### SYSC 5104. Methodologies for Discrete Event Modelling and Simulation

### Project Report

### Simulating Waterfall Software Development Model

### Team Members:

### Saima Hidayatullah Khan (Student# 8511041)

### Zahra Parishani (Student# 101060131)

**Part I - Overview:**

The model represents the software development life cycle for a waterfall model approach. We have five sub-models, which consists of two coupled models and three atomic models. The coupled models further consist of two atomic models. The function of each atomic and coupled model can be defined as below:

Estimator:

It estimates the time for each phase of software development. We assume that it always allots *10%* of the project time for Requirements, *20%* for the Design and the remaining *70%* time for Development. Also, it sends the *time allotted* to each phase as its outputs.

Requirements (Coupled Model):

The model is responsible to evaluate the *total time* for requirements gathering phase to the Time/Cost Calculator model and a *done* message for Design model.

It has two atomic models, each representing a Business Analyst Team and a Client as under:

* Business Analyst Team:

This model defines the total time for requirements analysis. It adds more *50%* of its estimated time if there is *change* request from the Client. The output of this model is the *total time* taken by the Business Analyst Team with or without change and the *verification* request to the Client for the requirements document.

* Client:

The model sends a change request indicator to the Business Analyst Team. If the requirements document has some change, the indicator is one else zero.

Design:

This processes the design time and outputs the total design time to the Time / Cost Calculator.

Implementation (Coupled Model):

The model consists of two atomic models as below:

* Development Team:

The development team adds the additional bug fixing time taken by the team to the *estimated design time* once there are no bugs. It outputs a testing request to the Testing Team and the total time consumed to the Time / Cost Calculator with or without bug fix. We assume that it adds *50%* of its estimated time if there is a *bug fix* request from the Testing Team.

* Testing Team:

This model verifies the product verification request sent by the Development Team. It generates a message as *bug* with value one or zero to the Development Team, depending if there is some bug fix required or not.

Time / Cost Calculator:

This model generates the *actual time* *taken* and *cost* of the project as outputs. It assumes that the hourly rate for the company is *$100*.

Bug Out

Time Out

Time Out

Time In

Development Team

Bug In=?

ChangeOut

Change In=?

Time In

Client

**ProjectIn**

**ProjectSize**

**Project Cost**

**Total Time**

Time / Cost Calculator

Total Design Time

Total Implementation Time

Total RequirementsTime

Requirements

Implementation Time

Design Time

Req. Time

Estimator

Design

Testing Team

Implementation

Business Analyst Team

**Part II – Formal Specifications:**

* **Atomic Models:**

1. **Estimator:**

Reqtime, Designtime, Imptime

Projectin

reqtime

S = {passive, active}

X = {Projectin, Projectsize}

Y = {totaltime, projectcost}

designtime

Projectsize

δint (active) = passive

Projectsize , Projectin

imptime

δext (Projectin, Projectsize, passive) = active

Projectin,

Projectsize

δext (Projectin, Projectsize, active) = active

λ(active)

{

send output to all the three phases their estimated time as 10%, 20% and 70% respectively.

}

ta(passive) = infinity

ta(active) = preparation time

1. **Businessanalystteam:**

bustimeout, changein

bustimein, changeout

bustimein

bustime out

S = {passive, active}

X = {Projectin, Projectsize}

Y = {changein, bustimeout}

δint (active) = passive

bustimein, changeout

changeout

δext (bustimein, changeout, passive) = active

δext (bustimein, changeout, active) = active

λ(active)

{

send output the “total time” to Time / Cost Calculator

send output the “changein” to the Client

}

ta(passive) = infinity

ta(active) = preparation time

1. **Client:**

S = {passive, active}

X = {changein}

Y = {changeout}

δint (active) = passive

changein

changein

changeout

changein

changeout

δext (changein, passive) = active

δext (changein, active) = active

λ(active)

{

send output the “changeout” to the Businessanalystteam

}

ta(passive) = infinity

ta(active) = preparation time

1. **Design:**

S = {passive, active}

X = {designtimein}

Y = {designtimeout}

δint (active) = passive

designtimein

designtimein

designtimeout

designtimein

designtimeout

δext (designtimein, passive) = active

δext (designtimein, active) = active

λ(active)

{

send output the “designtimeout” to the Time / Cost Calculator

}

ta(passive) = infinity

ta(active) = preparation time

1. **Developmentteam:**

S = {passive, active}

X = {devtimein,bugout}

Y = {devtimeout,bugin}

δint (active) = passive

Devtimein, bugout

devtimein

devtimeout

Devtimein, bugout

Devtimeout,bugin

δext (devtimein, bugout, passive) = active

bugin

bugout

δext (devtimein, bugout, active) = active

λ(active)

{

send output the “devtimeout” to the Time / Cost Calculator

}

ta(passive) = infinity

ta(active) = preparation time

1. **Testingteam:**

S = {passive, active}

X = {bugin}

Y = {bugout}

δint (active) = passive

bugin

bugin

bugin

bugout

δext (bugin, passive) = active

bugout

δext (bugin, active) = active

λ(active)

