

Спецификация требований программного обеспечения (Software Requirements Specification)

ЧУСИК™:

**Чудо-сборщик информации
Компьютера**

Маслов Роман

Морев Максим

Стасьев Денис

Гришнов Илья

Новиков Юрий

Долгополов Иван

Пахомов Михаил

Структура документа

1. Вступление

- 1.1. Цель
- 1.2. Возможности
- 1.3. Аудитория
- 1.4. Термины
- 1.5. Источники

2. Общее описание

- 2.1. Виденье
- 2.2. Общая функциональность
- 2.3. Классы пользователей
- 2.4. Ограничения

3. Специальные требования

- 3.1. Пользовательский интерфейс
- 3.2. Функциональность системы
- 3.3. Системное тестирование программного продукта

4. Приоритеты работ и план релиза

- 4.1. Приоритеты
- 4.2. Временные рамки разработки

5. Архитектура

- 5.1. Протокол взаимодействия
- 5.2. Формат файла .config
- 5.3. Log файлы

6. Внешние интерфейсы

- 6.1. Программные интерфейсы
- 6.2. Интерфейсы оборудования
- 6.3. UI/UX

1. Вступление

Этот раздел дает краткое описание возможностей всего того, что будет представлено в данном документе.

3.2 Цель

Целью настоящего документа является подробное описание требований для программного обеспечения «Чудо — сборщика информации компьютера» (ЧУСИК™). Документ продемонстрирует функционал и полное описание процесса разработки данного проекта. Так же он объяснит ограничения и возможности графического интерфейса.

1.2 Возможности

«Чудо — сборщик информации компьютера» - это ПО, способное удаленно получать информацию о производительности компьютера, а именно : CPU , оперативная память. Более того, программа способна получать данные сразу с нескольких компьютеров, тем самым собирая данные о производительности машин в локальной сети.

1.3 Аудитория

Данная программа может быть востребована для людей, которые желают посмотреть информацию о производительности отдельного компьютера и/или производительности целой локальной сети компьютеров.

1.4 Термины

Клиент – человек, с машины которого будут сниматься показатели производительности.

Администратор – человек, обладающей утилитой управления.

Утилита управления – часть программы, установленная на компьютере Администратора, и способная задавать настройки для снятия показаний с остальных машин сети.

Агент – часть программы, установленная на машинах клиентов.

Объект – компьютер, с которого снимаются показатели производительности.

1.5 Источники

Интернет и курс “Жизненный цикл ПО”.

2. Общее описание

Этот раздел предоставит общее описание проекта, а именно:

- a) Отобразит основной функционал продукта
- b) Распределит пользователей по взаимодействующим классам
- c) Покажет ограничения при работе
- d) Даст общее видение работы системы

2.1 Видение

Система будет состоять из двух частей:

- a) Утилита управления
- b) Агент

Агент устанавливается на подконтрольные машины, а утилита управления – на компьютер Администратора. Агенты должны периодически собирать информацию о производительности (CPU, RAM) и отправлять её на компьютер Администратора. Передача данных происходит по сети.

2.2 Общая функциональность

Функция 1: Вывод показателей производительности

Требование: Отображаемые значения выводятся в консоль.

Функция 2: Задание фильтров показателей

Требование: В списке фильтров для каждого процесса указывается его название, потребляемые им ресурсы (CPU в % от общей ; RAM в МБ), а также имя пользователя, от имени которого данный процесс запущен.

Функция 3: Возможность изменения периода сбора информации с компьютеров.

Требование: Период измеряется в секундах.

Функция 4: Журнал действий

Требование: История работы с приложением сохраняется в единый текстовый файл.

2.3 Классы пользователей

Существует два класса (типа) пользователей, взаимодействующих с данным программным обеспечением:

Пользователь агента (он же Клиент) – человек, на компьютере которого установлен агент.

Пользователь утилиты управления (он же Администратор) – человек, который имеет утилиту управления, посредством которой он взаимодействует с получаемой информацией о производительности подконтрольных компьютеров. Администратор задаёт настройки частоты приёма (периодичности приёма) данных, может сохранять получаемую информацию, а также менять список компьютеров, с которых производится сбор данных.

2.3 Ограничения

1. Максимальное количество контролируемых компьютеров – 10.

