МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**Национальный исследовательский ядерный университет «МИФИ»**

 **Институт Интеллектуальных Кибернетических Систем**

**КАФЕДРА КИБЕРНЕТИКИ**

**Направление подготовки 09.04.04 Программная инженерия**

**План разработки ПО**

“Веб ресурс StudMephi”

|  |  |
| --- | --- |
| Выполнили | Андрей Совков  Азат Султанов  Никита Румянцев  Светлана Тихомирова  Анастасия Богатырева  Анна Виноградова  Лидия Антонова  Стас Сегаль  Никита Олейников  Алексей Ильичёв  Александр Слиньков  Полина Горбатенко  Алексей Локтионов  Максим Ключников  Дмитрий Буянов  Дмитрий Сорокин |
| № группы | Б15-506, Б15-503 |
| Руководители | Козырев В.П. |

**Москва 2018**

**ОГЛАВЛЕНИЕ**

[**1. Введение**](#_wbeofeqfim0s) **3**

[1.1. Назначение](#_53871ri6k6li) 3

[1.2. Область применения](#_shoj2j3u5h1u) 3

[1.3. Ссылки](#_ilwdwycrcwse) 3

[1.4. Термины, определения и соглашения](#_26sh8g3371cs) 3

[**2. Организация жизненного цикла ПО**](#_2s2t85ssg0jq) **5**

[2.1. Ответственность](#_eca1tkc7zmwb) 5

[2.2. Взаимодействие процесса разработки с другими процессами ЖЦ ПО](#_9w1s075m9pyj) 5

[**3. Процессы разработки ПО**](#_yn46paj24jin) **6**

[3.1. Разработка требований к ПО](#_bg6xx1yt4st4) 7

[3.1.1. Входные данные процесса разработки требований к ПО](#_1k4qek9n1qcw) 7

[3.1.2. Условия входа в процесс разработки требований к ПО](#_jju2fmjdxacg) 7

[3.1.3 Описание процесса разработки требований к ПО](#_vhww8daba98m) 8

[3.1.4. Условия выхода из процесса разработки требований к ПО](#_resbu9kaejq7) 8

[3.1.5. Выходные данные процесса разработки требований к ПО](#_pvmmnhvc028j) 8

[3.2. Проектирование ПО](#_z3zlp3lfcs5a) 9

[3.2.1. Входные данные процесса проектирования ПО](#_fywmpj8wc1u7) 9

[3.2.2. Условия входа в процесс проектирования ПО](#_ors3ktx9lgdd) 9

[3.2.3. Описание процесса проектирования ПО](#_6kl2hh4tmtde) 9

[3.2.4. Условия выхода из процесса проектирования ПО](#_vi1joa5sdxjg) 10

[3.2.5. Выходные данные процесса проектирования ПО](#_jhoopkuajudx) 10

[3.3. Кодирование ПО](#_tfrznbz83c0o) 10

[3.3.1. Входные данные процесса кодирования ПО](#_or0v3n7ka03y) 10

[3.3.2. Условия входа в процесс кодирования ПО](#_br49x88jnsne) 10

[3.3.3. Описание процесса кодирования ПО](#_hjhmnhb8mqr0) 11

[3.3.4. Условия выхода из процесса кодирования ПО](#_1bbswpw3ppvp) 11

[3.3.5. Выходные данные процесса кодирования ПО](#_n02hm9kw8080) 11

[3.4. Интеграция ПО](#_5bfxlnzej1q) 11

[3.4.1. Входные данные процесса интеграции ПО](#_plsmobg77cmq) 11

[3.4.2. Условия входа в процесс интеграции ПО](#_4uvzx0mfrnt2) 12

[3.4.3. Описание процесса интеграции ПО](#_364maz9lt917) 12

[3.4.4. Условия выхода из процесса интеграции ПО](#_b8z5qmjsfefs) 12

[3.4.5. Выходные данные процесса интеграции ПО](#_d3dpky6gkz6p) 12

[**4. Среда разработки ПО**](#_kbh40y849nvb) **13**

[4.1. Инструменты разработки ПО](#_8tz3sehjuccn) 13

[4.2. Инструменты разработки требований к ПО и описаний проектов](#_u8ny55vs7qbr) 14

[4.3. Языки программирования](#_io3kjz4o529b) 14

# 1. Введение

## 1.1. Назначение

Данный документ определяет мероприятия, которые будут выполняться в рамках настоящего проекта (“Веб-ресурс StudMephi”) для достижения целей процессов разработки ПО, определенных в документе КТ-178B [1].

