**Курсов проект**

**по**

**Управление на софтуерни проекти**

**Изработили: Ирен Маринова**

**Виктор Йорданов**

**Валери Вълев  
Дилян Великов**

**Ф.Н.61662247**

**61662182**

**61662192**

**61662180**

**Специалност: СИТ**

**Курс: III**

**Група: 3б**

***Техническо задание на проекта***

# Бюджет

Изчисляването на бюджета се основава на разходи, които трябва да се покрият по време на етапите на разработка на софтуерния проект – първоначалния анализ на изискванията, като за него се отделя повече време с цел изясняване на очакванията и целите на разработвания проект на по-ранен етап, за да се избегнат допълнителни разходи, свързани с човешки ресурси, време и последвали скъпоструващи грешки, изготвяне на структура на проекта, преди започване на конкретната програмна имплементация (избор на подходяща архитектура, на която да се базира добре структуриран код), реализация на софтуера, посредством предходно изяснените спецификации, като този етап включва и почасовия труд на разработчиците и използваните от тях ресурси. Завършителните етапи, свързани с изчисляване на крайната сума на програмния продукт са верификация,валидация и експлоатация на готовия продукт.

Имайки предвид всичко това, сумата за продукта, готов за експлоатация се свежда до **6902** лв, като към нея се добавят и **10%** от общата сума за следващите процеси свързани с поддръжка на софтуера.

# Предмет на техническото задание

Предмет на този документ са изискванията на възложителя относно желаната функционалност на системата. Разработваната система предлага следене на личните/фирмените приходи и разходи,справка за конкретен месец по категории и справка за най-много и най-малко приходи и разходи.В техническото задание се описват пълните и точни първоначални изисвания към проекта, неговите характеристики, както и изпълняваните от него функции и крайната цел и идея на програмния продукт.

**Изисквания на възложителя**

Системата да съдържа екрани със следната функционалност:

**Основен екран**

Да съдържа полета за въвеждане за сума, описание, категория и месец за приход/разход, Бутони за добавяне приход/разход, бутони за справка по конкретен месец за приход/разход и поле в което да бъде изкарван резултатът от справката. При натискане на бутона за добавяне, въведаната информация в полетата сума,описание,категория,месец да се записват във външен файл. При натискане на бутон справка, да се извежда въведаната информация във файла за избрания месец.

# Обхват на разработката

Разработеният софтуерен продукт трябва да удовлетворява очакванията на потребителите и да им предоставя възможност за улеснено използване на заложените функционалности. Потребителите ще имат на разположение продукт, който им позволява да съхранят информацията за личните/фирмените приходи и разходи, като ще могат да правят справки за конкретен месец.

# Изисквания към програмния продукт за следене на личните/фирмените приходи и разходи

## Общи изисквания

Софтуерният продукт за следене на лични/фирмени приходи и разходи трябва да отговаря на следните изисквания. Всички случаи на използване и правилното поведение на системата трябва да са прецизно описани. Необходима е консистентност на информацията, състояща се в непротиворечащи си входни данни. Продуктът е реализиран като приложение за десктоп устройства понеже са най-масово използваните и ще бъде най-удобното устройство за конкретната задача. Програмният продукт е предназначен за устройства, които работят с Windows опрерационна система и поддържащи минималните долупосочени софтуерни и хардуерни изисквания.

Десктоп устройството трябва да притежава следните характеристики:

* **Хардуер**

• Computer and processor: 1 gigahertz (GHz) or faster x86-bit or x64-bit processor with SSE2  
  
• Memory: 1 GB RAM (32-bit); 2 GB RAM (64-bit)  
  
• Hard disk: 3.0 GB of available disk space  
  
• Display: 1024 x 768 screen resolution

* **Операционна** **система**

-Windows 7, Windows 8, Windows 10, Windows Server 2008,Windwos Server 2012

## 4.2. Функционални изисквания – системата за следене на лични/фирмени приходи и разходи трябва да предоставя следните възможности:

## 4.2.1. Модул за въвеждане на информация за приходи/разходи

* Възможност за избиране на сума
* Възможност за избиране на описание
* Възможност за избиране на категория
* Възможност за избиране на месец

### **4.2.2. Модул за справка за приходи/разходи по конкретен месец.**

* Възможност за справа по конкретен месец

# Системна документация

Изпълнителят трябва да подготви следната документация:

* „Описание на структурата и принципа на работа на системата”

Документът е необходимо да бъде на български език.

