SOFTWARE PROJECT PLAN

Contents

1. INTRODUCTION

This Software Project Plan is written to specify the project development plan of the path planning project which is named as Canicula Campusus. This project is game for android users. Game's name coming from latin word Canicula which means in English "Dog", and fabrication word Campuses. Our game encapsulate 6 different modes which are chasing cats, daily rutin of a dog, running from cars in a traffic in the campus, fastest arrive of a point, daily challange and multiplayer area wars.

1.1 Scope

This report is written to give general information about the software product development plan. Software development plan is divided into major topics and these topics are classified into sub-topics. Major topics are shown as the following:

- Introduction
- Project plan
- Estimates
- Resources
- Schedule
- Risk

In the Introduction part, the project is explained. This part also gives brief information about the project, its deliverables, epics and non-functional issues such as security, performance, usability and so on.

In the Project plan part contains work breakdown structure. Epics are decomposed into smaller components and these components are assigned a size on scale of tiny, small, medium, large and extra large. These components also include user stories.

In the Project Estimates part gives estimation information about the project in terms of each wbs components.

In the Resources part, team structure is explained. Roles of group members are determined.

In the Schedule part, gantt chart is created with the aid of estimates and resources.

In the Risks part, Possible problems and risks that can occur during the software development process.

1.2 Deliverables

Number	Name of Delivarables	Description of the Deliverables
1	Creating the plan of the game	This plan
2	Graphic Designs	Dog Garphic, Cat Graphic, Human Graphic, Campus Map, Building Graphic, Different Car Garphics, Meal Graphic, Background Graphic
3	Sounds	Dog Sounds, Campus Noise, Game Jingles
4	Scenarios of Chasing Cats	Chasing Cats Mode of the Game
5	Scenarios of Daily Routine	Daily Routine Mode of the Game
6	Scenarios of Running Cars in a Traffic	Running Cars in a Traffic Mode of the Game
7	Scenarios of Fastest Arrive of a Point	Fastest Arrive of a Point Mode of the Game
8	Scenarios of Daily Challange	Daily Challange Mode of the Game
9	Scenarios of Multiplayer Area Wars	Multiplayer Area Wars Mode of the Game
10	Creating Dog Objects and Its Controller	Main character of the game. Players play with dog objects.
11	Unit testing Dog Objects and Its Controller	Testing Dog Objects

12	Creating Cat Objects	Moveable Cat Objects
13	Unit Testing Cat Objects	Testing Cat Objects
14	Creating Human Objects	Moveable Human Objects
15	Unit Testing Human Objects	Testing Human Objects
16	Creating Building Objects	Unmoveable Building Objects
17	Unit Testing Building Objects	Testing Building Objects
18	Creating Car Objects	Moveable Car Objects
19	Unit Testing Car objects	Testing Car Objects
20	Creating Meal Objects	Unmoveable Meal Objects
21	Unit Testing Meal Objects	Testing Meal Objects
22	Creating the Health System for Dogs and Other Objects	Health system for dogs, cats and rival dog
23	Unit Testing Health System for Dogs and Other Objects	Testing the health system
24	Creating the Network Connection for Daily Challange of the Game and Multiplayer Mode	Network is neccessary for both multiplayer mode and daily challange.
25	Creating the Multiplayer System	Player can connect each others game randomly.
26	Integrated Tests for the Game	Integration tests for the game
27	Alpha Tests for the Game	Alpha tests for the game
28	Publishing the Game for Alpha Users	
29	Beta Tests for the Game	
30	Publishing the Game in the Google Play	

1.3 Epics

Number	Essent Tasks	Deliverables' Numbers
1	Planning	1, 4, 5, 6, 7, 8, 9
2	Design	2
3	Resources	3
4	Programming	10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25
5	Tests	26, 27, 28, 29
6	Publishing	30

1.4 Non-Functional Issues

1. Performance of the game according to hardware of the smart phone
2. Changing interface of the game according to screen size of the smart phone
3. Network connection speed of the Multiplayer Mode
4. Security of the Multiplayer Mode
5. Security of the Daily Challange

2. PROJECT PLAN

2.1 Stages of Project

Our first stage, will be planing our game design which is consist of four parts such as characters, gameplay, story and game goals.

In our next step, project members will start learning necessary tools for defined platforms which are Android and Unity 3D and usage of C#.

In the next step, audio and graphic resources will be obtained. Appropriate sounds and musics will be choosen according to pre-defined gameplay.

In design phase, map and gameplay will be uniquely designed according to each level.

In our next phase, player, AI, gameplay constraints and level unlock system will be developed.