## 1.2. Область применения

Данный документ применяется исключительно для демонстрации мероприятий, которые могут использоваться при достижении целей процессов, определенных рамками данного проекта (“Веб-ресурс StudMephi”) в соответствии с КТ-178B.

Положения, описанные в данном документе, могут быть неприменимы при выполнении реальных проектов и должны быть пересмотрены с учетом нужд реального проекта в каждом конкретном случае, то есть, в ходе разработки возможны расхождения с регламентом текущего документа.

## 1.3. Ссылки

[1] КТ-178В.

[2] План верификации ПО.

[3] План управления конфигурацией ПО.

[4] План управления требованиями ПО.

[5] Проектный план.

[6] Спецификация требований.

## 1.4. Термины, определения и соглашения

*Таблица 1. Термины, определения и соглашения*

|  |  |
| --- | --- |
| Термин | Определение, толкование |
| Сообщение о проблеме | Документ процесса УК ПО, содержащий описание несоответствия в данных ЖЦ ПО или в описании процессов ЖЦ ПО. |
| Запрос на изменение | Документ процесса УК ПО, созданный для внесения изменений в данные ЖЦ ПО с целью устранения несоответствия, описанного в СП. |
| Артефакт | Атомарное данное ЖЦ ПО. Артефактом может быть, например, требование или файл исходного кода. |
| Агрегация | Составное данное ЖЦ ПО, которое включает в себя однотипные артефакты и/или агрегации. |
| Конфигурация | Составное данное ЖЦ ПО, которое может включать в себя любые артефакты, агрегации и конфигурации. |
| Спецификация программных требований | Совокупность всех требований, предъявляемых к программному обеспечению, и дополнительных сведений, необходимых для правильной интерпретации этих требований. Спецификация требований организована как агрегация в. |
| Формальная инспекция | Способ верификации документов, основанный на экспертной оценке их правильности, выполняемой одним или несколькими инспекторами, как правило, с использованием проверочных перечней. |
| Тип единицы конфигурации | Тип документов ЖЦ ПО, соответствует разделам 11.1 – 11.20 КТ-178В. |
| Базовая версия | Утверждённая зарегистрированная конфигурация одной или более единиц конфигурации, которая в дальнейшем служит основой для последующей разработки и изменяется только в соответствии с процедурами управления изменениями. |
| Проблема | Тип сообщений в системе GitHub «Issue» с возможностью присвоения тегов, связки с версиями данных ЖЦ ПО, назначения ответственных разработчиков и комментирования изменений. |

*Таблица 2. Сокращения*

|  |  |
| --- | --- |
| Термин | Определение, толкование |
| ЖЦ | Жизненный цикл |
| ЗИ | Запрос на изменение |
| ПО | Программное обеспечение |
| СПР | Система поддержки разработки |
| УК | Управление конфигурацией |
| ФИ | Формальная инспекция |
| СП | Сообщение о проблеме |