Документът трябва да бъде предоставен в два екземпляра, на хартиен и електронен носител.

# Софтуерна поддръжка – възможност за разработване на допълнителни модули

Със закупуване на софтуерния продукт възложителят получава право над програмния код, който при желание от негова страна, може да бъде доразработен от изпълнителя, включвайки нови функционалности и допълнителни модули или промяна на съществуващи такива – създаване на нови версии на програмния продукт. За осъществяване на по-нататъшни срещи, свързани с доразвиване на системата, поръчителят може да се свърже с представително лице на посочения имейл – prihodi.razhodi@abv.bg или на мобилен телефон 0889616616.

# Инсталиране и тестване и поддръжка

За инсталиране на приложението е достатъчно да се копира предоставеният набор от файлове.За правилното функциониране на системата се използва предоставеното ръководство за употреба и инсталация. Според подписания с възложителя договор изпълнителят поема отговорност за безплатна поддръжка в рамките на първите 3 месеца. След изтичане на този период поръчителят трябва да заплаща сума от 200 лева месечно за поддръжка. При възникване на аномалии в поведението на програмния продукт, екипът от разработчици ще следи и отстранява грешки.

1. **Система за контрол на версиите**

[Система за контрол на версиите](http://bg.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%B7%D0%B0_%D0%BA%D0%BE%D0%BD%D1%82%D1%80%D0%BE%D0%BB_%D0%BD%D0%B0_%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D0%B8%D1%82%D0%B5) се ползва за записване на промените по даден файл или колекция от файлове. Тя позволява да се запази история на промените, да се върне предишна версия, да се добави описание на промените (като например защо и на какво е извършена промяната и други).

**Git**

За разработването на този софтуерен продукт е използвана децентрализираната система за контрол на версиите – Git. Git е впечатляващо бърза, ефективна и разполага с невероятна branching система за нелинейна разработка, като това е само част от причината тази система да бъде избрана за настоящия проект заедно с останалите, които ще бъдат споменати по-нататък . С Git много потребители могат да работят едновременно по един и същ файл, без това да доведе до бъркотия и хаос в съдържанието на файла. Git е създаден и разработен през април 2005г. от не кой да е, а [Линус Торвалдс](http://bg.wikipedia.org/wiki/%D0%9B%D0%B8%D0%BD%D1%83%D1%81_%D0%A2%D0%BE%D1%80%D0%B2%D0%B0%D0%BB%D0%B4%D1%81) (човекът зад [Linux ядрото](http://bg.wikipedia.org/wiki/%D0%9B%D0%B8%D0%BD%D1%83%D0%BA%D1%81_%28%D1%8F%D0%B4%D1%80%D0%BE%29)). Торвалдс създава Git, за да го ползва за разработката на неговия проект Linux ядро. От тогава до днес Git се развива и подобрява от множество потребители. Предимствата, които предоставя тази система са свързани най-вече със следните характеристики :

* Скорост
* Опростен дизайн
* Мощна поддръжка за нелинейна разработка (хиляди паралелни клонове код)
* Напълно разпределена работа

Друго нещо, което трябва да се спомене е, че Git съхранява и разглежда информацията съвсем различно в сравнение с другите системи, макар потребителският ѝ интерфейс да е подобен с техните. Ето и част от разликите :

**Snapshot-и вместо разлики**

Основната разлика между Git и другите системи за контрол на версиите е начинът, по който Git третира данните. Концептуално, повечето други системи записват информацията като списък от файлово-базирани промени, тоест тези системи виждат информацията, която съхраняват като колекция от файлове и промените направени във файловете във времето. Вместо това, Git възприема своите данни по-скоро като колекция от snapshots (моментни снимки на статуса) на една миниатюрна файлова система. Всеки път когато къмитвате или записвате статуса на вашия проект в Git, системата най-общо казано прави снимка на това как изглеждат файловете ви в този момент и съхранява референция към този snapshot. Това е важна разлика между Git и почти всички други VCS.