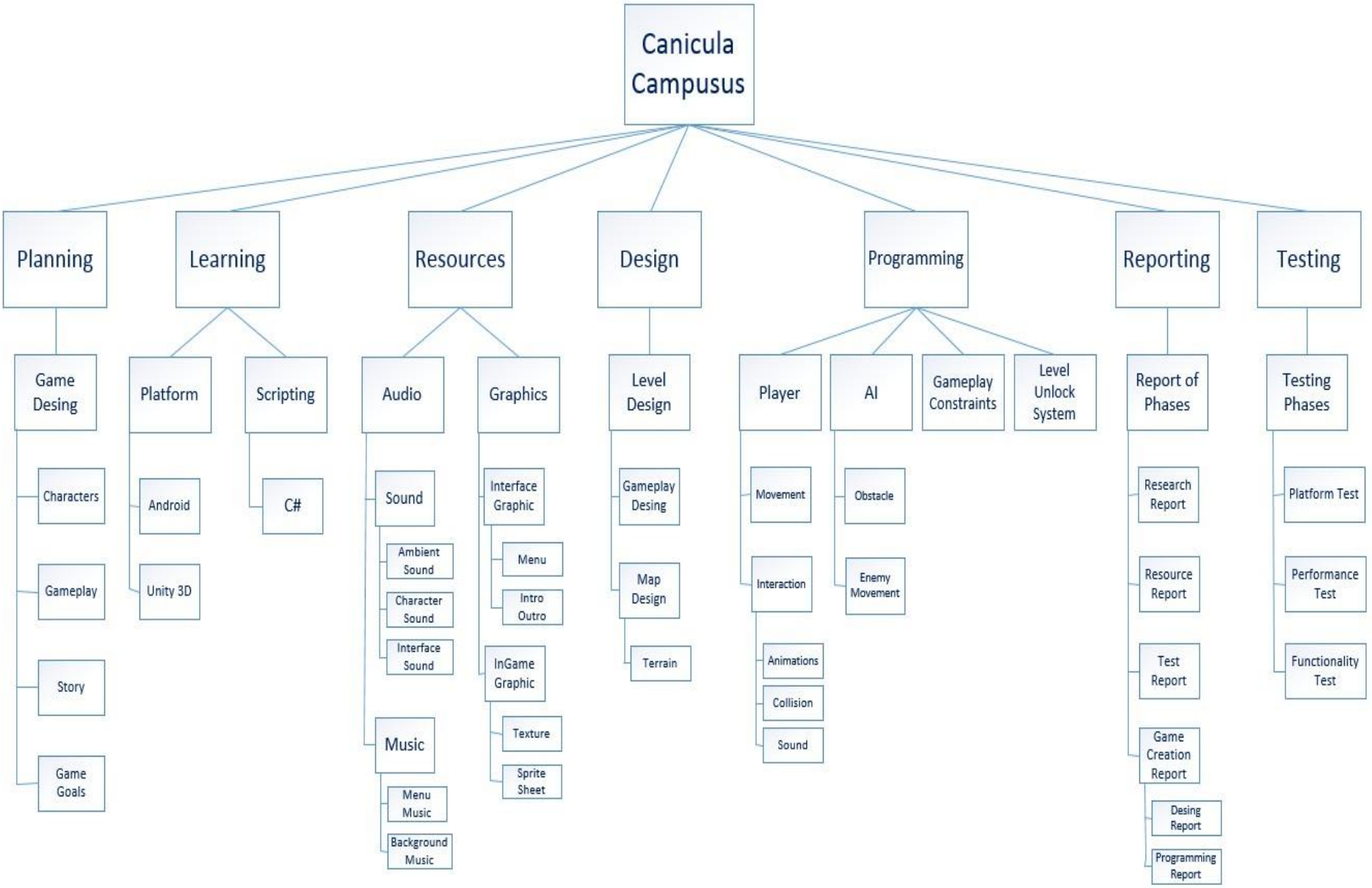
Our reporting phase will continue through the whole project process. It contains four stages such as research, resource, test, game creation reports.

Last but not least, in our testing stage, platform test, performance test and functionality test will be performed.

2.2 Work Breakdown Structure (WBS)

A work breakdown structure (WBS), in project management and systems engineering, is a deliverable-oriented decomposition of a project into smaller components.

The WBS is shown as the following:


Canicula Campusus
Planning
Learning
Resources
Design
Programming
Reporting
Testing
Game Desing
Characters
Gameplay
Story
Game Goals
Platform
Android
Unity 3D
Scripting
C#
Audio
Sound
Ambient Sound
Character Sound
Interface Sound
Music
Menu Music
Background Music
Graphics
Interface Graphic
Menu
Intro Outro
InGame Graphic
Texture
Sprite Sheet
Level Design
Gameplay Desing
Map Design
Terrain
Player
Movement
Interaction
Animations
Collision
Sound
AI
Obstacle
Enemy Movement
Gameplay Constraints
Level Unlock System
Report of Phases
Research Report
Resource Report
Test Report
Game Creation Report
Desing Report
Programming Report
Testing Phases
Platform Test
Performance Test
Functionality Test

3. PROJECT ESTIMATE

Canicula Campusus game is an Android phone application. Our game encapsulate 6 different modes:

- Chasing Cats
- Daily Routine of a dog
- Running from cars in a traffic in the campus
- Fastest arrive of a point
- Daily challenge
- Multiplayer area wars

