**Курсовой проект по курсу**

**«Техгология программирования кибернетических систем»**

**План разработки**

**2016 г.**

Оглавление

[1. Составление технического задания 2](#_Toc469440331)

[1.1. Разработка и утверждение технического задания 2](#_Toc469440332)

[1.1.1. Определение требований к программе 2](#_Toc469440333)

[1.1.2. Определение стадий, этапов и сроков разработки программы и документации на нее 2](#_Toc469440334)

[1.1.3. Выбор языков программирования 3](#_Toc469440335)

[1.1.4. Согласование и утверждение технического задания 3](#_Toc469440336)

[2. Технический проект 3](#_Toc469440337)

[2.1. Разработка технического проекта 3](#_Toc469440338)

[2.1.1. Разработка структуры системы 3](#_Toc469440339)

[2.2. Утверждение технического проекта 3](#_Toc469440340)

[2.2.1. Разработка плана мероприятий по разработке и внедрению программ 3](#_Toc469440341)

[2.2.2. Согласование и утверждение технического проекта 4](#_Toc469440342)

[3. Создание рабочего проекта 4](#_Toc469440343)

[3.1. Разработка программы 4](#_Toc469440344)

[3.1.1. Используемые средства 4](#_Toc469440345)

[Установка 4](#_Toc469440346)

[3.1.2. Организация процесса разработки 5](#_Toc469440347)

[3.1.3. Программирование и отладка программы 5](#_Toc469440348)

[3.2. Разработка программной документации 6](#_Toc469440349)

[3.2.1. Разработка программных документов 6](#_Toc469440350)

[3.3. Тестирование системы 6](#_Toc469440351)

[3.3.1. Разработка, согласование и утверждение программы и методики тестирования 6](#_Toc469440352)

[3.3.2. Проведение тестирования 6](#_Toc469440353)

[3.3.3. Корректировка программы и программной документации по результатам тестирования 6](#_Toc469440354)

[4. Внедрение 7](#_Toc469440355)

[4.1. Подготовка и передача системы 7](#_Toc469440356)

[4.1.1. Подготовка и передача системы и программной документации для сопровождения 7](#_Toc469440357)

[4.1.2. Передача программы заказчику 7](#_Toc469440358)

1. Составление технического задания
   1. Разработка и утверждение технического задания
      1. Определение требований к программе

На начальном этапе необходимо сформировать требования самого высокого уровня, определяющие функциональность и свойства системы в целом. Необходимо, чтобы данные требования однозначно описывали разрабатываемый продукт. Разработка требований этого уровня подразумевает уточнение всех деталей у заказчика и помогает определиться с тем, какой результат нужно получить в итоге. Далее, при необходимости, разрабатываются требования более низких уровней, например, какие средства необходимо использовать при разработке или какие стандарты будут реализованы. Более низких уровней должно быть столько, чтобы каждый участник команды отчетливо понимал свою задачу и то, каким образом ее требуется решить. Эти требования, в отличие от требований верхнего уровня, согласовывать с заказчиком не требуется.

* + 1. Определение стадий, этапов и сроков разработки [программы и документации](javascript:void(11448)) на нее

|  |  |
| --- | --- |
| Стадия | Сроки |
| Изучение предметной области | 24.09.2016 – 01.10.2016 |
| Определение инструментов и технологий, которые будут использованы при разработке | 01.10.2016 – 08.10.2016 |
| Формирование требований и составление ТЗ | 08.10.2016 – 15.10.2016 |
| Верификация требований и ТЗ и исправление ошибок | 15.10.2016 – 22.10.2016 |
| Разработка документации | 15.10.2016 – 05.11.2016 |
| Разработка системы | 15.10.2016 - 05.11.2016 |
| Верификация документации и исправление ошибок | 05.11.2016 - |
| Верификация системы и исправление ошибок | 05.11.2016 - |
| Подготовка системы к передаче заказчику |  |

* + 1. Выбор [языков программирования](javascript:void(23473))

В связи с тем, что разрабатываемая система базируется на системе управления курсами Moodle, в качестве основного будет использован язык PHP 5.6. Поскольку система представляет из себя веб-приложение, для создания и исправления шаблонов будет использован язык HTML 5, а для обеспечения интерактивности и выполнения некоторых функций системы – JavaScript.

* + 1. [Согласование](javascript:void(9745)) и [утверждение](javascript:void(9746)) [технического задания](javascript:void(11959))

После разработки технического задания его необходимо верифицировать на основании плана верификации. После прохождения процедуры верификации техническое задание необходимо утвердить.

1. [Технический проект](javascript:void(18925))
   1. Разработка технического проекта

Данный пункт подразумевает разработку следующих документов: пояснительная записка, описание комплекса аппаратных и программных средств и спецификация оборудования и программных средств.

* + 1. Разработка [структуры](javascript:void(17183)) системы

В рамках разработки структуры системы необходимо выделить ее основные компоненты (подсистемы) и связи между ними, а также состав этих подсистем и выполняемые ими функции. Структура, т.е. подсистемы и связи между ними, должны быть выражены графически в виде диаграмм.

* 1. Утверждение технического проекта
     1. Разработка плана мероприятий по разработке и внедрению программ

Стадия ввода системы в действие включает в себя следующие этапы:

* Подготовка системы к вводу в действие.
* Подготовка существующей системы к объединению с разрабатываемой
* Подготовка стенда, на котором будет установлена система.
* Установка системы на стенд.
* Проведение предварительных испытаний.
* Проведение опытной эксплуатации.