# 2. Организация жизненного цикла ПО

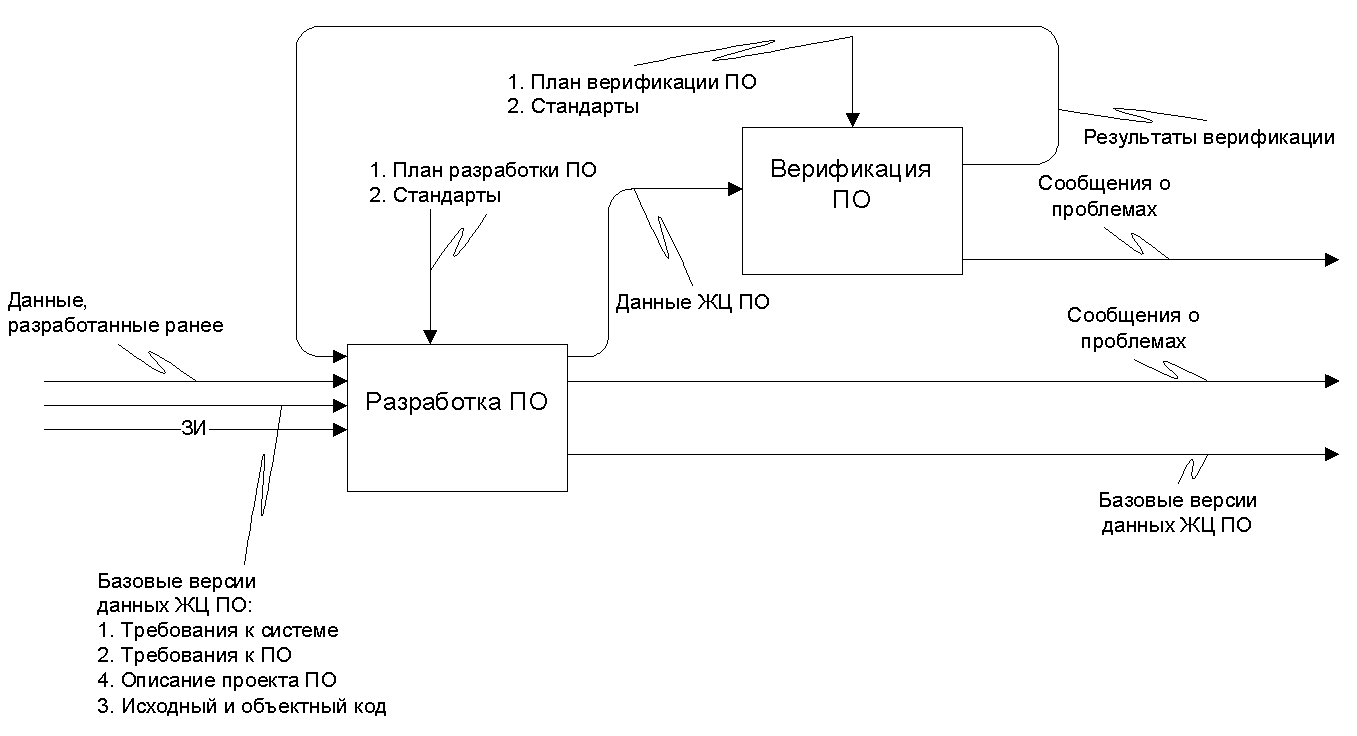
## 2.1. Ответственность

Руководитель проекта ответствен за распределение задач сотрудникам посредством назначения ЗИ ответственным разработчикам. Назначение ЗИ ответственным разработчикам должно проводиться таким образом, чтобы исключить одновременную работу различных сотрудников над одними и теми же частями документов.

Проектная команда, ответственна за выполнение соответствующих мероприятий, приведенных в разделе “Процессы разработки ПО”, в соответствии с положениями данного документа.

## 2.2. Взаимодействие процесса разработки с другими процессами ЖЦ ПО

Схема взаимодействия процесса разработки ПО с другими процессами ЖЦ ПО представлена на рисунке 1. Для облегчения восприятия рисунка на нем не показаны процессы УК ПО (на самом деле все входы и выходы взаимодействующих процессов проходят через процесс УК ПО). Полный перечень входов и выходов процесса верификации ПО также не показан, показаны лишь те, которые имеют непосредственное отношение к описанному в данном документе процессу разработки ПО.