**Почти всяка операция е локална**

Повечето операции в Git се нуждаят само от локални файлове и ресурси - общо взето не се нуждаете от информация намираща се в мрежата. Например, за да ви покаже историята на проекта, Git не се нуждае да контактува със сървъра а просто чете директно от локалната си база данни и вие виждате историята почти незабавно. Ако желаете да видите промените в даден файл между текущата му версия и тази отпреди месец, Git ще направи локална калкулация на разликите, вместо да трябва да пита отдалечения сървър за това или да трябва да издърпва по-стара версия от сървъра и едва след това да калкулира разликите локално. Това също значи, че можете да правите почти всичко когато сте офлайн или когато VPN връзката ви не работи например.

**Git само добавя данни**

Когато се прави нещо в Git, тези действия само добавят информация към базата данни на Git. Трудно е системата да направи каквото и да е без то да може да бъде възстановено или пък да се изтрият данни безвъзвратно. Разбира се, подобно на всяка друга VCS, могат да бъдат загубени или объркани промените, които не са били къмитнати, но веднъж направили snapshot-а в Git - е много трудно да се загубят данни, особено пък ако редовно се изпраща базата към отдалечено хранилище.

**Мониторинг на интегритета на данните**

Преди да запазите файл,индексът се определя под формата на контролна сума, изчислена директно от Git. Какво представлява контролната сума? Тази стойност се изчислява, като се използват специални алгоритми и се използва за проверка на целостта на данните по време на съхранение и предаване. Невъзможно е да се промени нищо без знанието на Гит и това е важна част от философията на системата.

**Предимството на Git пред SVN/CVS и TFS**

Използването на Git решава най-големия недостатък на другите две – в случай на повреда в централния или локалния компютър, всички данни на проекта могат да бъдат загубени, тъй като се намират на едно физическо място. Предимството на Git е, че тази система за контрол на версиите е децентрализирана поради, което проектът се сваля и копира локално на компютъра на всеки участващ потребител, като така се създава бекъп. В случай на необходимост, проектът може да се възстанови от локалното копие на някой от потребителите.

**Недостатъци**

* Състои се от много команди и опции и е нужно високо ниво на разбиране, тъй като някои команди и аргументи са непоследователни до някяква степен.
* Не достатъчно добра поддръжка под Windows
* Слаби инструменти за бинарни файлове
* Не поддържа празни директории
* Липса на ограничение за контрол на достъпа

1. **Система за управление на проекта**

Project 2010 представлява софтуерен инструмент за управление и автоматозиране на работата свързана с

управление на проекти. В него може да се въвежда информация за задачите по проекта, тяхното начало

и продължителност на изпълнение, колко време ще отнеме изпълнението им и кой трябва да извърши работата.

**Предимства**:

-Притежава гъвкавост.

-Могат да се интегрират и други инструменти на Microsoft като Outlook, PowerPoint, Access,Excel, Word и други.

-Предлага различни начини за комуникация като Outlook,Skype, Yammer и други. По този начин лесно можеш да се свържеш със своя екип.

**Недостатъци**:

-MS Project е сложна система за начинаещи.Налага се обучение за новите работници по проекта.

-Не е идеален за колаборация между екипа и споделяне на данни.

- Може да се използва само на машини с Windows операционна система.

-Скъпа в сравнение с другите системи.

-Прекалено много неизползващи се функционалности.

-Ако клиент поиска някакви данни, той може да ги достъпи само ако има Microsoft Project.

1. **Методология на разработване**

**Екстремно програмиране, Extreme Programming, XP**

Методологията на разработване, която сме избрали е Екстремно програмиране, тази методология спада към Гъвкавите технологии, които притежават следните пет принципа:

* Реализация на „разрастващо се" (инкрементално) разработване, като изискванията за всяка следваща итерация се преформулират въз основа на оценяване на текущата;
* Регламентиране на стил на работа, основан на компетентност, мотивираност и екипност; стимулиращ творчеството психологически климат;
* Приемане на неизбежността на промените, съпътстващи софтуерните проекти и адаптиране на техниките на разработване към тях;
* Придържане към опростени конструкции и схеми на работа, с цел минимизиране на ресурсите за осъществяването им.
* Активно включване на потребителите в процеса на разработване на Софтуера;

**Други характеристики на този вид технологии са :**

• Гъвкавите технологии не предлагат универсално решение на проблемите в софтуерното производство.

