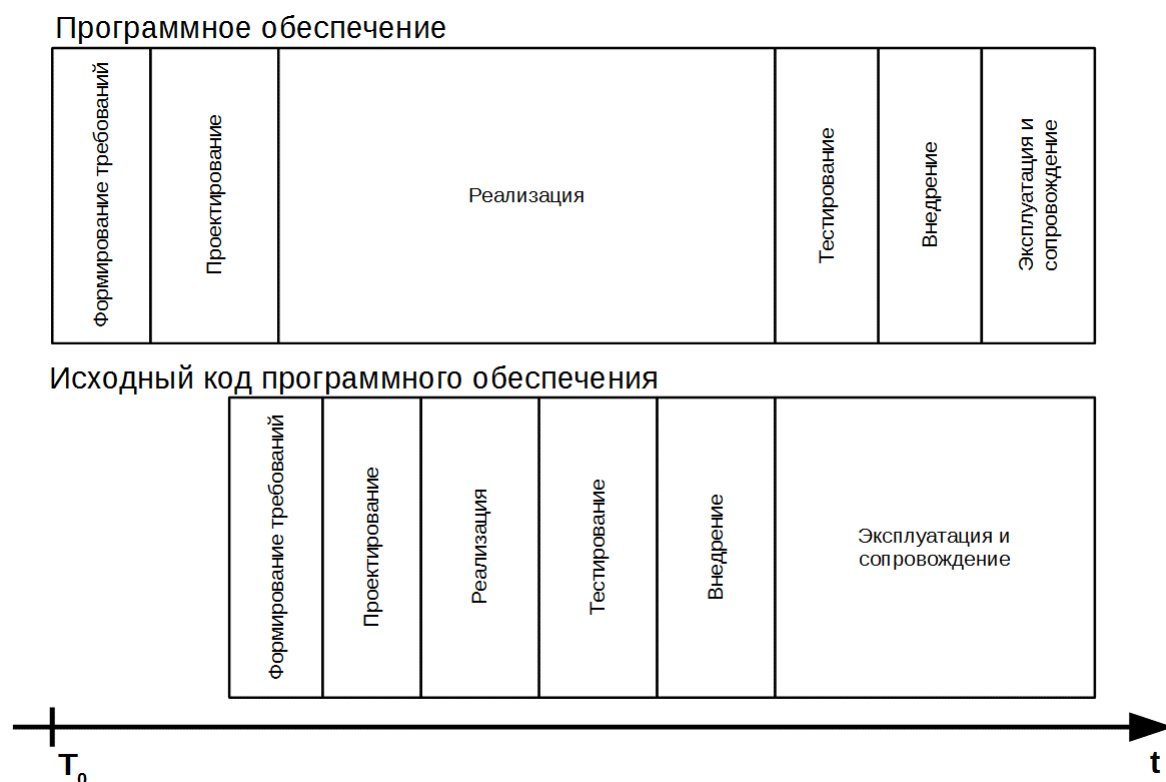


Рис. 1. Соотношение времени стадий жизненного цикла ПО и его исходного кода

Рассмотрим каждую из стадий исходного кода отдельно.

Формирование требований

Хотя исходный код, как было сказано выше, является промежуточным продуктом и сам по себе не интересует конечных пользователей ПО, самым первым и самым важным требованием к нему является соответствие исходного кода требованиям пользователей к ПО. Например, если в требованиях к будущему ПО записано, что время выполнения критической

операции не должно превышать 5 сек, то исходный код должен быть написан так, чтобы на целевой вычислительной системе это требование выполнялось. И если текущий алгоритм этого не обеспечивает, то он должен быть заменен на другой.

Но к исходному коду кроме указанного требования могут быть предъявлены свои, специфические требования. Эти требования являются следствием того, что у исходного кода есть свои непосредственные пользователи - его разработчики. Они и предъявляют к исходному коду дополнительные требования.

Примеры требований к исходному коду:

- правила именования переменных, функций, методов и т.д.;
- стиль отступов;
- ограничение исходного кода по горизонтали;
- запрет использования отдельных небезопасных функций и библиотек.

Требования разработчиков к исходному коду можно условно разделить на две группы:

1. Общие требования.

Общие требования не уникальны по отношению к текущей разработке ПО, а являются результатом опыта и знаний разработчиков, накопленных к ее началу. Время жизни таких требований не ограничено временем жизни разрабатываемого ПО. Общие требования, как правило, находят отражение в корпоративных стандартах оформления исходного кода.

Примеры общих требований:

- правила именования переменных, функций, методов и т.д.;
- стиль отступов;
- ограничение исходного кода по горизонтали;
- запрет использования отдельных небезопасных функций и библиотек.

2. Специфические требования.

Специфические требования уникальны по отношению к текущей разработке ПО, т.к. зависят от структуры конкретного программного проекта, состава разработчиков и т.д., и поэтому формируются под конкретную разработку. Время жизни специфических требований ограничено временем окончания сопровождения ПО.

Примеры специфических требований:

- требования по разбиению исходного кода по файлам и папкам;
- соглашения по составу и именам глобальных переменных.

Стадия формирования требований к исходному коду требуется лишь для специфических требований. Общие требования к исходному коду к этому моменту уже сформированы, как было выше сказано, на основе ранее накопленного опыта и знаний разработчиков. Поэтому, если нет никаких исключительных ситуаций, для общих требований не требуется отдельная

стадия формирования. Но из этого не следует, что общие требования никогда не меняются. Их изменение происходит в ходе работы по мере приобретения разработчиками нового опыта или знакомства с чужим опытом.

