# Отчёт

Контрольные вопросы и ответы:

* 1. **Кратко опишите основные этапы разработки программного продукта**.

1. Разработка алгоритмов – фактически, создание логики работы программы;
2. Написание исходного кода;
3. Компиляция – преобразование в машинный код;
4. Тестирование и отладка – речь, главным образом, о юнит-тестировании.
   1. **Что представляют собой артефакты программного продукта?**

В жизненном цикле разработки программного обеспечения (SDLC) под артефактом обычно понимаются «вещи», которые производятся людьми, вовлеченными в процесс.

* 1. **Опишите основные стратегии разработки ПО.**

1. *Линейная последовательность этапов разработки*– однократный проход (водопадная стратегия)

2. *Инкрементная стратегия*, когда сначала определяются все требования (пользовательские и системные), а затем оставшаяся часть разработки выполняется в виде последовательности версий, первая из которых реализует часть запланированных возможностей, а все последующие версии реализуют дополнительные возможности до тех пор, пока не будет получена полная система.

3. *Эволюционная стратегия*.

При этой стратегии начальный этап не содержит полного объема требования, они уточняются в ходе разработки новых последовательных версий.

* 1. **Кратко охарактеризуйте системы принципов разработки ПО (PSP, TSP, CMM).**

**Индивидуальный процесс разработки** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Personal software process*, *PSP*) — процесс разработки [ПО](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), помогающий разработчикам понимать и улучшать собственную производительность. Создан для применения принципов [модели зрелости процессов](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B7%D1%80%D0%B5%D0%BB%D0%BE%D1%81%D1%82%D0%B8_%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D0%B2) к практике одного разработчика.

TSP это формализованный процесс разработки программного обеспечения. Команда, использующая TSP, обычно состоит из 3-15 человек. Не все участники команды должны быть программистами Большие проекты выполняются несколькими командами, работающими по процессу TSPm

Capability Maturity Model — модель зрелости возможностей создания ПО: эволюционная модель развития способности компании разрабатывать программное обеспечение

* 1. **Состав персонала разработки программного продукта.**

Главным активом компании по разработке ПО является команда. Она должна быть грамотной, квалифицированной и эффективной. Хотя команда создаётся из отдельных людей, нужно работать с коллективом, причем обдуманно и осторожно. К основам формирования коллектива можно отнести: анализ резюме, собеседование с кандидатами и создание необходимых для работы условий.

* 1. **Укажите основные метрики качества программного продукта.**

1 Функциональность.

2 Надежность.

3 Юзабилити (удобство использования).

4 Эффективность.

5 Удобство сопровождения.

6 Портативность.

7 Совместимость

8 Защищенность, т.е. минимизация угроз, связанных с несанкционированным чтением, изменением информации и т. д.

* 1. **В чем заключается процесс инспектирования проекта? Укажите принципы инспектирования.**

Инспектирование программ – это просмотр и проверка программ с целью обнаружения в них ошибок.

Принципы:

1. Вскрытие дефектов.

2. Участие коллег.

3. Распределение ролей. 

4. Тщательная подготовка

* 1. **Приведите примеры функциональных и нефункциональных требований к программному продукту.**

Функциональное требование описывает, *что* должна делать программная система, в то время как нефункциональные требования накладывают ограничения на то, *как* система будет это делать.

Функциональное требование – «Фреймворк Alert PHP должен уметь выводить информацию в браузер»

Нефункциональное требование – «Фреймворк Alert PHP должен выводить информацию в браузер с помощью функции alert из языка js, закрывая тег <?php и открывая тег <script>»

* 1. **Что представляют собой варианты использования программного продукта?**

Варианты использования ПО - это описание последовательности действий, которые может осуществлять система в ответ на внешние воздействия пользователей или других программных систем

* 1. **Приведите схему процесса анализа С-требований.**

**С-требования -** Требования заказчика к системе.

**Порядок анализа требований заказчика включает следующие шаги:**  
**1. Если требование простое и не имеет связей с другими требованиями, его выражают четкими предложениями в соответствующем разделе SRS.**  
**2. Если требование представляет собой взаимодействие пользователя и приложения, то его, как правило, выражают с помощью варианта использования. Для этого:**  
2.1. Задают имена вариантам использования;  
2.2. Определяют действующие лица;  
2.3. Записывают последовательность действий пользователя и приложения;  
2.4. Минимизируют ветвление.  
**3. Если требование затрагивает элементы обработки, каждый из которых получает и выдает данные, используют диаграмму потоков данных (DFD). При этом:**  
3.1. Определяют элементы обработки (обычно высокого уровня);  
3.2. Определяют хранилища данных;  
3.3. Показывают пути передачи данных между обрабатывающими элементами. Указывают типы данных, передаваемых в каждом случае.  
**4. Если требование затрагивает состояния, в которых может находиться программа (или части программы), строят диаграмму состояний в следующей последовательности:**  
4.1. Определяют состояния (каждое состояние обычно определяют отглагольным существительным, например «Ожидание»);  
4.2. Указывают исходное и конечное состояние.  
4.3. Определяют события, происходящие вне рассматриваемой части системы, и приводящие к переходу между состояниями;  
4.4. Определяют вложенные состояния.

* 1. **Приведите схему процесса анализа D-требований.**

 D-требования состоят из полного списка конкретных свойств и функциональности, которую должна иметь программа.