• Те са подходящи, ако в организацията е налице нужният стил на работа, при това за разработка на не „критичен” софтуер.

Причината за избора на тази методология е, че тя е предназначена за малки и средни колективи, разработващи софтуер при неясни или бързо променящи се изисквания, както и заради скоростното получаване на конкретен резултат. Тъй като проектът може да претърпи промени в бъдеще, тази методология е предпочитана пред стандартните методологии, при които цената за реализация на промените е скъпоструваща.

Екстремното програмиране в частност притежава следните характеристики:

* кратки цикли
* метод на инкрементално планиране(стъпка по стъпка)
* непрекъснато тестване като тестовете се задават от потребителя
* непрекъсната интензивна комуникация
* тясно сътрудничество м/у програмистите
* ще могат да променят насоката на проекта по време на реализацията му без това да бъде свързано с големи разходи.

Екип, роли и задачи

1. Екип

* И – Ирена Маринова
* В – Виктор Йорданов
* ВА – Валери Вълев
* Д – Дилян Великов

1. Роли

# Мениджър на проекта – И

* Софтуерен архитект – В
* Разработчици на бизнес логика – И,В,ВА,Д
* Дизайнер на потребителския интерфейс – В
* Технически писател – Д
* Тестери –ВА

1. Начална дата
   1. Проектиране – 4/15/19
   2. Разработване – 4/22/19
   3. Тестване – 4/26/19

Документиране –4/29/19

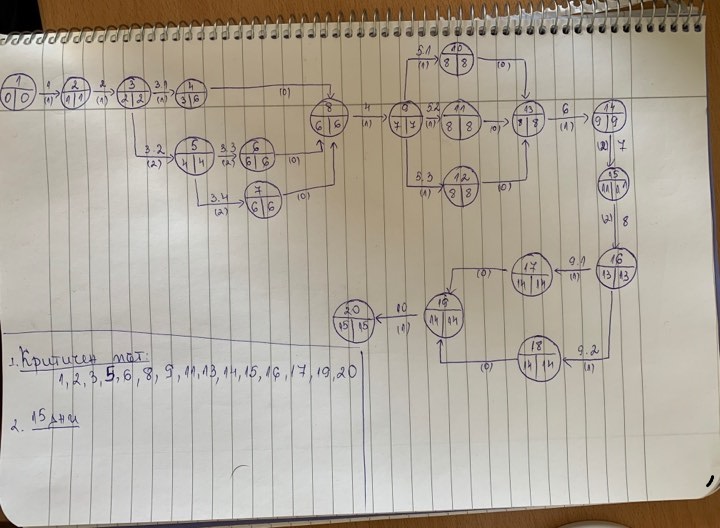
1. Програмни средства за реализация
   1. Език за програмиране – C#
   2. Microsoft Visual Studio
2. Стъпки, зависещи от конкретния подход на разработка

Избраният метод на разработване предполага стъпките на разработване да са инкрементиращи се, всеки от тях да бъде кратък и да се изпълняват стъпка по стъпка, както и непрекъснато тестване, като по време на целия проект за тази методология е важна комуникацията и сътрудничеството между представителите на екипа.

1. Жалонни точки

|  |  |  |
| --- | --- | --- |
| Събитие | Описание |  |
| 1 | Начало на проекта |  |
| 2 | Получаване на разрешение за започване на проекта |  |
| 3 | Съобразяване на изискванията на клиента |  |
| 4 | Избор на подходяща архитектура |  |
| 5 | Планиране на качеството |  |
| 6 | Бизнес логика |  |
| 7 | Първоначален дизайн на интерфейса |  |
| 9 | Създаване на техническото задание на проекта |  |
| 10 | Избор на дизайн |  |
| 11 | Изготвяне |  |
| 12 | Избор от прототипи |  |
| 14 | Изготвяне структурата на програмната система |  |
| 15 | Проектиране |  |
| 16 | Реализация |  |
| 17 | Тестване на бизнес логика |  |
| 18 | Тестване на потребителски интерфейс |  |
| 19 | Създаване на потребителска документация |  |
| 20 | Изпълнение на тестове за демонстрация |  |
| 21 | Пускане в производство на програмния продукт |  |

