

# Unit Testing Plan

## for Coffee machine System

- Test Plan
- Test Design Specification
- Test Cases Specification

Project Team

**2 Team**

Date

**2016-11-06**

---

### Team Information

201511284 문 성찬

201511284 이 종빈

201211356 송 원중

## Table of Contents

1	Introduction .....	4
1.1	Objectives.....	4
1.2	Background .....	4
1.3	Scope.....	4
1.4	Project plan .....	4
1.5	Configuration management plan.....	4
1.6	References.....	4
2	Test items .....	4
3	Features to be tested.....	5
4	Features not to be tested .....	5
5	Approach.....	6
6	Item pass/fail criteria .....	6
7	Unit test design specification.....	6
7.1	Test design specification identifier .....	6
7.2	Features to be tested .....	7
7.3	Approach refinements.....	7
7.4	Test identification .....	7
7.5	Feature pass/fail criteria .....	8
8	Unit test case specification.....	8
8.1	Test case specification identifier .....	8
8.2	Test items .....	9

8.3	Input specifications.....	9
8.4	Output specifications.....	9
9	Testing tasks .....	9
10	Environmental needs .....	9
11	Unit Test deliverables.....	10
12	Schedules .....	10

## 1 Introduction

### 1.1 Objectives

본 문서는 <Coffee Machine>을 구현하기 위한 요구사항들을 명세한 문서이다. Coffee Machine은 사용자의 선택에 따라 원하는 커피를 추출해 준다

### 1.2 Background

Coffee Machine은 사용자의 기호에 맞는 커피를 추출해주는 기계이다.

Unit Test는 시스템을 구성하는 최소 단위 모듈들을 대상으로 하는 test이며, 시스템에 관련된 데이터 및 프로세스들이 요구사항을 만족하고 제대로 작동하는지 확인할 수 있는 기본적인 테스트 방법이다.

### 1.3 Scope

Coffee Machine은 자신의 상태를 관리하며 사용자들에게 알려주고, 사용자로부터 명령 및 입력 내용을 받아 출력을 만들어내는 역할을 한다. Coffee Machine은, Machine의 상태 즉 원두와 물의 부족 유무와 청소 상태를 바탕을 standby인지 working 인지과 결합하여 사용자의 요구를 실행한다. 모든 시스템은 SW 만으로 구현한다. HW가 필요한 부분은 SW모듈을 만들어 가상의 HW 를 구현한다.

### 1.4 Project plan

### 1.5 Configuration management plan

Coffee machine의 program source code및 unit test를 위한 code는 식그윈(Cygwin) 환경에서 이루어지며 각 함수 별로 나누어 입력 값과 출력 값을 예상하고 맞는지 판단한다. 이때 code들은 지속적으로 변경 및 개선이 이루어진다.

### 1.6 References

T2\_2016\_SRA 2.0

T2\_2016\_SDS 1.0

## 2 Test items

Coffee Machine을 구성하는 최소 단위의 모듈들은 T2\_2016\_SRA\_2.0를 기반으로 작성되었으며, 각 모듈 별로 입력 값(input)을 집어 넣었을 경우 적절한/예상한 출력 값(output)이 나오

는지를 테스트 한다.

### 3 Features to be tested

ID	Name	Description
2.1.8	Queueing	예약 시 받는 요구 시간을 int형으로 받은 것을 현 시간 (real time)이랑 비교하게 바꿔주는 기능. Input: 시+분 int*2/output: time.
2.1.8	Queueing	예약시간들을 sorting시킨다. Input: 정렬되지 않은 시간+UC(명령)/ output: 시간 순서대로 정렬된 UC.
2.1.1	Command distributor	명령(UC)를 Remain(재료)바탕으로 실행이 가능한지 판단. 및 실행가능하면 Extract/Clean trigger. 안될 시 Reject. Input: UC & remain structure & machine status Output: Process trigger or reject.
2.1.7	Refill	Refill 데이터가 들어왔을 때 재료의 잔량이 제한된 범위를 벗어나면 reject, 범위 안에 있을 시 각 재료의 충전량만큼 더 해준다. Input: 자료명+양 Output: trigger 재료/reject

### 4 Features not to be tested

단순 데이터 전달 프로세스, 단순 입력에 대한 출력 값 저장, 센서, 단순 데이터 취합은 unit test에서 제외된다.

ID	Name	Description
1.1	Clean interface	사용자의 선택으로 청소를 지시함. 사용자의 선택 전달.
1.2	Reserve interface	사용자의 선택으로 예약시간 및 명령을 지정. 예약 시간 및 명령 전달.
1.3	Power interface	Coffee machine이 on/off 되어있는지 판단하여 데이터 전달.