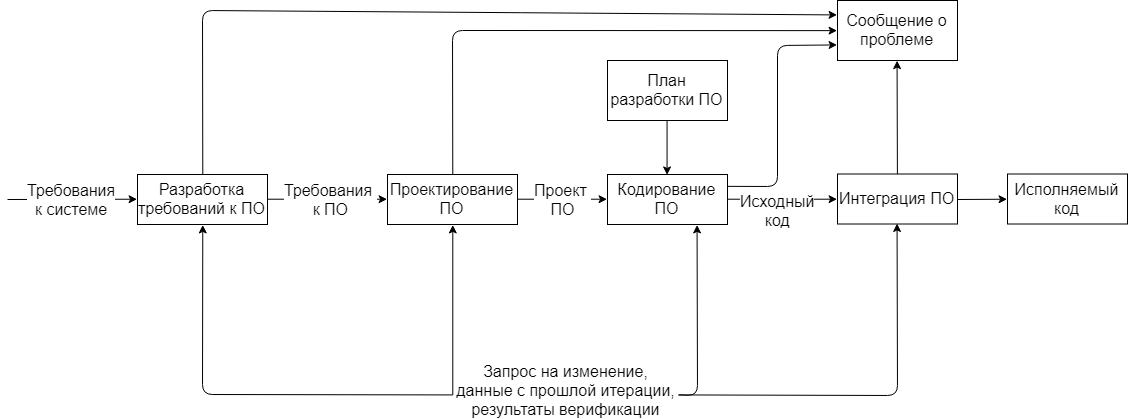
*Рисунок 1. Схема взаимодействия процесса разработки ПО с другими процессами ЖЦ ПО*

Как видно из рисунка, на вход процесса разработки ПО поступают (от процесса УК ПО) базовые версии данных ЖЦ ПО: требования к системе, требования к ПО, описание проекта ПО, исходный код ПО. В результате выполнения мероприятий процесса разработки появляются новые версии указанных данных ЖЦ ПО, которые передаются в процесс верификации ПО. По результатам выполнения процесса верификации в процесс разработки передается результат верификации, который может потребовать дополнительного изменения данных ЖЦ ПО или же, если в процессе верификации не найдено несоответствий, устанавливается новая базовая версия для данных ЖЦ ПО.

Несоответствия, выявленные в результате выполнения мероприятий процесса разработки ПО, фиксируются посредством сообщений о проблемах, передаваемых в процесс УК ПО. Базовые версии данных поступают в процесс УК ПО.

# 3. Процессы разработки ПО

Процесс разработки ПО состоит из процессов, показанных на рисунке 2. Процесс УК ПО, а также управляющие воздействия для процессов не показаны. Также не показан процесс верификации ПО, через который проходят все выходы процессов разработки ПО прежде, чем пойти на вход следующего процесса разработки ПО.



*Рисунок 2. Процессы разработки ПО*

Детальное описание каждого из процессов разработки ПО дано в последующих разделах.

Критерии перехода между процессами разработки представляют собой формальные правила, определяющие возможность начала некоторого процесса или его повторения при наличии необходимых для этого процесса данных. Определение критериев перехода для конкретных процессов ЖЦ ПО приведено ниже при описании этих процессов.

## 3.1. Разработка требований к ПО

Целями процесса разработки требований к ПО являются:

* Разработка требований к ПО высокого уровня;
* Разработка производных требования к ПО высокого уровня.

### 3.1.1. Входные данные процесса разработки требований к ПО

* План разработки ПО (настоящий документ);
* Требования к системе;
* Описания архитектуры системы и ее программно-аппаратных интерфейсов;
* Запрос на изменение;
* Запись о ФИ требований к ПО (при повторных входах в процесс по результатам ФИ).

### 3.1.2. Условия входа в процесс разработки требований к ПО

Вход в процесс разработки требований к ПО осуществляется в случае выполнения любого из приведенных ниже условий:

1. ЗИ требований к ПО назначен ответственному разработчику и помечен тегом «In Progress», при этом установлены базовые версии всех входных данных процесса разработки требований к ПО.
2. в результате формальной инспекции ЗИ требований к ПО обнаружены несоответствия и связанная проблема заново помечена тегом «In Progress» с прикрепленными комментариями о выявленных замечаниях.
3. Если входных данных процесса разработки ещё не существует, должен быть создан ЗИ требований к ПО, помеченный тегами «In Progress» и «Initial», и назначен ответственному разработчику.