1. **Критичен път**

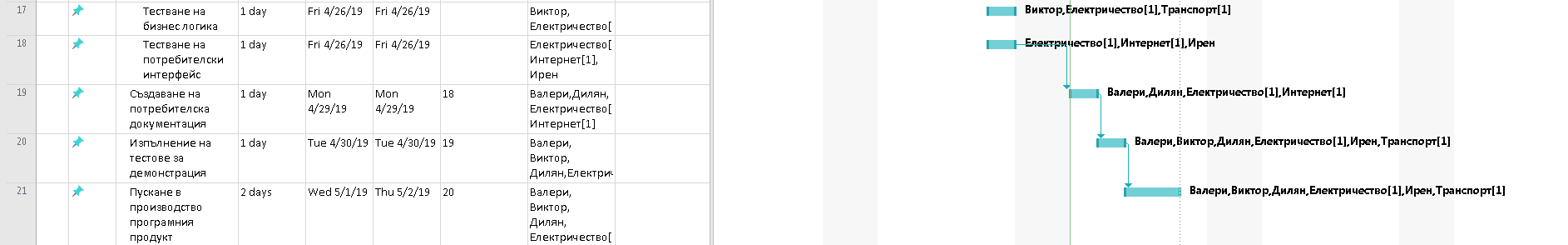
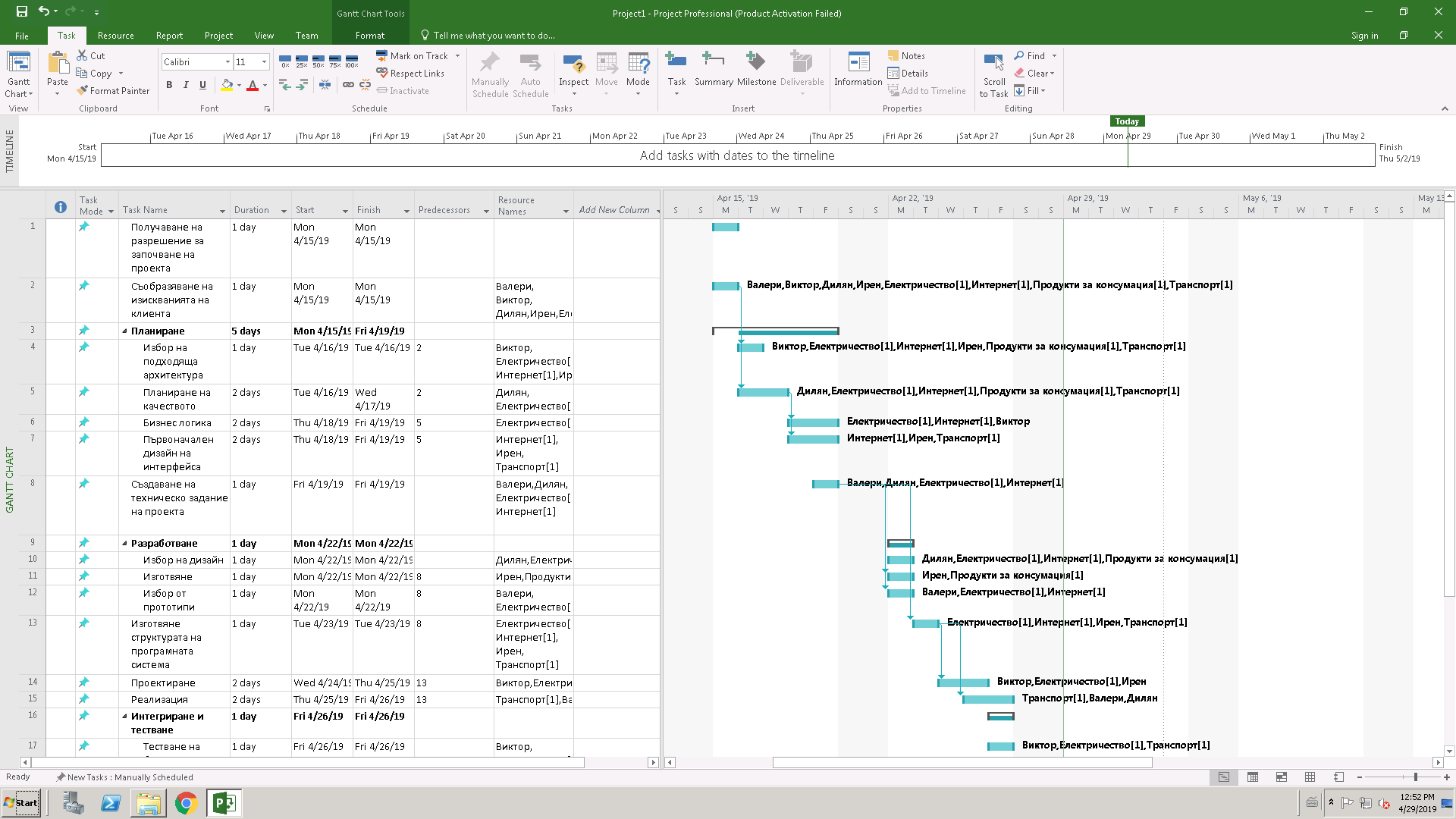