1.4	Extract interface	사용자의 선택으로 추출을 지시함. 사용자의 선택 전달.
1.5	Heat interface	사용자의 선택으로 추출할 커피의 온도를 결정. 온 오프인지 전달.
1.6	Concentration interface	사용자의 선택으로 커피의 농도 조절. 3가지 중 하나의 선택 전달.
1.7	Coffee interface	커피원두의 잔량 전달.
1.8	Water interface	물의 잔량 전달
1.9	Leftover interface	커피 찌꺼기 존재 유/무 전달
1.10	Coffee dust interface	커피 가루의 잔량 전달.
2.1.8	queueing	현재 시간과 명령이 예정되어있는 시간을 비교하여 명령 판단 프로세스 실행.
2.1.5.1	Clean data process	청소 process가 트리거 될 시 청소를 실행.
2.1.6.1	Extract data process	추출 process가 트리거 시 추출을 실행
2.1.7.1	Refill process	Refill process가 트리거 시 특정 재료의 보충이 이루어짐.

## 5 Approach

Coffee machine의 Program source code 및 Unit test 를 위한 test code는 Cygwin+vi+gcc환경에서 이루어지며, Program source code/test code의 변경 및 수정사항은 지속적으로 통합하여서 test하며 각 모듈의 Unit test를 마무리 짓는다.

## 6 Item pass/fail criteria

Unit test할 내용들이 특정 input에 대하여 정확한(예상된) output이 나올 경우 pass이며 예상된 값 이외의 오류가 발생할 시 fail이다.

## 7 Unit test design specification

### 7.1 Test design specification identifier

T2\_Coffeemachine\_(name of process)\_(number of unit test in process)

## 7.2 Features to be tested

-본 문서 3. Features to be tested 참조

## 7.3 Approach refinements

T2의 최소한 단위의 모듈을 각 요구사항별로 맞는지 확인하는 test code를 작성하고  
이가 제대로 작동하는지 unit test를 이용하여 확인한다..

## 7.4 Test identification

<Table3: test identification>

Identifier	Feature	Valid/Invalid value
T2_Coffeemachine_2.1.1_000	2.1.1 Command distributor	Input: UC(user choice) Remain Machine status Output: trigger extract/clean/refill reject beep+display
T2_Coffeemachine_2.1.1_001	2.1.1 Command distributor	Input UC(extract) Remain(bean/dust) Machine status(leftover/is_running/power/concentration/heat) Output Trigger extract Reject
T2_Coffeemachine_2.1.1_002	2.1.1 Command distributor	Input UC(Clean) Remain(water) Machine status(is_running/power) Output Trigger clean Reject
T2_Coffeemachine_2.1.8_000	2.1.8 queueing	Input: UC(user choice)

		Time(int*2) Output: UC+time(time.h)
T2_Coffeemachine_2.1.8_001	2.1.8 queueing	Input: UC+time(time.h) Output: Sorted time based UC
T2_Coffeemachine_2.1.8_002	2.1.8 queueing	Input: Sorted time based UC/real time Output: UC //예약 시간과 현재 시간이 일치할 시 저장되어있는 명령 실행.
T2_Coffeemachine_2.1.7_000	Refill	Input: UC(refill ingredient+amount) Output: Trigger(ingredient process)+amount Reject display

### 7.5 Feature pass/fail criteria

각 범위 별로 한계 범위(제한사항)들을 고려하여 테스트해보고, 입력할 수 있는 값의 최대 최소를 넣어보고 그 범위 밖의 값을 넣어서 결과 값이 올바르게 나오는지 확인한다.

## 8 Unit test case specification

### 8.1 Test case specification identifier

<Table4: test case identification>

Test case identifier	Input specification	Output Specification
T2_Coffeemachine_2.1.1_000	UC(user choice) Remain Machine status	trigger extract/clean/refill reject beep+display
T2_Coffeemachine_2.1.1_001	UC(extract) Remain(bean/dust) Machine	Trigger extract Reject



	status(leftover/is_running/power/concentration/heat)	
T2_Coffeemachine_2.1.1_002	UC(Clean) Remain(water) Machine status(is_running/power)	Trigger clean Reject
T2_Coffeemachine_2.1.8_000	UC(user choice)+time(int*2)	UC+time(time.h)
T2_Coffeemachine_2.1.8_001	UC+time(time.h)	Sorted time based UC
T2_Coffeemachine_2.1.8_002	Sorted time based UC, Clock based time(tick)	UC
T2_Coffeemachine_2.1.7_000	UC(refill ingredient+amount)	Trigger(ingredient process)+amount Reject display

## 8.2 Test items

<Table3: Test design identification> 참조

## 8.3 Input specifications

<Table4: Test case identification> input 참조

## 8.4 Output specifications

<Table4: Test case identification> output 참조

## 9 Testing tasks

1)SA/SD작성

2)Process를 모듈 단위로 나누기

3)모듈별 input/output 정하기

4)소스 코드 작성

5)Unit test 실시

6)Test report

## 10 Environmental needs

-Unit test환경설정: Cygwin gcc vi

11 Unit Test deliverables

- 1) Unit test plan
- 2) Unit test design specification
- 3) Unit test case specification
- 4) Unit test report

12 Schedules

9 Testing task 참조.