### 3.1.3 Описание процесса разработки требований к ПО

При входе в процесс по условию А или С ответственный разработчик должен пометить ЗИ тегом «In Progress» , тем самым извещая руководителя проекта о начале работы по ЗИ, после чего он должен выполнить действия, заданные в ЗИ.

На основании анализа описания ЗИ (при входе в процесс по условию А), записи о ФИ (при входе в процесс по условию Б) и требований к системе разработчик определяет те документы требований к ПО, которые необходимо изменить, создать или удалить.

Если документы требований к ПО уже существуют, то ответственный разработчик вносит изменения в указанные в ЗИ данные. Требования к ПО разрабатываются в соответствии со стандартом на разработку требований к ПО [4] на основе анализа требований к системе.

Если в процессе разработки требований обнаружены несоответствия во входных данных процесса, то ответственный разработчик должен создать сообщение о проблеме.

Для всех требований к ПО, не являющихся производными, необходимо установить трассировочные связи с соответствующими им требованиями к системе. Для производных требований к ПО необходимо установить связи с требованиями, необходимость выполнения которых требует выполнения производных требований.

После завершения работы над требованиями ответственный разработчик должен известить об этом руководителя проекта. Ответственный разработчик должен пометить нужную проблему тегом “Review”. в тоже время запись о проблеме становится и записью в ФИ, работа с которой описана в плане УК ПО.

### 3.1.4. Условия выхода из процесса разработки требований к ПО

Выход из процесса разработки требований к ПО происходит при выполнении всех следующих условий:

* Все найденные во входных данных несоответствия зарегистрированы в сообщениях о проблемах.
* Работы, начатые по ЗИ или по результатам ФИ, завершены и зафиксированы измененные требования к ПО.

### 3.1.5. Выходные данные процесса разработки требований к ПО

* Требования к ПО;
* Сообщения о проблемах.

## 3.2. Проектирование ПО

Целями процесса проектирования ПО являются:

* Определение архитектуры ПО;
* Разработка требований к ПО низкого уровня;
* Разработка производных требования к ПО низкого уровня.

### 3.2.1. Входные данные процесса проектирования ПО

* План разработки ПО (настоящий документ);
* Требования к ПО;
* Запрос на изменение;
* Запись о ФИ описания проекта ПО (при повторных входах в процесс по результатам ФИ);

### 3.2.2. Условия входа в процесс проектирования ПО

Вход в процесс проектирования ПО осуществляется в случае выполнения любого из приведенных ниже условий.

1. ЗИ описания проекта ПО (запись о проблеме на GitHub) назначен ответственному разработчику и помечен тегом «In Progress», при этом установлены базовые версии всех входных данных процесса проектирования ПО.
2. В результате формальной инспекции ЗИ описания проекта ПО обнаружены несоответствия и связанная ФИ помечена тегом «In Progress» с добавлением комментария о необходимости доработки.
3. Если входных данных процесса разработки ещё не существует, должен быть создан ЗИ требований к ПО, помеченный тегами «In Progress» и «Initial», и назначен ответственному разработчику.

### 3.2.3. Описание процесса проектирования ПО

С процедурной точки зрения процесс внесения изменений в файлы описания проекта ПО аналогичен процессу внесения изменений в требования к ПО, описанному в разделе 3.1.3. Отличие заключается в том, что разработчик вносит изменения в описание проекта ПО и использует в качестве входных данных сами требования к ПО, а не требования к системе.

Основным методом при разработке описания проекта ПО будет являться метод структурного проектирования, представляющий собой процесс последовательного разбиения (декомпозиции) ПО на компоненты, а компонента на подкомпоненты и функции.