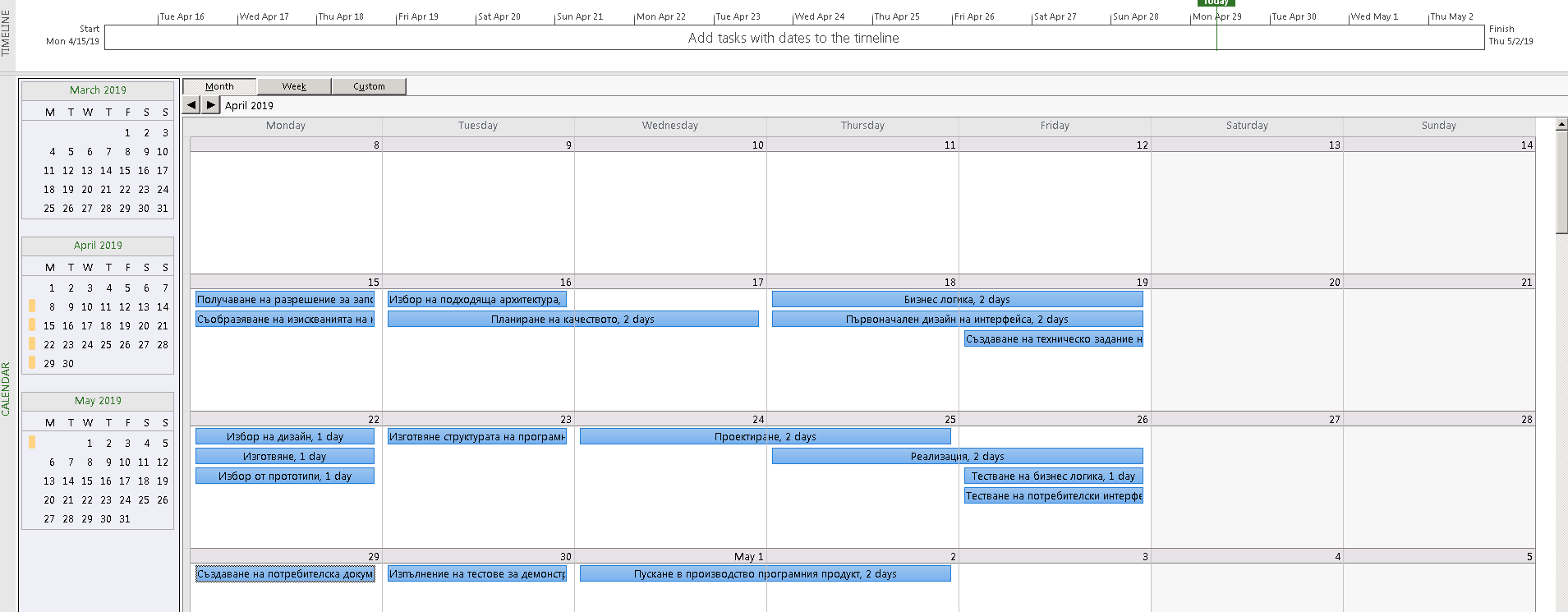


Изпълнението на заложения проект отнема 15 дни.