3.1 Estimation Techniques

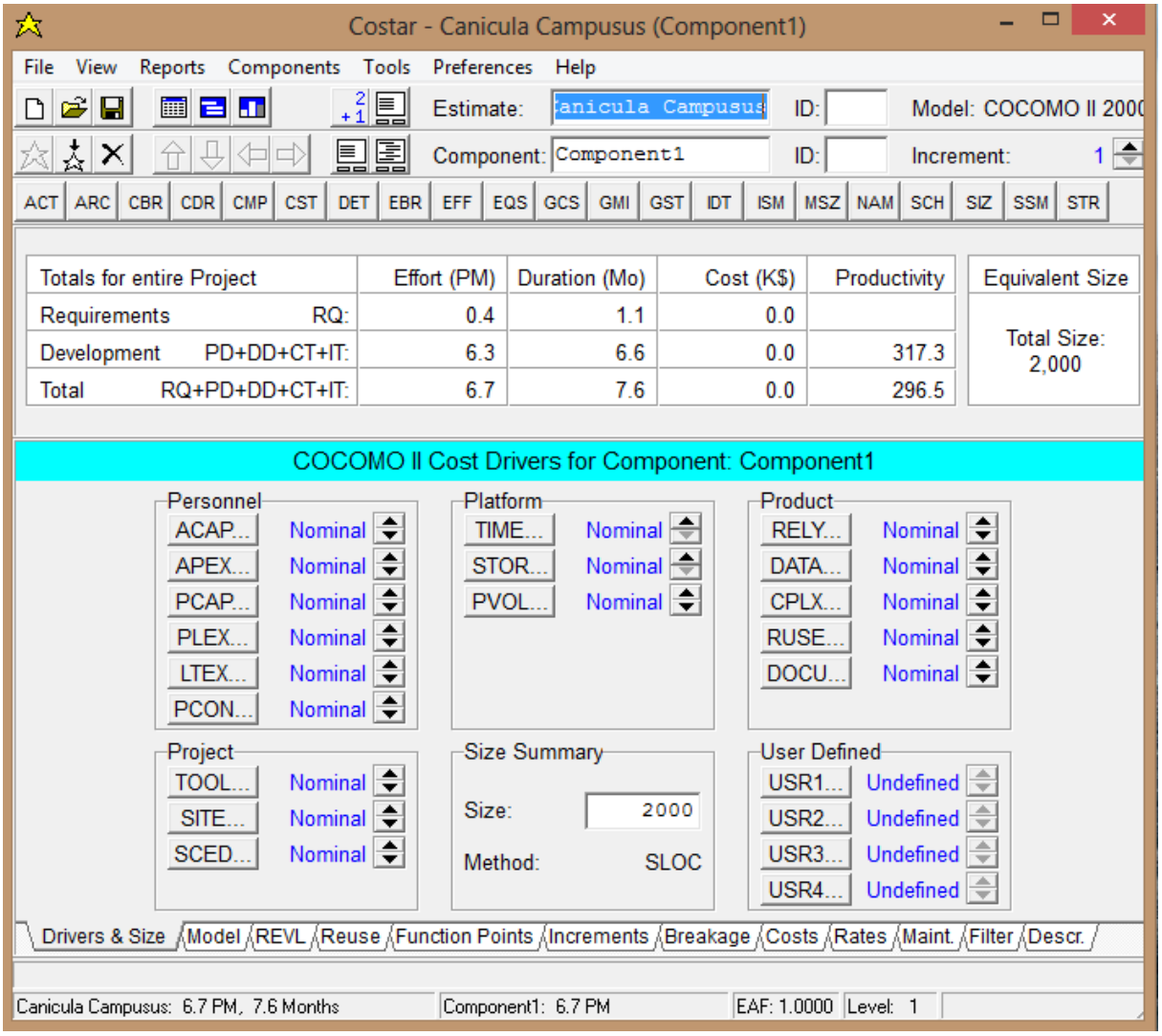
3.1.a Cocomo Techniques

The Constructive Cost Model (COCOMO) is an algorithmic software cost estimation model developed by Barry W. Boehm. The model uses a basic regression formula with parameters that are derived from historical project data and current as well as future project characteristics. **CO**nstructive **CO**st **MO**del II (COCOMO II) is a model that allows one to estimate the cost, effort, and schedule when planning a new software development activity. COCOMO® II is the latest major extension to the original COCOMO (COCOMO 81) model published in 1981. It consists of three submodels, each one offering increased fidelity the further along one is in the project planning and design process. Listed in increasing fidelity, these submodels are called the Applications Composition, Early Design, and Post-architecture models.