Разработка, внедрение и эксплуатация системы осуществляется проектной документацией, учитывающей конкретные условия функционирования системы.

* + 1. Согласование и утверждение технического проекта

Осуществляется согласование и утверждение следующих документов: технического задания, пояснительной записки, планов верификации, конфигурационного управления и разработки. На рассмотрение, согласование и утверждение представляют документы технического проекта, прошедшие верификацию и отвечающие всем предъявленным к нему требованиям и стандартам.

1. Создание рабочего проекта
   1. Разработка программы

Возникающие задачи разделяются между всеми участниками команды в соответствии с их навыками и возможностями и ставятся с помощью сервиса Trello. Данный процесс описан в плане конфигурационного управления.

* + 1. Используемые средства

Необходимое ПО

* [VirtualBox](https://www.virtualbox.org/wiki/Downloads)
* [vagrant](https://www.vagrantup.com/downloads.html)

# Установка

* Сделать fork репозитория в свой аккаунт;
* Сделать git clone к себе на машину;
* Установить VirtualBox и vagrant;
* В консоли запусить команду

vagrant plugin install vagrant-hostsupdater

* vagrant up в директории с проектом.

Работа

* После запуска vagrant - moodle станет доступен по адресу  
  [http://cspt-moodle.dev](http://cspt-moodle.dev/)
* Конфигурация сервера:

Ubuntu 16.04

PHP 5.6

MariaDB 10.1 (MySQL) (username/password: root/root)

Moodle 3.1 (db: moodle/moodle, admin: admin/test)

* + 1. Организация процесса разработки
* Уведомления о новых задачах или о выполнении задач проводятся с помощью Trello, а также в общем чате вконтакте.
* Задачи может создавать любой участник команды. При постановке задачи создается новая карточка в Trello в списке To Do. В названии карточки необходимо кратко описать задачу. В поле Description изложено более подробное описание задачи(если это требуется). Назначается крайний срок, до наступления которого задача должна быть выполнена. Одновременно заводится аналогичный Isuue в GiHub. Issue необходимо привязать к созданной в Trello карточке используя Addon;
* Если у задачи из списка To Do отсутствует исполнитель, любой участник команды может указать себя в качестве исполнителя задачи. Если задача долго находится в списке To Do без указанного исполнителя - менеджер проекта самостоятельно назначает исполнителя;
* Когда исполнитель начинает работать над задачей, он должен перенести ее из списка To Do в список In Progress. В своем fork’е основного репозитория на аккаунте GitHub необходимо сделать отдельную ветку(от ветки master) со следующим названием: “Issue[Номер Issue на основном репозитории на GitHub]” (например: Issue1);
* Когда исполнитель задачи закончил работу над задачей, он помещает ее в список Testing. В сервисе GitHub необходимо сделать Pull Request своей ветки(напр.: Issue1) в ветку master основного репозитория;
* Группа верификации и остальные разработчики должны выполнить верификацию кода и написать комментарии, если были обнаружены ошибки. Для исправления мелких ошибок исполнитель может обновлять свой Pull Request и отвечать на соответствующие комментарии(напр.: Исправлено). При наличии существенных ошибок или недоработок, карточка задачи переносится обратно в список In Progress. После исправления, разработчик обновляет свой Pull Request и переводит карточку задачи обратно в список Testing;
* Если у группы верификации нет претензий или комментарий к Pull Request’у, то они самостоятельно переводят карточку задачи в список Done и выполняют Merge Pull Request’а.
  + 1. [Программирование](javascript:void(16832)) и [отладка программы](javascript:void(11446))

Написанный программный код должен соответствовать стандартам кода соответствующего языка. Требований к среде разработки не предъявляется.

* 1. Разработка программной документации
     1. Разработка программных документов

Необходимо разработать документацию двух типов: проектную — обзор программного обеспечения, включающий описание рабочей среды и принципов, которые должны быть использованы при создании ПО, и техническую — документацию на [код](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D1%85%D0%BE%D0%B4%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4), [алгоритмы](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC), интерфейсы, [API](https://ru.wikipedia.org/wiki/API) итд. Разработка программных документов ведется в соответствии с календарным планом, приведенным выше, разработанные документы проходят верификацию. Документация должна храниться в общем репозитории для обеспечения ее версионного контроля.

* 1. Тестирование системы
     1. Разработка, согласование и утверждение [программы и методики](javascript:void(11958)) тестирования

Результатом данного пункта является план верификации и тестирования, разрабатываемый на основе взаимодействия группы разработчиков и группы верификации.

* + 1. Проведение тестирования

Проведение тестирования регламентируется пунктами «Проверка нового функционала» и «Функциональное тетирование нового фнкционала» плана верификации и тестирования.

* + 1. [Корректировка программы и программной документации](javascript:void(9688)) по [результатам](javascript:void(24714)) тестирования

Данный процесс связан с процессом, описанным в пункте «Регрессионное тестирование» плана верификации и тестирования. В случае нахождения проблем при проведении регрессионного тестирования команда разработчиков уведомляется об этом и приступает к исправлению кода.

1. [Внедрение](javascript:void(18927))
   1. Подготовка и передача системы
      1. Подготовка и передача системы и [программной документации](javascript:void(10666)) для [сопровождения](javascript:void(11449))

Данный этап подразумевает верификацию системы и всех пунктов документации, а также выполнение действий, необходимых для ее подготовки к внедрению в уже существующую систему.

* + 1. Передача программы заказчику

Заказчику должна быть передана работающая внедренная в использование система, а также описание структуры системы и инструкция по ее использованию