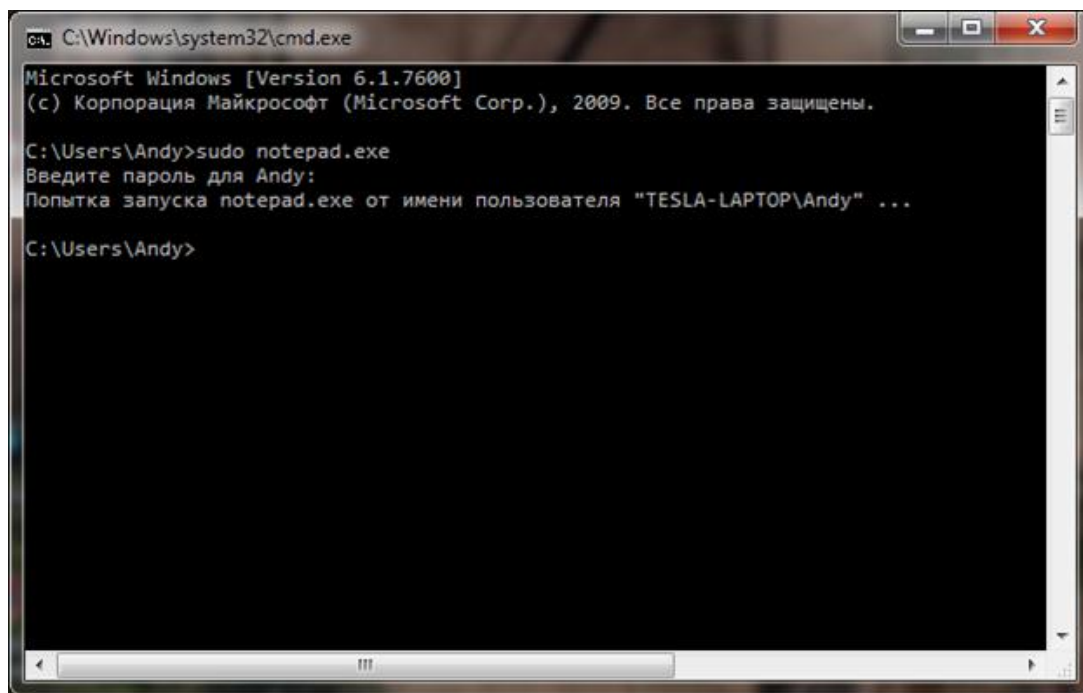
(Чтобы не перегружать сеть)

2. Минимальное количество контролируемых компьютеров – 1;

3. Специальные требования

Этот раздел содержит все специальные функциональные требования, а также требования к пользовательскому интерфейсу программы.

3.1.1 Интерфейс утилиты управления



Интерфейс утилиты управления (на первом этапе) представляет собой консоль Windows.

3.2 Функциональность системы

Этот раздел содержит в себе требования и описания функциональных сторон продукта.

3.2.1 Функциональный блок пользователя агента для контролируемого компьютера

1. Портативность сервиса:

Агент должен уместиться на портативный flash-usb носитель.

2.Изменение частоты посылаемых данных:

Пользователь может изменить периодичность пересылки данных, изменив соответствующий параметр в файле config.

3.2.2 Функциональный блок пользователя утилиты управления

1. Портативность утилиты:

Утилита должна уместиться на портативный flash-usb носитель.

2. Добавление компьютеров в список:

Пользователь может добавить в список один из компьютеров, доступных для работы с ними.

3. Просмотр компьютеров, доступных для трансляции данных:

Пользователь может посмотреть, какие компьютеры готовы к передаче информации.

4. Остановка передачи данных:

Пользователь может остановить трансляцию данных.

5. Вывод передаваемых данных в консоль:

Пользователь утилиты разрешает приложению вывод данных в консоль с указанного компьютера.

6. Изменение параметров выводимых данных:

Пользователь утилиты может изменить выводимую в консоль информацию, изменив соответствующие параметры в файле config.

7. Остановка передачи данных в консоль:

Пользователь утилиты запрещает приложению вывод данных в консоль с указанного компьютера.

8. Удаление компьютера из списка:

Пользователь может удалить ненужный объект из списка компьютеров, после чего возможность вывода данных в консоль прекратится.

3.3 Системное тестирование программного продукта

Необходимо провести тестирование корректности получения и обработки большого числа входных данных, корректности работы графического интерфейса. Отдельным этапом в этом тестировании должна стать проверка времени выполнения, обработки и отображения информации.

4. Приоритеты работ и план релиза

4.1 Приоритеты

- 1) Создание системы удаленного мониторинга производительности компьютеров. То есть создание клиентской и серверной частей только с простейшими функциями.
- 2) Создание возможности вывода списка запущенных на подконтрольных компьютерах процессов, с информацией об именах этих процессов, потребляемыми ими ресурсами (CPU, RAM), а также имени пользователя, от прав которого эти процессы запущены.
- 3) Создания возможности изменения настроек программы в ходе её работы.
- 4) Реализация журнала.
- 5) Написание графического интерфейса.
- 6) Организация базы данных.
- 7) Кроссплатформенность.

4.2 Временные рамки разработки

До конца нынешнего семестра.

5. Архитектура

5.1 Протокол взаимодействия

Приложение будет основано на шаблоне “Издатель-Подписчик” (Данная схема легко масштабируется).

Утилита управления будет рассылать на Агенты информацию (настройки, подписывается / не подписывается, ...), на основе которой Агент будет рассылать своим подписчикам (т.е. Утилите управления) данные. Взаимодействие между Утилитой и Агентом будет осуществляться на базе `printf(“%s”, libraryname1)` в сериализованном виде на базе `printf(“%s”, libraryname2)`. На запросы Утилиты Агент будет возвращать код ошибки, если та будет иметь место.