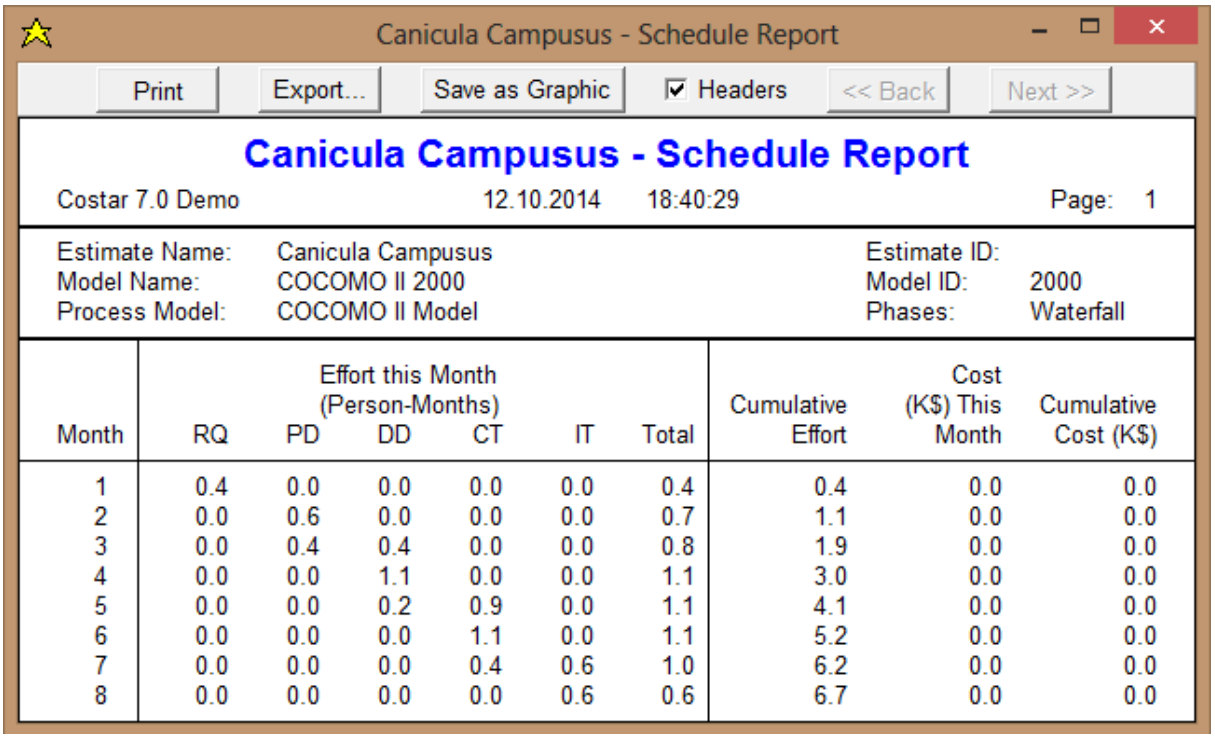
COCOMO II can be used for the following major decision situations

- Making investment or other financial decisions involving a software development effort
- Setting project budgets and schedules as a basis for planning and control
- Deciding on or negotiating tradeoffs among software cost, schedule, functionality, performance or quality factors
- Making software cost and schedule risk management decisions
- Deciding which parts of a software system to develop, reuse, lease, or purchase
- Making legacy software inventory decisions: what parts to modify, phase out, outsource, etc

- Setting mixed investment strategies to improve organization's software capability, via reuse, tools, process maturity, outsourcing, etc
- Deciding how to implement a process improvement strategy, such as that provided in the SEI CMM