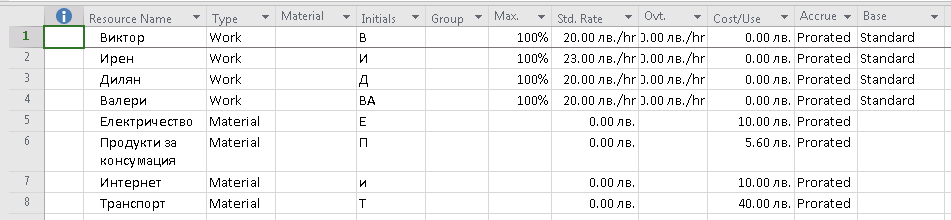
**Microsoft Project**

**Календар**

**Диаграма на Гант**



**Ресурсен лист**



**Диаграма на бюджет**

**Таблица на бюджета**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Actual cost** | **Baseline Cost** | **Cost** | **Cost variance** |
| Съобразяване на изискванията | 664 лв. | 0,00 лв. | 664 лв. | 0,00 лв. |
| Избор на подходяща архитектура | 344 лв. | 0,00 лв. | 344 лв. | 0,00 лв. |
| Планиране на качество | 320 лв. | 0,00 лв. | 320 лв. | 0,00 лв. |
| Бизнес логика | 320 лв. | 0,00 лв. | 320 лв. | 0,00 лв. |
| Първоначален интерфейс на дизайна | 368 лв. | 0,00 лв. | 368 лв. | 0,00 лв. |
| Създаване на техническо задание на проекта | 160 лв. | 0,00 лв. | 160 лв. | 0,00 лв. |
| Избор на дизайн | 160 лв. | 0,00 лв. | 160 лв. | 0,00 лв. |
| Изготвяне | 184 лв. | 0,00 лв. | 184 лв. | 0,00 лв. |
| Избор от прототипи | 160 лв. | 0,00 лв. | 160 лв. | 0,00 лв. |
| Изготвяне на структура на програмната система | 184 лв. | 0,00 лв. | 184 лв. | 0,00 лв. |
| Проектиране | 640 лв. | 0,00 лв. | 640 лв. | 0,00 лв. |
| Реализация | 640 лв. | 0,00 лв. | 640 лв. | 0,00 лв. |
| Тестване на бизнес логика | 184 лв. | 0,00 лв. | 184 лв. | 0,00 лв. |
| Тестване на потребителски интерфейс | 184 лв. | 0,00 лв. | 184 лв. | 0,00 лв. |
| Изпълнение на тестове за документация | 664 лв. | 0,00 лв. | 664 лв. | 0,00 лв. |
| Пускане в производството | 1328 лв. | 0,00 лв. | 1328 лв. | 0,00 лв. |

**Диаграма на ресурсите**

**Таблица на ресурсите**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Cost** | **Baseline Cost** | **Cost Variance** |
| Ирен | 2116,00 лв. | 0,00 лв. | 2116,00 лв. |
| Виктор | 1520,00 лв. | 0,00 лв. | 1520,00 лв. |
| Дилян | 1760,00 лв. | 0,00 лв. | 1760,00 лв. |
| Валери | 1440,00 лв. | 0,00 лв. | 1440,00 лв. |
| Електричество | 10 лв. | 0,00 лв. | 10 лв. |
| Продукти за  консумация | 5,60 лв. | 0,00 лв. | 5,60 лв. |
| Интернет | 10 лв. | 0,00 лв. | 10 лв. |
| Транспорт | 40 лв. | 0,00 лв. | 40 лв. |

**Изводи:**

За достигане на успешно завършен проект, удовлетворяващ потребителските изисквания и влизащ в бюджета и заложеното време за изпълнение, е нужно да се отдели достатъчно време за етапите на анализа на клиентските изисквания и проектирането. Използването на системи за управление на проекта значително повишава производителността на участниците в него. Системата за контрол на версиите дава възможност за независима разработка и лесно синхронизиране на извършената работа, както и създаване на back-up на проекта в хранилището