Проектирование осуществляется с помощью диаграмм в нотации UML 2.5 и инструментального средства Draw.io.

### 3.2.4. Условия выхода из процесса проектирования ПО

Выход из процесса проектирования ПО происходит при выполнении всех следующих условий:

1. Все найденные во входных данных несоответствия зарегистрированы в сообщениях о проблемах.
2. Работы, начатые по ЗИ или по результатам ФИ, завершены и измененное описание проекта ПО зафиксировано в соответствующей проблеме на GitHub.

### 3.2.5. Выходные данные процесса проектирования ПО

* Описание проекта ПО;
* Сообщения о проблемах.

## 3.3. Кодирование ПО

Цель процесса кодирования ПО: разработать исходный код трассируемый, верифицируемый, непротиворечивый и правильно реализующий требования к ПО низкого уровня.

### 3.3.1. Входные данные процесса кодирования ПО

* План разработки ПО (настоящий документ);
* Описание проекта ПО;
* Требования к ПО;
* Запрос на изменение;
* Запись о ФИ исходного кода (при повторных входах в процесс по результатам ФИ);

### 3.3.2. Условия входа в процесс кодирования ПО

Вход в процесс кодирования ПО осуществляется в случае выполнения любого из приведенных ниже условий.

1. ЗИ исходного кода ПО назначен ответственному разработчику и помечен тегом «In Progress», при этом установлены базовые версии всех входных данных процесса кодирования.
2. В результате формальной инспекции ЗИ исходного кода ПО обнаружены несоответствия и связанная ФИ помечена тегом «In Progress» с добавлением комментария о необходимости доработки.
3. Если входных данных процесса разработки ещё не существует, должен быть создан ЗИ требований к ПО, помеченный тегами «In Progress» и «Initial», и назначен ответственному разработчику.

### 3.3.3. Описание процесса кодирования ПО

С процедурной точки зрения процесс внесения изменений в файлы модулей исходного кода аналогичен процессу внесения изменений в требования к ПО, описанному в разделе 3.1.3.

Отличия заключаются в следующем:

1. разработчики вносят изменения в модули исходного кода и используют в качестве входных данных описание проекта ПО и требования к ПО, а не требования к системе;
2. при выполнении мероприятий ответственный разработчик использует инструментарий процесса кодирования (см. раздел 4.1).

### 3.3.4. Условия выхода из процесса кодирования ПО

Выход из процесса кодирования ПО происходит при выполнении всех следующих условий:

1. Все найденные во входных данных несоответствия зарегистрированы в сообщениях о проблемах.
2. Работы, начатые по ЗИ или по результатам ФИ, завершены и измененный исходный код ПО зафиксирован в соответствующей проблеме на GitHub.

### 3.3.5. Выходные данные процесса кодирования ПО

* Исходный код ПО;
* Сообщения о проблемах.

## 3.4. Интеграция ПО

Цель процесса интеграции ПО: скомпилировать исполняемый код ПО и запустить его для интеграции ПО и аппаратных средств.

### 3.4.1. Входные данные процесса интеграции ПО

* План разработки ПО (настоящий документ);
* Описание проекта ПО;
* Исходный код ПО;
* Запрос на изменение;
* Запись о ФИ результатов интеграции ПО (при повторных входах в процесс по результатам ФИ).

### 3.4.2. Условия входа в процесс интеграции ПО

Вход в процесс интеграции ПО осуществляется в случае выполнения любого из приведенных ниже условий.

1. ЗИ для проведения интеграции назначен ответственному разработчику и помечен тегом «In Progress», при этом установлены базовые версии всех входных данных процесса интеграции ПО.
2. В результате формальной инспекции результатов интеграции ПО обнаружены несоответствия и связанная ФИ помечена тегом «In Progress» с добавлением комментария о необходимости доработки.
3. Если входных данных процесса разработки ещё не существует, должен быть создан ЗИ требований к ПО, помеченный тегами «In Progress» и «Initial», и назначен ответственному разработчику.