Cocomo Output



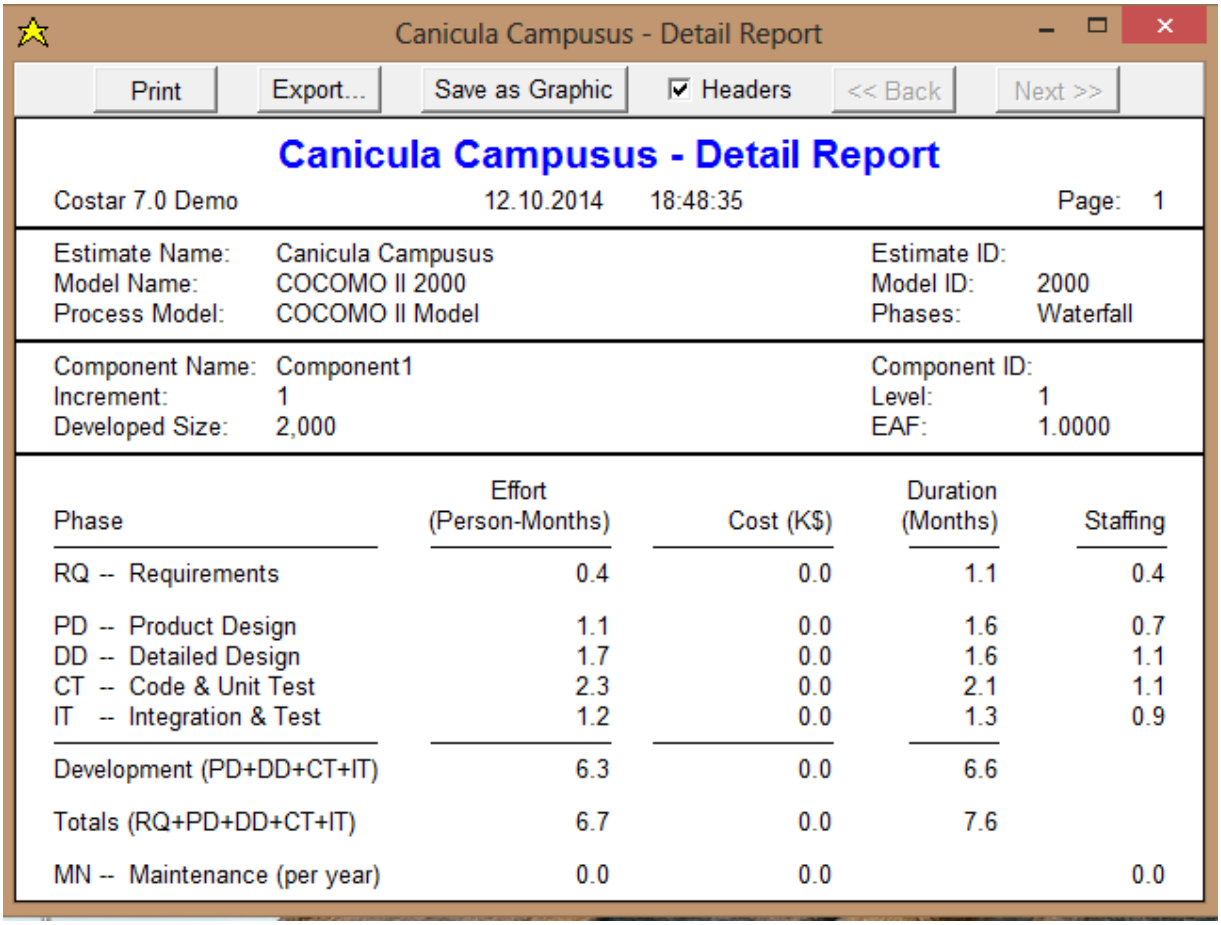
Canicula Campusus - Schedule Report

Costar 7.0 Demo 12.10.2014 18:40:29 Page: 1

Estimate Name: Canicula Campusus
Model Name: COCOMO II 2000
Process Model: COCOMO II Model
Estimate ID:
Model ID: 2000
Phases: Waterfall

Month	Effort this Month (Person-Months) RQ	PD	DD	CT	IT	Total	Cumulative Effort	Cost (K$) This Month	Cumulative Cost (K$)
1	0.4	0.0	0.0	0.0	0.0	0.4	0.4	0.0	0.0
2	0.0	0.6	0.0	0.0	0.0	0.7	1.1	0.0	0.0
3	0.0	0.4	0.4	0.0	0.0	0.8	1.9	0.0	0.0
4	0.0	0.0	1.1	0.0	0.0	1.1	3.0	0.0	0.0
5	0.0	0.0	0.2	0.9	0.0	1.1	4.1	0.0	0.0
6	0.0	0.0	0.0	1.1	0.0	1.1	5.2	0.0	0.0
7	0.0	0.0	0.0	0.4	0.6	1.0	6.2	0.0	0.0
8	0.0	0.0	0.0	0.0	0.6	0.6	6.7	0.0	0.0

Schedule Report



Canicula Campusus - Detail Report

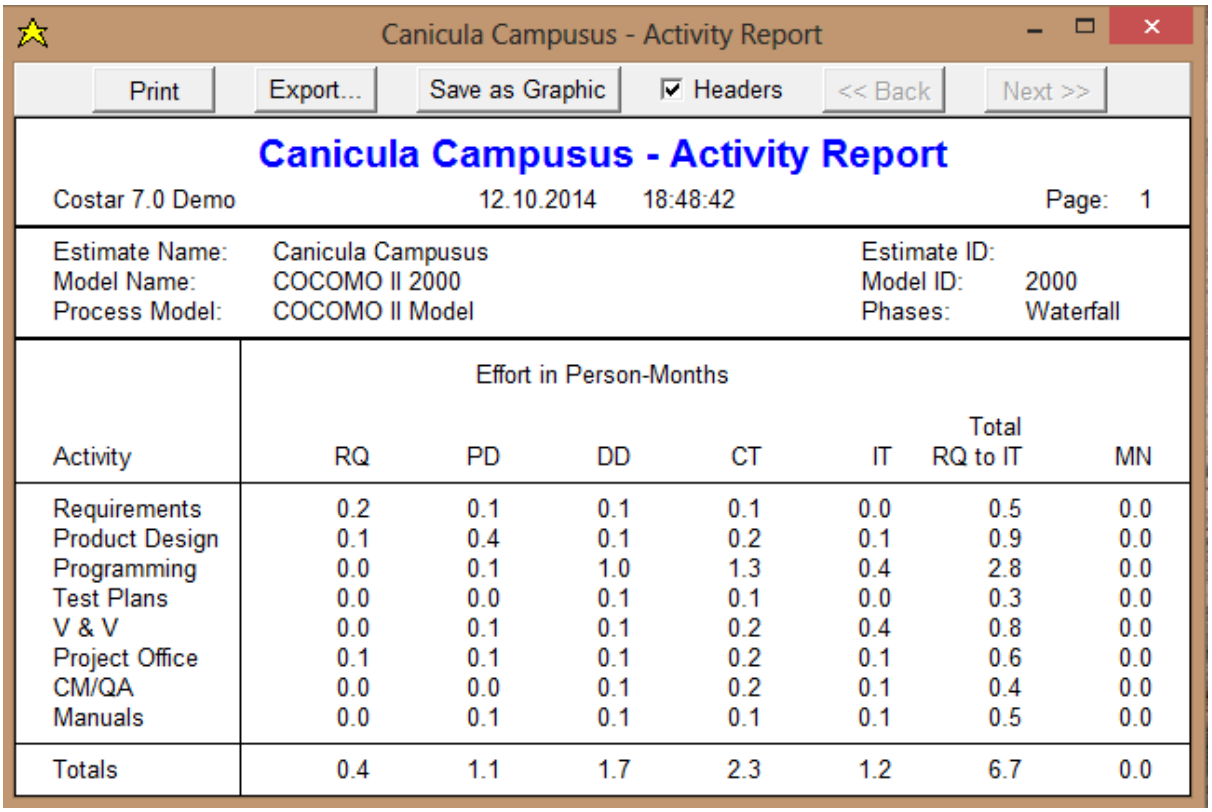
Costar 7.0 Demo 12.10.2014 18:48:35 Page: 1