Анализ D-требований

1. Предварительный анализ требования:
   1. Классификация требования как функциональное или нефункциональное (рекомендуется использовать подсказки IEEE SRS для большинства нефункциональных требований);
   2. выбор метода организации функциональных требований.
2. Обеспечение прослеживания требования. Убедиться в возможности прослеживания при проектировании и реализации.
3. Обеспечение тестируемости требования. Спланировать конкретный тест, устанавливающий выполнение требования.
4. Проверка недвусмысленности требования.
5. Назначение требованию приоритет. Например, высокий («важно»), средний («желательно») или низкий («не обязательно»).
6. Проверка полноты требования. Для каждого требования следует убедиться в присутствии всех остальных необходимых или сопутствующих требований.
7. Добавление состояния ошибки:
   1. сформулировать, что требуется выполнить при возникновении нештатных ситуаций;
   2. в критичных местах добавить состояния ошибок программирования.
8. Проверка согласованности. Необходимо убедиться, что ни одно требование не противоречит каким-либо аспектам другого требования.

# Анализ предметной области

**Вывод диалоговых окон с помощью PHP.** Задача вывода диалогового окна с помощью фреймворка Alert PHP заключается в том, чтобы вывести сообщение на экран пользователя с помощью методов классов, которые будет предоставлять фреймворк.

Для реализации задачи будут использованы различные классы, в которых функции вывода сообщений на экран будут стилизованы различным образом. Для того, чтобы вывести что-либо на экран пользователя будут использованы возможности языка программирования JavaScript, теги которого будут открываться прямо в функциях(методах) языка программирования PHP.

**Вывод с помощью класса DefaultAlert.** Данный класс позволит выводить различные сообщения с помощью методов, уже реализованных в языке JavaScript.

Например, для вывода сообщения с помощью alert будет использована функция с одноимённым названием

    public function alert()

    {

        ?>

        <script>

             let txt = "<?php echo $this->txt; ?>";

             alert(txt);

        </script>

        <?php

    }

**Вывод с помощью класса SweetAlert.** Данный класс позволит выводить различные сообщения с помощью методов, уже реализованных в плагине SweetAlert языка JavaScript.

**Вывод с помощью класса BeautifulAlert.** Данный класс позволит выводить различные сообщения с помощью методов, реализацию которых пропишет автор фреймворка. Однако, внешний вид вывода можно будет настраивать и пользователю фреймворка, что не запрещает ему использовать дефолтный.

# 2 Техническое задание

## 2.1 Описание исходных данных приложения

Данный фреймворк может быть использован для создания сайтов на языке PHP; компаниями, предоставляющими услуги по созданию сайтов; фрилансерами и т.д. В системе должна обрабатываться следующая информация:

1. Ajax запросы, созданные с помощью плагина jQuery, посылаемые в тот php-скрипт который будет создан пользователем для обработки запроса или его результата.

2. Данные, вводимые пользователем при помощи функций наподобие prompt

3. Результат функции confirm.

Исходными данными для приложения являются:

1. Сообщения, заглавия, вопросы и т.д., которые необходимо будет передать пользователю.

2. PHP-скрипты, обрабатывающие ajax запросы.

## 2.2 Требования к аппаратным и программным интерфейсам

Необходимо обеспечить программное взаимодействие системы c операционными системами Windows, Linux, MacOS.

Фреймворк должен не иметь сильных утечек памяти, не слишком негативно сказываться на оптимизации.

Модули фреймворка должны занимать также не более 28 Мб памяти на жестком диске.

## 2.3 Требования к пользователям продукта

* Владение языком программирования php на уровне junior разработчика;
* Понимание принципов работы ajax jQuery.
* Знание основ javaScript.

## 2.4 Функции продукта

Основной функционал продукта:

1. Возможность вывода сообщений в браузере.
2. Возможность выводить различные сообщения различными способами.
3. Настройка вывода сообщений.
4. Изменение дизайна сообщений и всего, что они включают в себя.
5. Позволить пользователю ввести данные в «форму».
6. Отправка ответа пользователю в зависимости от его выбора.
7. Позволить пользователю выбрать один из 2 вариантов ответа.
8. Отправить ajax запрос с данными, введёнными пользователем, php скрипту.
9. Отправить ajax запрос с ответом на confirm php скрипту.

Сценарий «отправить пользователю сообщение с input’ом»:

1. Пользователь читает текст сообщения.
2. Вводит данные.
3. Нажимает кнопку «ок» (или подобную).
4. В случае успеха данные отправляются php скрипту

Сценарий «Отправка и обработка данных с помощью ajax и jQuery»:

1. Данные отправляются php скрипту, который должен быть создан тем, кто использует Alert PHP.
2. PHP-скрипт обрабатывает данные так, как это нужно программисту.

Сценарий «Вывод confirm»:

1. Пользователь читает текст сообщения.
2. Делает выбор между true и false;
3. Если включен вывод ответа на выбор пользователя, то пользователь получает ответ;
4. В случае успеха – выбор пользователя отсылается php-скрипту.

Сценарий «Вывод alert»:

Пользователь получает текст

## 2.5 Ограничения

1. Продукт будет поддерживать исключительно английский язык.
2. Продукт не предусматривает автоматического перехода на платформы, не перечисленные в данном документе.