### 3.4.3. Описание процесса интеграции ПО

Ответственный разработчик извлекает указанную в ЗИ версию исходного кода и выполняет сборку объектного кода, получая из него исполняемый код. Разработчик проверяет, что собранный исполняемый код может быть успешно загружен в целевой вычислитель.

Если в ходе выполнения компиляции, сборки или загрузки возникают проблемы, то разработчик открывает соответствующие сообщения о проблемах, а также фиксирует проблемы на GitHub.

Если компиляция, сборка и загрузка кода прошли успешно, разработчик сохраняет полученные выходные данные интеграции и завершает процесс интеграции по назначенному ЗИ или по результатам ФИ (при входе в процесс по условию Б).

При успешном завершении процесса интеграции, когда в результате этого процесса получены данные для ФИ, автор извещает об этом руководителя проекта способом, аналогичным описанному в 3.1.3.

При неуспешном завершении процесса, когда данные для ФИ не могут быть получены, автор извещает об этом путем создания соответствующего сообщения о проблеме. В этом случае повторный вход в процесс интеграции будет осуществлен по новому ЗИ (с новой версией исходных данных).

### 3.4.4. Условия выхода из процесса интеграции ПО

Выход из процесса происходит при выполнении всех следующих условий:

1. Все найденные во входных данных несоответствия зарегистрированы в сообщениях о проблемах.
2. Работы, начатые по ЗИ или по результатам ФИ (3.5.2), завершены и все выходные данные процесса интеграции ПО зафиксированы в соответствующей карточке.

### 3.4.5. Выходные данные процесса интеграции ПО

* Исполняемый объектный код;
* Файлы, порождаемые компилятором и линковщиком;
* Сообщения о проблемах.

# 4. Среда разработки ПО

Процесс разработки ПО проводится с использованием интегрированной среды разработки Eclipse PDT. Среда разработки устанавливается на рабочих местах (компьютерах) разработчиков и функционирует под управлением ОС Windows. Кроме системы разработчики могут использовать и другие инструменты разработки, являющиеся внешними по отношению к ней. Список программных средств, устанавливаемых на рабочих местах участников практикума, приведен в Таблице 3. Схема организации среды разработки ПО приведена на рисунке 3.

## 4.1. Инструменты разработки ПО

На компьютерах разработчиков ПО устанавливаются следующие инструменты.

*Таблица 3. Программные средства, устанавливаемые на рабочих местах разработчиков ПО*

|  |  |  |
| --- | --- | --- |
| Название | Разработчик | Назначение |
| Windows 7 или более поздние версии | Microsoft Corp. | Операционная система |
| Eclipse PDT | Eclipse Foundation | Интегрированная среда разработки |
| PHP | PHP Project | Язык программирования, используемый для создания серверной части приложения |
| Apache HTTP Server | Apache Foundation | Веб-сервер |
| MySQL | Oracle Corp. | Система управления базами данных |
| phpMyAdmin | phpMyAdmin Developer Team | Веб-интерфейс для администрирования СУБД MySQL |
| Bootstrap | Bootstrap Core Team | Набор инструментов для создания клиентской части приложения на языках HTML и CSS |
| GitHub | Microsoft Corp. | Система поддержки разработки и контроля версий |

## 4.2. Инструменты разработки требований к ПО и описаний проектов

Разработка требований к ПО и описания проекта проводится с помощью систем Microsoft Word и Draw.io. Методы разработки требований и описания проекта, а также правила представления их в системе описаны в стандартах [4] и [5]. Требования к ПО (высокого уровня), описания архитектуры ПО и требования низкого уровня к компонентам ПО размещаются на GitHub. Каждый разработчик имеет доступ ко всем инструментам.

## 4.3. Языки программирования

Основным языком программирования при разработке серверной части является язык PHP, на клиентской части используется язык JavaScript. Разработка исходного кода ПО должна вестись в соответствии со стандартом [6]. При разработке используется версия языка PHP 5.6, и JavaScript ES 6.