Estimate Name: Canicula Campusus
Model Name: COCOMO II 2000
Process Model: COCOMO II Model
Estimate ID:
Model ID: 2000
Phases: Waterfall

Component Name: Component1
Increment: 1
Developed Size: 2,000
Component ID:
Level: 1
EAF: 1.0000

Phase	Effort (Person-Months)	Cost (K$)	Duration (Months)	Staffing
RQ -- Requirements	0.4	0.0	1.1	0.4
PD -- Product Design	1.1	0.0	1.6	0.7
DD -- Detailed Design	1.7	0.0	1.6	1.1
CT -- Code & Unit Test	2.3	0.0	2.1	1.1
IT -- Integration & Test	1.2	0.0	1.3	0.9
Development (PD+DD+CT+IT)	6.3	0.0	6.6	
Totals (RQ+PD+DD+CT+IT)	6.7	0.0	7.6	
MN -- Maintenance (per year)	0.0	0.0		0.0

Detailed Report



Canicula Campusus - Activity Report

Print | Export... | Save as Graphic | Headers | << Back | Next >>

Canicula Campusus - Activity Report

Costar 7.0 Demo 12.10.2014 18:48:42 Page: 1

Estimate Name:	Canicula Campusus	Estimate ID:	
Model Name:	COCOMO II 2000	Model ID:	2000
Process Model:	COCOMO II Model	Phases:	Waterfall

Effort in Person-Months

Activity	RQ	PD	DD	CT	IT	Total RQ to IT	MN
Requirements	0.2	0.1	0.1	0.1	0.0	0.5	0.0
Product Design	0.1	0.4	0.1	0.2	0.1	0.9	0.0
Programming	0.0	0.1	1.0	1.3	0.4	2.8	0.0
Test Plans	0.0	0.0	0.1	0.1	0.0	0.3	0.0
V & V	0.0	0.1	0.1	0.2	0.4	0.8	0.0
Project Office	0.1	0.1	0.1	0.2	0.1	0.6	0.0
CM/QA	0.0	0.0	0.1	0.2	0.1	0.4	0.0
Manuals	0.0	0.1	0.1	0.1	0.1	0.5	0.0
Totals	0.4	1.1	1.7	2.3	1.2	6.7	0.0

Activity Report

3.1.b Estimation by Analogy Technique

- Planning: 45 days
  - Game Design: 35 days
    - Characters: 10 days
    - Gameplay:8 days
    - Story:8 days
    - Game Goals:9 days
- Learning: 8 days
  - Platform: 6 days
    - Android: 3 days
    - Unity3D: 3 days
  - Scripting: 2 days
    - C#: 2 days
- Resources: 18 days
  - Audio: 6 days
    - Sounds: 3 days
      - Ambient Sound:1 day
      - Character Sound:1 day
      - Interface Sound:1 day
    - Musics:3 days
      - Menu Music:1 day
      - Background Music:2 days

- Graphic: 12 days
    - Interface Graphic: 4 days
        - Menu:2 days
        - Intro, Outro: 2 days
    - InGame Graphic:8 days
        - Texture: 4 days
        - Sprite Sheet: 4 days
- Design: 21 days
    - Level Design: 21 days
        - Gameplay Design: 12 days
        - Map Design: 9 days
            - Terrain: 9 days
- Programming: 44 days
    - Player: 14 days
        - Movement: 6 days
        - Interaction:8 days
            - Animations:3 days
            - Collisions:3 days
            - Sounds:2 days
    - AI: 12 days
        - Obstacle: 6 days
        - Enemy Movement:6 days
    - Gameplay Constraint: 8 days
    - Level Unlock System: 10 days
- Reporting
    - Report of Phases
        - Research Report
        - Resource Report
        - Test Report
        - Game Creation Report
            - Design Report
            - Programming Report
- Testing:16 days
    - Testing Phases:16 days
        - Platform Test:4 days
        - Performance Test:5 days
        - Fuctionality Test:7 days

4. PROJECT RESOURCES

4.1 Group Members

4.2 Mapping Between Group Member

Definition of the group members' roles:

- **Developer:** Developer is responsible of implementation of necessary algorithms such as character movements, artificial intelligence and gameplay.
- **Game Scenarist:** Game Scenarist is responsible of creating and designing all story.
- **Analyzer:** Analyzer should be able to sense and locate the missing pieces in scenerio gameplay and development phases.
- **Gameplay Designer:** Gameplay Designer handles all the gameplay logic.
- **Graphical Designer:** Graphical Designer is responsible of creating and finding of suitable graphics.
- **Tester:** Tester helps project in achieving promissed gameplay and implementations.