Проектирование

Как было сказано выше, общие требования к исходному коду являются результатом ранее накопленного опыта и знаний разработчиков. Поэтому для общих требований не требуется отдельная стадия формирования требований. Соответственно, не требуется и отдельная стадия проектирования для них.

Специфические требования меняются от проекта к проекту, и поэтому им требуется своя стадия проектирования. Сюда относятся такие вопросы, как:

- где будет храниться исходный код
- как он будет разбит по папкам
- и т.д.

Начало стадии проектирования исходного кода может совпадать с началом стадии реализации ПО, но может начинаться и раньше – в конце стадии проектирования ПО. Это зависит от того, насколько детально выполнена работа на стадии проектирования ПО. Если на стадии проектирования ПО решаются такие относящиеся к исходному коду вопросы, как, например, его разбиение по папкам, то начало стадии проектирования исходного кода лежит в стадии проектирования ПО.

Реализация

К стадии реализации относится собственно написание исходного кода.

При создании ПО результатом стадии реализации является «сырой» (еще не проверенный на соответствие требованиям заказчика) бинарный код.

При создании исходного кода результатом является «сырой» исходный код, соответствие которого требованиям еще не проверено. Так же, как «сырой» бинарный, «сырой» исходный код требует последующей проверки на стадии тестирования. В этом и ПО, и исходный код полностью аналогичны, поэтому детально разбирать данную стадию не будем.

Тестирование

Тестирование ПО имеет целью выявить ситуации, когда ПО не соответствует предъявляемым ему требованиям.

Аналогично тестирование исходного кода имеет целью выявить ситуации, когда исходный код не соответствует предъявляемым к нему требованиям, например, требованиям по оформлению, стилю написания, используемым внешним библиотекам, безопасности и т.п.

В настоящее время стадия тестирования исходного кода стала распространенной. Разработано множество инструментов анализа исходного кода, позволяющих

автоматизировать эту стадию. Ниже приведены некоторые из них:

- *PVS-Studio* российской компании "*CuПроВер*" (языки C/C++/C++11) [4];
- *ReSharper* компании *JetBrains* (C#, VB.NET, XAML, ASP.NET, ASP.NET MVC, JavaScript, CSS, HTML и XML) [5];
- *Coverity SAVE* компании *Coverity* (языки C/C++, Java и C#) [2];
- *Klocwork* компании *Klocwork* (языки C/C++, Java и C#) [3].

Кроме этого, исходный код может проверяться и вручную: просмотр кода (*code review*) или инспекция кода (*code inspection*).

Внедрение

На стадии внедрения ПО встраивается в целевую среду применения. Необходимость этой стадии вызвана тем, что ПО разрабатывается в одной среде, а эксплуатироваться будет в другой среде. Смена среды может повлиять на работоспособность разработанного ПО.

В случае исходного кода также возможны ситуации, когда целевая среда не совпадает со средой разработки. Например, когда целевая среда – кроссплатформенная система в исходных текстах, а исходный код ее дополнения разрабатывался только на одной из платформ. В этих случаях исходный код также проходит стадию внедрения. В остальных случаях стадия внедрения исходного кода не требуется.

Эксплуатация и сопровождение

Стадия эксплуатации исходного кода начинается с момента перехода к тестированию ПО в бинарном виде. Предполагается, что к этому моменту работа над исходным кодом должна быть полностью завершена. Ошибки исходного кода, не обнаруженные на стадии его тестирования, но обнаруженные на стадии тестирования бинарного кода, аналогичны ошибкам бинарного кода, не обнаруженным на стадии тестирования бинарного кода, но обнаруженным на стадии его эксплуатации.

Заключение

Исходный код проходит ряд стадий в течение своего жизненного цикла. Во многом эти стадии схожи со стадиями жизненного цикла программного обеспечения, но есть и некоторые различия. На основе сходства стадий возможно взаимообогащение обеих технологий разработки: найденные решения проблем в одной из них могут помочь в решении схожих проблем в другой.

Список литературы

1. ГОСТ Р 54593-2011. Информационные технологии. Свободное программное обеспечение. Общие положения. М.: Стандартинформ, 2011.

2. Coverity SAVE [Электронный ресурс]. — Режим доступа: <http://www.coverity.com/products/coverity-save/> (дата обращения: 31.10.2014).
3. Klocwork Insight [Электронный ресурс]. — Режим доступа: <http://www.klocwork.com/products-services/klocwork> (дата обращения: 31.10.2014).
4. PVS-Studio [Электронный ресурс]. — Режим доступа: <http://www.viva64.com/ru/pvs-studio/> (дата обращения: 31.10.2014).
5. ReSharper [Электронный ресурс]. — Режим доступа: <http://www.jetbrains.com/resharper/> (дата обращения: 31.10.2014).

Рецензенты:

Артемьев И.Т., д.ф.-м.н., профессор кафедры математического и аппаратного обеспечения информационных систем, (ФГБОУ ВПО «ЧГУ им. И.Н.Ульянова»), г. Чебоксары.

Славутский Л.А., д.ф.-м.н., профессор кафедры управления и информатики (ФГБОУ ВПО «ЧГУ им. И.Н.Ульянова»), г. Чебоксары.