TDT4240

Software Architecture

Spring 2012

Group A17
Android SDK

Steinar Haram

Emil Andreas Mork

Stian Sørebø

Ole Jørgen Rishoff

ATAM

Primary focus:
Maintainability

Secondary focus:
Usability

Contents

List of Tables

List of Figures

1 Introduction

1.1 Evaluated group - A16

Group members:

Alexander Benjamin Asplund	alexaasp@stud.ntnu.no
Stephan Nordnes Eriksen	stephann@stud.ntnu.no
Erik Stokken Samuelson	samuelss@stud.ntnu.no
Simen Fivelstad Smaaberg	smaaberg@stud.ntnu.no
Lichao Tang	lichao@stud.ntnu.no

1.2 Focus attributes

Primary focus:	Testability
Secondary focus:	Modifiability
Tertiary focus:	Usability

2 Utility analysis

2.1 Attribute utility table

Under modifiability there were originally three attributes. The two attributes "Add Weapon" and "Add ingame objects" are both split into add simple and add advanced, weapons and ingame objects respectively. This is done following our meeting with A16. The attribute "Change GUI" is specific to simple changes to menu buttons and layout.

Attribute	**Scenario**	**Priority**	**Details**
Testability	Unit testing	(H,M)	90% bug discovery
Testability	Code coverage	(H,M)	80% of code covered by tests
Testability	Test game functionality	(H,L)	Tests should run in 30 seconds
Modifiability	Add simple weapon	(M,L)	Implemented and tested in 1 hour
Modifiability	Add advanced weapon	(L,H)	Implemented and tested in 5 hours
Modifiability	Add simple object	(M,L)	Implemented and tested in 1 hour
Modifiability	Add advanced object	(L,H)	Implemented and tested in 5 hours
Modifiability	Change GUI	(M,L)	Implement changes in 2 hours

Table 1: Attribute utility table

2.2 Attribute utility tree

The utility tree corresponds to the table structure from the previous section.