4.3 Project Component

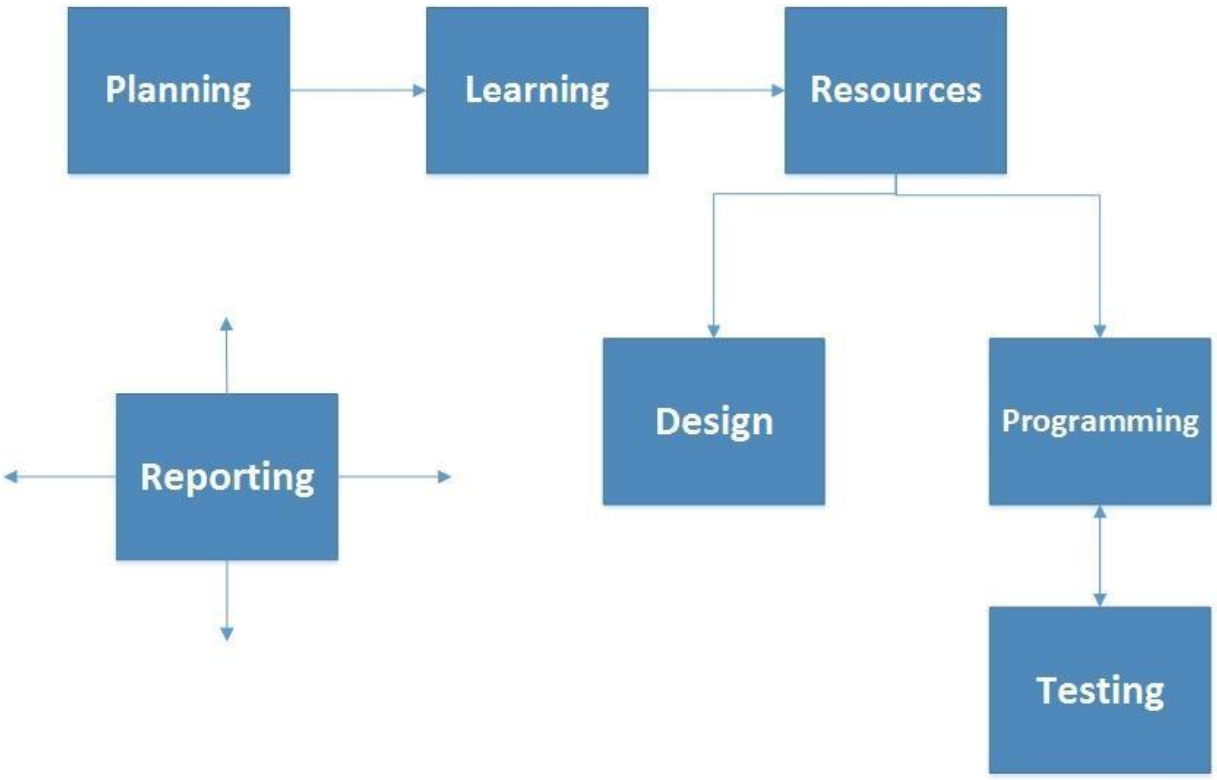
Canicula Campusus will be coded in C# language. Our system requirements are shown below:

- **Operating Sytems**: Windows XP SP2+, 7 SP1+, 8
- **GPU**: Graphics card with DX9 (shader model 2.0) capabilities
- Game Graphics
- Java Development Kit (JDK)
- Android SDK
- Android Device
- Unity3D
- MonoDevelop
- Google Play Account

5. SCHEDULE

5.1 Task Network

The Network diagram also known as Pert Chart for this project is shown as the foloowing. MS Visio is used for this Pert Chart.