{

send output the “bugout” to the Developmentteam

}

ta(passive) = infinity

ta(active) = normal distribution

1. **Timecostcalculator:**

S = {passive, active}

X = {totreqtime, totdesigntime, totimptime}

Y = {totaltime, projectcost}

totdesigntime

Totreqtime,totdesigntime,totimptime

Totaltime, projectcost

δint (active) = passive

totreqtime

δext (totreqtime, totdesigntime, totimptime, passive) = active

Projectcost

Totaltime

totimptime

δext (totreqtime, totdesigntime, totimptime, active) = active

Totreqtime,totdesigntime,totimptime

λ(active)

{

send output the “totaltime”, “projectcost” to the whole model and simulation

}

ta(passive) = infinity

ta(active) = preparation time

* **Coupled Models:**

1. **Requirements:**

X = {reqtime};

Y = {reqtimeout};

D = {businessanalysttime, client};

EIC {Requirements.reqtimein, businessanalystteam.bustimein, businessanalystteam.changeout, businessanalystteam.changein, client.changein, client.changeout}

EOC {Requirements.reqtimeout, businessanalystteam.bustimeout}

SELECT: ({businessanalystteam,client}) = businessanalystteam;

1. **Implementation:**

X = {imptime};

Y = {imptimeout};

D = {developmentteam, testingteam};

EIC {Implementation.imptimein, developmentteam.devtimein, developmentteam.bugout, developmentteam.bugin, testingteam.bugin, testingteam.bugout}

EOC {Implementation.imptimeout, developmentteam.devtimeout}

SELECT: ({developmentteam, testingteam}) = developmentteam;

1. **Waterfall Software Development Model:**

X = {Projectin,Projectsize};

Y = {Totaltime, Projectcost };

D = {Requirements, Design, Implementation, Timecostcalculator};

EIC { WSDM.Projectin, Estimator.Projectin, WSDM.Projectsize, Estimator.Projectsize, Requirements.reqtimein, Implementation.imptimein, Design.designtimein, Requirements.reqtimeout, Implementation.imptimeout, Design.designtimeout}

EOC {Timecostcalculator.Totaltime, Timecostcalculator.Projectcost}

SELECT: ({Requirements, Design, Implementation, Timecostcalculator }) = Timecostcalculator;

**Part III – Testing Cases:**

* **Atomic Models:**

1. **Estimator:**

Input:

00:00:05:00 Projectin 1 00:00:05:00 Projectsize 600

00:00:20:00 Projectin 1 00:00:20:00 Projectsize 500

00:00:35:00 Projectin 1 00:00:35:00 Projectsize 900

Output:

00:00:10:000 reqtime 60

00:00:10:000 designtime 120

00:00:10:000 imptime 420

00:00:25:000 reqtime 50

00:00:25:000 designtime 100

00:00:25:000 imptime 350

00:00:40:000 reqtime 90

00:00:40:000 designtime 180

00:00:40:000 imptime 630

1. **Businessanalystteam:**

Input:

00:00:06:00 bustimein 90

00:00:12:00 changeout 0

00:00:18:00 bustimein 120

00:00:24:00 changeout 1

00:00:30:00 bustimein 130

00:00:36:00 changeout 0

Output:

00:00:11:000 bustimeout 90

00:00:23:000 bustimeout 120

00:00:35:000 bustimeout 195

00:00:35:000 changein 1

1. **Client:**

Input:

00:00:01:00 changein 1

00:00:07:00 changein 0

Output:

00:00:06:000 changeout 1

00:00:12:000 changeout 0

1. **Design:**

Input:

00:00:06:00 designtimein 100

00:00:12:00 designtimein 50

00:00:18:00 designtimein 40

Output:

00:00:11:000 designtimeout 100

00:00:17:000 designtimeout 50

00:00:23:000 designtimeout 40

1. **Developmentteam:**

Input:

00:00:06:00 devtimein 50 00:00:06:00 bugout 0

00:00:14:00 devtimein 120 00:00:14:00 bugout 1

00:00:22:00 devtimein 130 00:00:22:00 bugout 0

Output:

00:00:08:987 bugin 1

00:00:08:987 devtimeout 75

00:00:15:796 bugin 1

00:00:15:796 devtimeout 180

00:00:23:957 bugin 1

00:00:23:957 devtimeout 195

1. **Testingteam:**

Input:

00:00:01:00 bugin 1

00:00:07:00 bugin 0

Output:

00:00:06:000 bugout 1

00:00:12:000 bugout 0

1. **Timecostcalculator:**

Input:

00:00:06:00 totalreqtime 180

00:00:12:00 totaldesigntime 60

00:00:18:00 totalimptime 100

00:00:24:00 totalreqtime 120

00:00:30:00 totaldesigntime 160

00:00:36:00 totalimptime 90

00:00:42:00 totalreqtime 80

00:00:48:00 totaldesigntime 120

00:00:54:00 totalimptime 200

Output:

00:00:23:000 totaltime 340

00:00:23:000 projectcost 34000

00:00:41:000 totaltime 370

00:00:41:000 projectcost 37000

00:00:59:000 totaltime 400

00:00:59:000 projectcost 40000

**Waterfall Software Development Model:**

Input:

00:01:00:00 Projectin 1

00:01:10:00 Projectsize 600

00:30:00:00 Projectin 1

00:30:10:00 Projectsize 1000

01:00:00:00 Projectin 1

01:00:10:00 Projectsize 800

Output:

00:01:25:000 totaltime 120

00:01:25:000 projectcost 12000

00:30:25:000 totaltime 200

00:30:25:000 projectcost 20000

01:00:25:000 totaltime 160

01:00:25:000 projectcost 16000