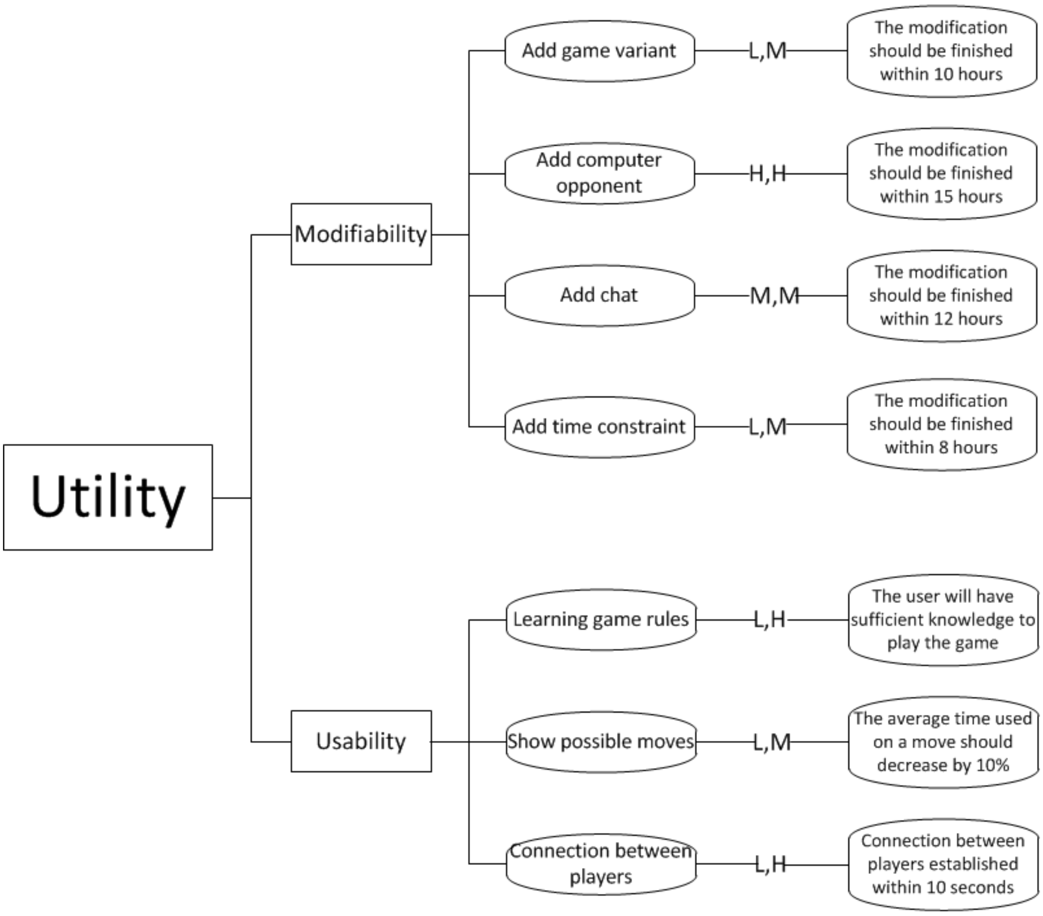
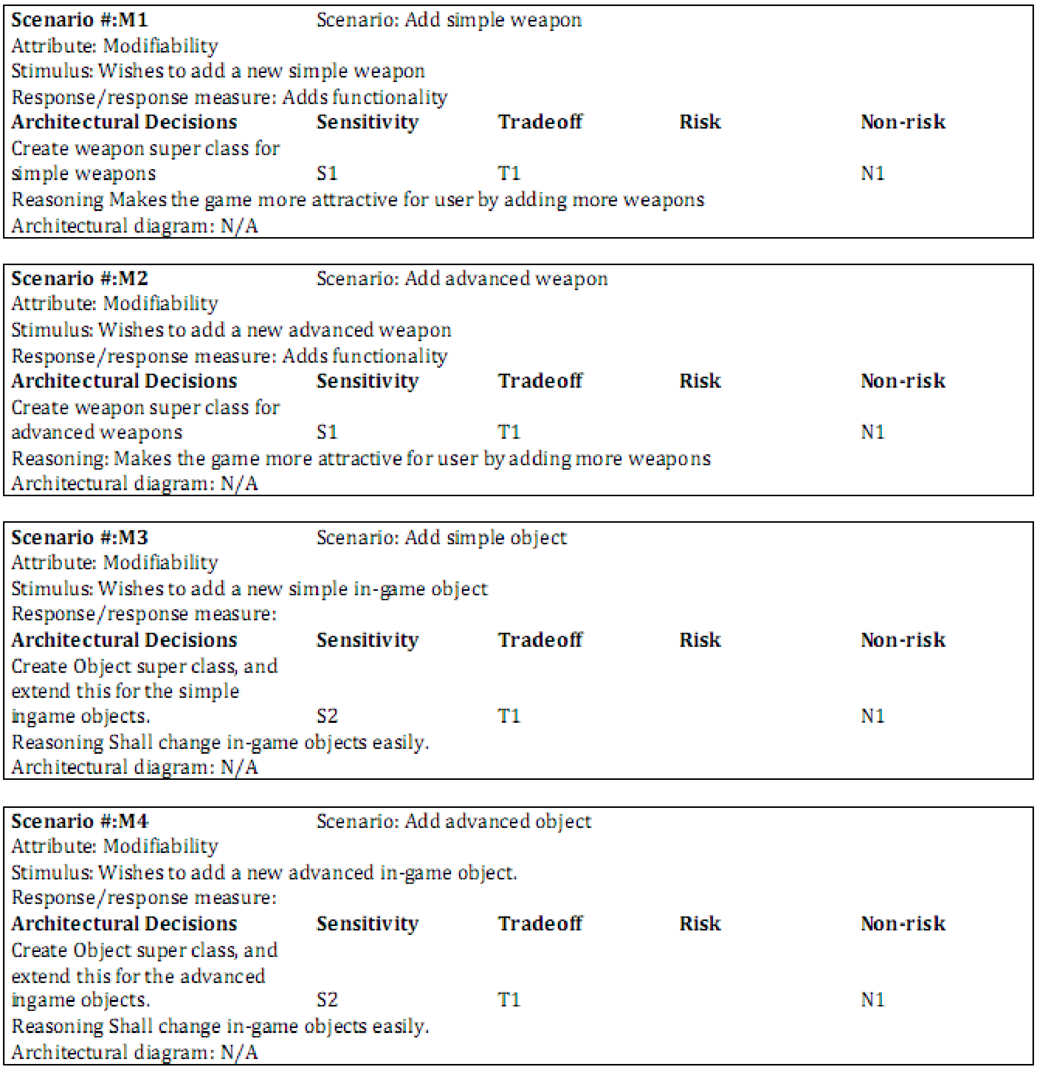

Utility
Modifiability
Add game variant
L,M
The modification should be finished within 10 hours
Add computer opponent
H,H
The modification should be finished within 15 hours
Add chat
M,M
The modification should be finished within 12 hours
Add time constraint
L,M
The modification should be finished within 8 hours
Usability
Learning game rules
L,H
The user will have sufficient knowledge to play the game
Show possible moves
L,M
The average time used on a move should decrease by 10%
Connection between players
L,H
Connection between players established within 10 seconds

