

Software Development Plan of XXX software		
Doc # XXX-SDP	Version: 1.3	Page 1 / 7

REVISION HISTORY

Date	Version	Description	Author
17.10.2021	1.1	All fields filled.	Başak Balcı
17.10.2021	1.2	UML Diagram added.	Başak Balcı
17.10.2021	1.3	Flow Charts added.	Başak Balcı

Software Development Plan of XXX software		
Doc # XXX-SDP	Version: 1.3	Page 2 / 7

TABLE OF CONTENTS

Revision History	1
1 Identification	3
1.1 <i>Document overview</i>	3
1.2 <i>Abbreviations</i>	3
1.2.1 Abbreviations	3
2 Software Development Activities	3
2.1 <i>Software development process</i>	3
2.1.1 Overview of process phases	3
2.1.2 Technical documentation	3
2.1.3 Deliverables	3
2.2 <i>Software development tools</i>	3
2.2.1 Workstation	3
2.2.2 Requirements management and documentation	4
2.2.3 Software Design	4
2.2.4 Coding and automated tests	4
2.2.5 Configuration management	4
2.3 <i>Software development rules and standards</i>	4
3 Responsibilities	6
3.1 <i>Activities and responsibilities</i>	6
4 Risk Assessment	6
4.1 <i>Risk Analysis</i>	6
4.2 <i>Risk Planning</i>	6

Software Development Plan of XXX software		
Doc # XXX-SDP	Version: 1.3	Page 3 / 7

1 Identification

1.1 Document overview

This document contains the software development plan of software DFASimulator.

This project simulates the behavior of a DFA. It takes an file as an input and returns an file as an output. In the input file there are information about the DFA: number of states, number of variables, number of goal states, states, goal state, variables, transition function and the strings to be computes. In the output file there are information about if the given strings are accepted or not and the route taken. The algorithm computes the behavior of the defined DFA.

1.2 Abbreviations

1.2.1 Abbreviations

DFA: Deterministic Finite Automata

2 Software Development Activities

The section lists and describes the software development activities of DFASimulator software development project.

2.1 Software development process

This is a course project, which adopts the waterfall model as the software development process.

2.1.1 Overview of process phases

The software development process for the project will be composed of the following phases:

- Design
- Implementation

These phases will follow each other sequentially, where each phase starts just after the completion of the previous one.

2.1.2 Technical documentation

Only this documentation will be produced

- Software Development Plan: SDP

2.1.3 Deliverables

The following items will be delivered at the end of the process:

- Technical documentation as outlined in Section 2.1.2
- Software and its configuration files

2.2 Software development tools

2.2.1 Workstation

- Developers Computer:
 - CPU: Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz 2.40 GHz
 - GPU: AMD RADEON R7 M360
 - RAM: 8,00 GB

2.2.2 Requirements management and documentation

- Microsoft Word 16

2.2.3 Software Design

- Draw IO 9.7.3

2.2.4 Coding and automated tests

- IntelliJIDEA 2020.3.1

2.2.5 Configuration management

GitHub¹, in private, will be used for software configuration management and tracking issues regarding the software development. A public repository will be created for this purpose.

2.3 Software development rules and standards

UML² will be used for software design documentation.