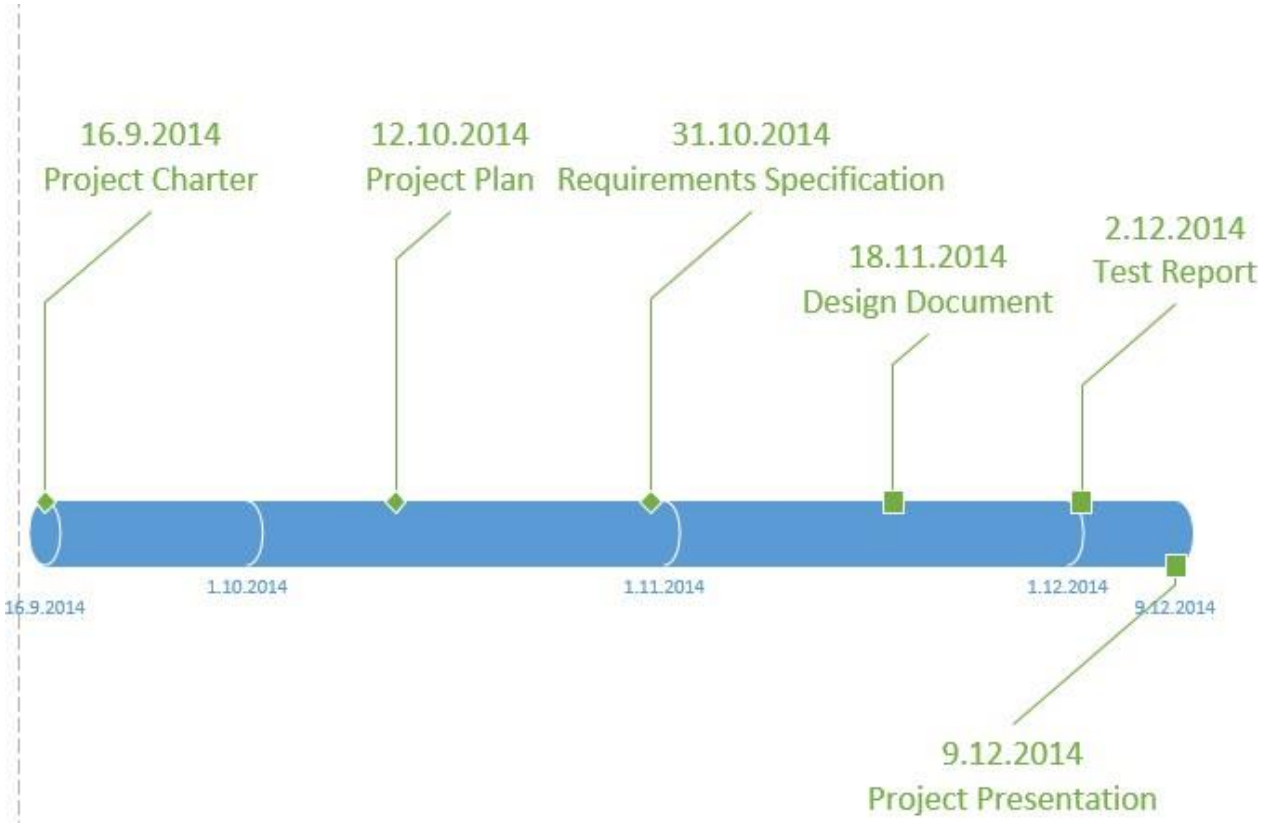


Pert Chart

- Planning
- Learning
- Resources
- Design
- Programming
- Testing
- Reporting

5.2 Timeline Chart

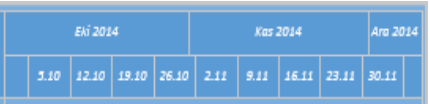
MS Visio is used for this timeline chart. This chart is provided in order to show the stage of project. It is shown as the following:



5.3 Gantt Chart

Gantt chart is used for so as to illustrate the start and finish dates of terminal elements and summary elements of our projects. MS Visio is again used to provide this Gantt Chart. It is shown as the following:

Kimlik	Görev Adı	Başlangıç	Bitiş	Süre	Kaynaklar
1	Planning	1.10.2014	13.10.2014	9g	Group
2	Learning	14.10.2014	17.10.2014	4g	Developers
3	Resources	14.10.2014	21.10.2014	6g	Designers, Analyzer
4	Designing	22.10.2014	30.10.2014	7g	Designers, Analyzer
5	Programming	22.10.2014	20.11.2014	22g	Developers
6	Reporting	1.10.2014	2.12.2014	45g	Group
7	Testing	21.11.2014	2.12.2014	8g	Testers
8	Final Presentation	9.12.2014	9.12.2014	1g	Group



Gantt Chart

6. RISK MANAGEMENT

6.1 Project Risks

Project risks are inexperience, time, lack of training, inadequate technology and reusability.

6.2 Risk Table

Risk	Cat	Prob	Impact	Remedy
Technology will not meet expectations	TE	30	1	A
Might not finish on time	BU	20	1	B
Less re-use than planned	PS	70	2	C
Lack of training on tools	TE	50	2	D
Staff inexperienced	ST	30	3	D

6.3 RMM Plan

A) Technology will not meet expectations.

Mitigation: This risk has high impact and to avoid the risk requirements must be keep in average level which depending on common usage of android versions.

Monitoring: To determine the occurrence of the risk the simulation device must be low quality so possible lack of performance can be recognized in coding state.

Management: To avoid that risk project will be tested in an average version of android and a low quality mobile phone.

B) Might not finish on time.

Mitigation: To avoid this risk each group members must do own part according to project plan.

Monitoring: Checking the all group members' work and setting weekly meetings and comparing works and Schedule helps to determine this risk.

Management: The best way to manage this risk create a working system and aware each team member.

C) Less reuse than planned.

Mitigation: This risk is also has important impact. To avoid that, design of program must be done carefully.

Monitoring: Counting of using and planning objects can be compared to avoid that risk.

Management: Review and revise of project design in some level of coding helps to manage this risk.

D) Lack of training on tools.

Mitigation: To avoid this risk, documentation of tools must be analyzed by group members.

Monitoring: The best way to monitor this risk is inspect the group member and checking the time of works.

Management: Arranging training and extra studies help to manage this risk.

E) Staff inexperienced.

Mitigation: This risk has not high impact on project because team member can gain experience during the project, yet this process causes losing time. To avoid this risk, team members has to have some background information.

Monitoring: Interaction with team members and checking their work helps to monitor this risk.

Management: The best way to manage this risk is choosing a team member which qualified and assign him/her to educate and lead other team members.

6.4 Overview of Risk Mitigation, Monitoring, Management

To avoid risks which listed above, project must be well designed and team members must keep in connection and discuss the project in regularly meetings. Documentations must be read and inexperienced members must work harder.