Figure 1: Attribute utility tree

3 Analysis of architectural approach



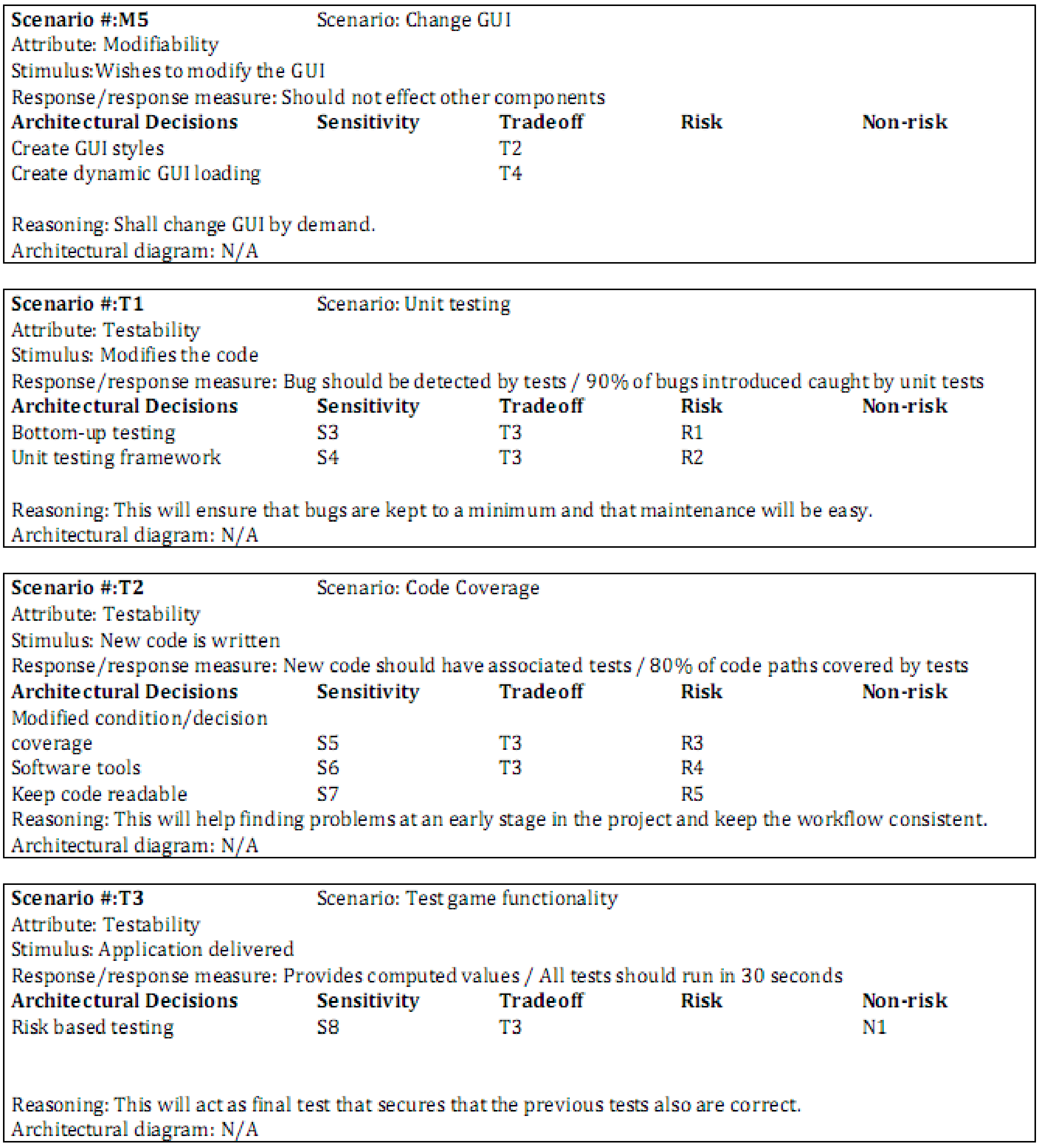
Scenario #:M1	Scenario: Add simple weapon			
Attribute: Modifiability				
Stimulus: Wishes to add a new simple weapon				
Response/response measure: Adds functionality				
Architectural Decisions	**Sensitivity**	**Tradeoff**	**Risk**	**Non-risk**
Create weapon super class for simple weapons	S1	T1		N1
Reasoning Makes the game more attractive for user by adding more weapons				
Architectural diagram: N/A				

Scenario #:M2	Scenario: Add advanced weapon			
Attribute: Modifiability				
Stimulus: Wishes to add a new advanced weapon				
Response/response measure: Adds functionality				
Architectural Decisions	**Sensitivity**	**Tradeoff**	**Risk**	**Non-risk**
Create weapon super class for advanced weapons	S1	T1		N1
Reasoning: Makes the game more attractive for user by adding more weapons				
Architectural diagram: N/A				