5.2 Формат файла .config

Задать конфигурацию системы можно как на машинах Клиентов, так и на машине Администратора. На каждой машине будет присутствовать файл .config в формате XML, содержащий поля настроек и некоторую служебную информацию.

Пример для машины Клиента:

```
<settings>
    ...
    <timelimit> 4000 </timelimit>           //Период обновления данных в мс
    ...
</settings>

<subscribe>
    <id> ... </id>                           //Подписчик - машина Администратора
    <where> ... </where>
</subscribe>).
```

Работа программы с файлами .config будет осуществляться с помощью библиотеки libxml2, работа Администратора с файлами .config – с помощью командной строки (а потом и GUI оболочки), работа Клиента с файлами .config не предполагается.

5.3 Log файлы

В программе будет присутствовать журналирование: на машинах Клиентов и машине Администратора будут располагаться файлы log/<DATA>.log. В папке log будут содержаться файлы логов за время N в днях (настраивается в .settings) в формате:

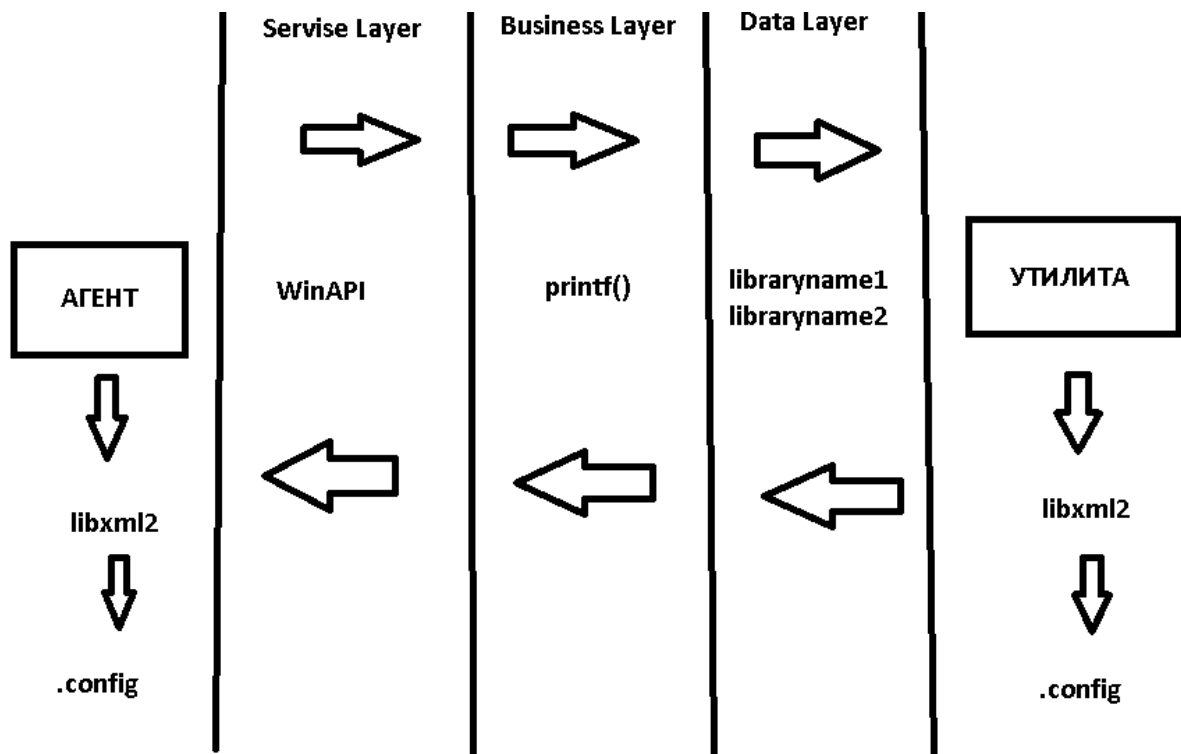
```
[dd.mm.yyhh.mm.ss] <Начало сессии>
...
[dd.mm.yyhh.mm.ss]   Было изменено время ....
[dd.mm.yyhh.mm.ss]   Добавлен новый subscriber_<ID>
...
```


[dd.mm.yyhh.mm.ss] <Конец сессии>

...

Формат файлов логов совпадает на машине Администратора и машинах Клиентов, работа программы с файлами логов на основе самописной библиотеки.

Работа с файлами логов Клиента на машине Клиента – ‘Визуально-Тыкательная’ в блокноте. Работа администратора до появления GUI совпадает со схемой работы Клиента.



Примечания:

Восстановления после сбоя на данном этапе не предусмотрено, обработка ошибок будет осуществляться на базе возвращенного Агентом кода ошибки запроса Утилиты.

Проверки корректности данных на данном этапе не предусмотрено.

На данном этапе приложение будет небезопасно.

На данном этапе не будут осуществлены механизмы транзакции и контроля состояния.

На данном этапе не будет проводиться кэширование и оптимизация процессов.

6. Внешние интерфейсы

Часть информации будет перенесена из раздела Архитектура.