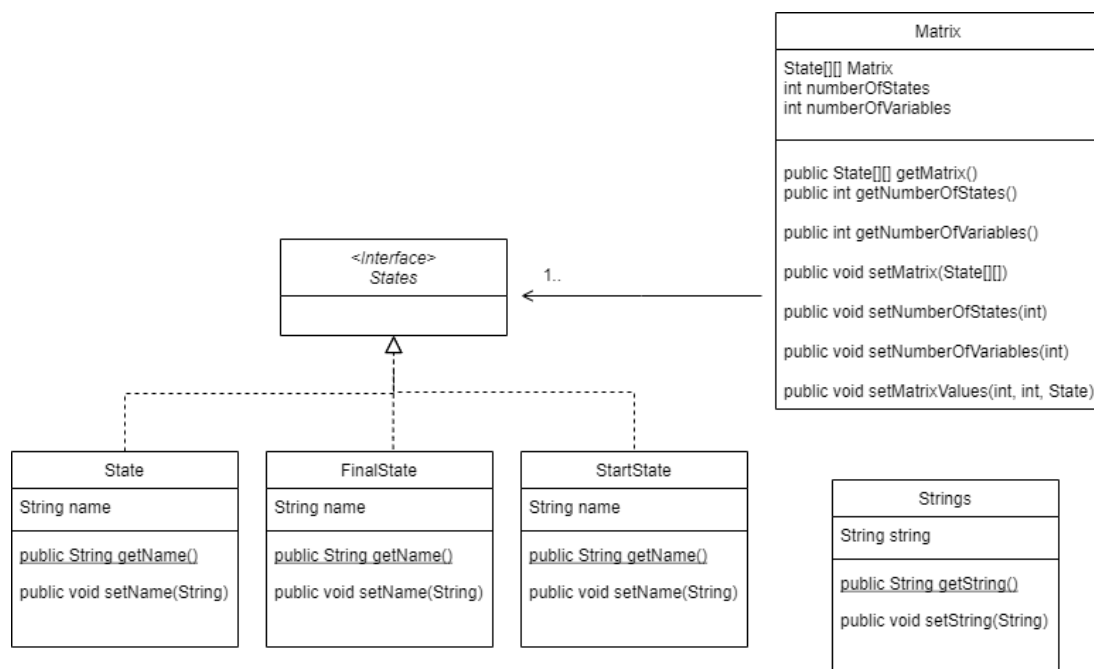


Figure 1 UML Diagram

¹ <http://www.github.com>

² <http://www.uml.org/>

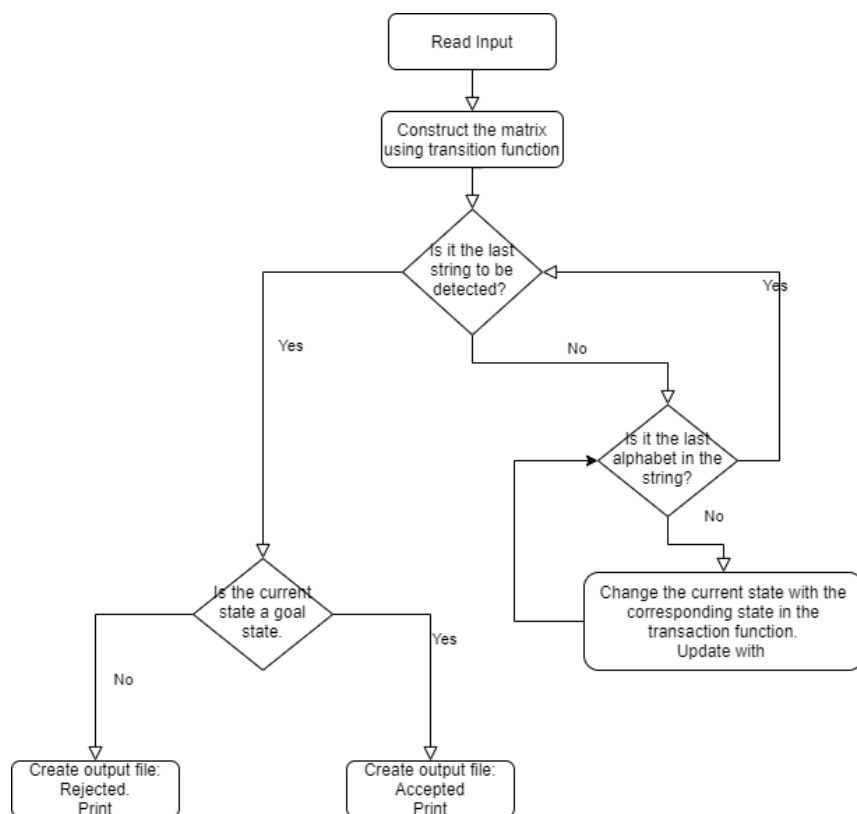


Figure 2 Algorithm Flow Diagram

Input file transition function example				Transition Function as Matrix		
	1	2	3		a	b
i ₁	q1	a	q1	q1	X	.
i ₂	q1	b	q2	q2	.	.
i ₃	q2	a	q2	.	.	.
i ₄	q2	b	q1	.	.	.
.
.
.

Column of X:	i % number of variables
Row of X:	i / number of variables
Value of X:	(Input file transition function example) [i] [2]

Figure 3 Calculation of the Transition Function Matrix

3 Responsibilities

3.1 Activities and responsibilities

Activity	Responsibility	Comment
Project management	Başak Balcı	
Configuration tools management	Başak Balcı	
Setting up the Development tools	Başak Balcı	

4 Risk Assessment

4.1 Risk Analysis

Risk Type	Possible Risk
Tools	Software tools cannot work together in an integrated way.
Requirements	Changes to requirements that require major design rework are proposed.
Estimation	The time required to develop the software is underestimated. The rate of defect repair is underestimated. The size of the software is underestimated.

4.2 Risk Planning

Risk	Probability	Effects
Software tools cannot be integrated	High	Tolerable
Changes to requirements that require major design rework are proposed.	Moderate	Tolerable
The time required to develop the software is underestimated	High	Serious
The rate of defect repair is underestimated	Moderate	Tolerable
The size of the software is underestimated	High	Tolerable

Risk	Strategy
Tools	In the beginning of the project all tool integration will be done and in possible failure support will be requested.
Requirements	All the requirements will be listed as will be

Software Development Plan of XXX software		
Doc # XXX-SDP	Version: 1.3	Page 7 / 7

	closely followed.
Estimation	Design will be planned very detailed to reduce this risk.