Scenario #:M3	Scenario: Add simple object			
Attribute: Modifiability				
Stimulus: Wishes to add a new simple in-game object				
Response/response measure:				
Architectural Decisions	**Sensitivity**	**Tradeoff**	**Risk**	**Non-risk**
Create Object super class, and extend this for the simple ingame objects.	S2	T1		N1
Reasoning Shall change in-game objects easily.				
Architectural diagram: N/A				

Scenario #:M4	Scenario: Add advanced object			
Attribute: Modifiability				
Stimulus: Wishes to add a new advanced in-game object.				
Response/response measure:				
Architectural Decisions	**Sensitivity**	**Tradeoff**	**Risk**	**Non-risk**
Create Object super class, and extend this for the advanced ingame objects.	S2	T1		N1
Reasoning Shall change in-game objects easily.				
Architectural diagram: N/A				

Figure 2: Architectural approach, part one



Scenario #:M5 Scenario: Change GUI

Attribute: Modifiability

Stimulus:Wishes to modify the GUI

Response/response measure: Should not effect other components

Architectural Decisions	Sensitivity	Tradeoff	Risk	Non-risk
Create GUI styles		T2		
Create dynamic GUI loading		T4		

Reasoning: Shall change GUI by demand.

Architectural diagram: N/A

Scenario #:T1 Scenario: Unit testing

Attribute: Testability

Stimulus: Modifies the code

Response/response measure: Bug should be detected by tests / 90% of bugs introduced caught by unit tests

Architectural Decisions	Sensitivity	Tradeoff	Risk	Non-risk
Bottom-up testing	S3	T3	R1	
Unit testing framework	S4	T3	R2	

Reasoning: This will ensure that bugs are kept to a minimum and that maintenance will be easy.

Architectural diagram: N/A

Scenario #:T2 Scenario: Code Coverage

Attribute: Testability

Stimulus: New code is written

Response/response measure: New code should have associated tests / 80% of code paths covered by tests

Architectural Decisions	Sensitivity	Tradeoff	Risk	Non-risk
Modified condition/decision coverage	S5	T3	R3	
Software tools	S6	T3	R4	
Keep code readable	S7		R5	

Reasoning: This will help finding problems at an early stage in the project and keep the workflow consistent.

Architectural diagram: N/A

Scenario #:T3 Scenario: Test game functionality

Attribute: Testability

Stimulus: Application delivered

Response/response measure: Provides computed values / All tests should run in 30 seconds

Architectural Decisions	Sensitivity	Tradeoff	Risk	Non-risk
Risk based testing	S8	T3		N1

Reasoning: This will act as final test that secures that the previous tests also are correct.

Architectural diagram: N/A

Figure 3: Architectural approach, part two

4 Sensitivity points

S1	A weapon super class affects modifiability positively
S2	An object super class affects modifiability positively
S3	Integration testing becomes much easier
S4	Help simplify the process of unit testing
S5	Every condition in a decision in the program has taken all possible outcomes at least once
S6	Improves the code reliability
S7	Documentation affects modifiability positively
S8	Exercises the most important tests first

Table 2: Sensitivity points

5 Trade-off points

T1	Affects modifiability positively and testability negatively
T2	Affects modifiability positively and testability negatively
T3	Affects testability positively and the performance negatively
T4	Affects modiafibility positively and performance negatively

Table 3: Trade-off points

6 Risks and non-risks

R1	Runs the risk that modules may be coded without having a clear idea of how they link to other parts of the system
R2	Advanced unit test features may be missing or must be hand-coded
R3	The test cover more than it needs, and time can be wasted
R4	The software tools may not cover all the errors that you may find by the manual approach
R5	Time consuming considering the time limit and size on the project
N1	There are no drawbacks associated with this activity

Table 4: Risks and non-risks

7 Own experiences using ATAM

The experience of discussing our own and our opponent group's architecture was good, and forced us to think critically about our architecture, while coming up with suggestions and risks associated with our opponent group's tactics.

We mainly used the short ATAM summary provided, and followed the steps in the process in our meeting. We feel that the steps are fine, but the detail level in the summary could be higher.

In addition to an increased level of insight in architectural problematics, we also gained an interest as stakeholders to our opponent group's project.

8 Problems and issues

Our opponent group was unable to meet with us on Monday in the first evaluation phase, meaning that we were unable to take a deeper look into their documentation before Thursday's meeting.

The rules for the conduct of the exercise was poorly specified, and the template differs a lot from the steps in the textbook. We feel that the assistants in the course should take responsibility for the distribution of the documentation. In addition, we expected that there would be some kind of facilitation of the first meeting of the groups. This was not the case, and we wasted our time showing up for this.

The ATAM-template is poorly detailed, and has caused a lot of confusion during this project phase.

9 Change log

12.03.2012	First version of the document
27.04.2012	LaTeXversion of the document

Table 5: Document changes