Name: Muhammad Bilal

Father Name: Haneef Qureshi

Section: A

Teacher: Dr. Shahab Ahmed Siddique

Course: CSSE - 313

Seat Number: EB-20103075

Program: BSSE

Department: UBIT

Assignment: Lab

Q1. Make a Project Plan of at least 50 activities for the following types of projects: [2.5 marks each]

a. Applying Booch Method

b. Applying RUP

c. Applying Waterfall

d. Applying Spiral

Choose your own system.

A. I have chosen **Bank Management System**. Bank Management System is based on dot NET and is a major project for students. It is used to Keep the records of clients, employee etc. in Bank. The bank management system is an application for maintaining a person's account in a bank . The system provides the access to the customer to create an account, deposit/withdraw the cash from his account, also to view reports of all accounts present. The following presentation provides the specification for the system.

Following are the **Projects Plan** for **Bank Management System.**

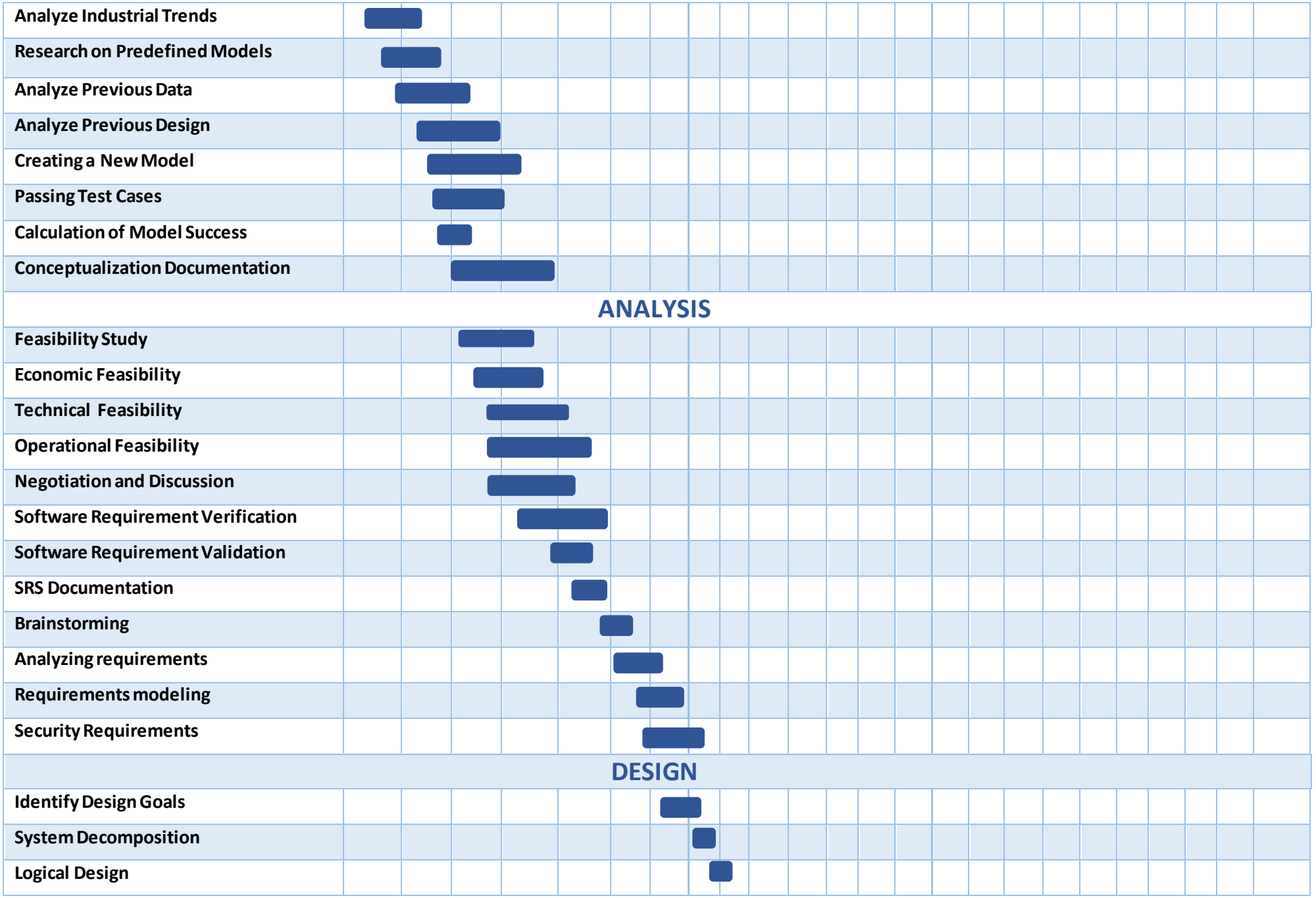
(a)

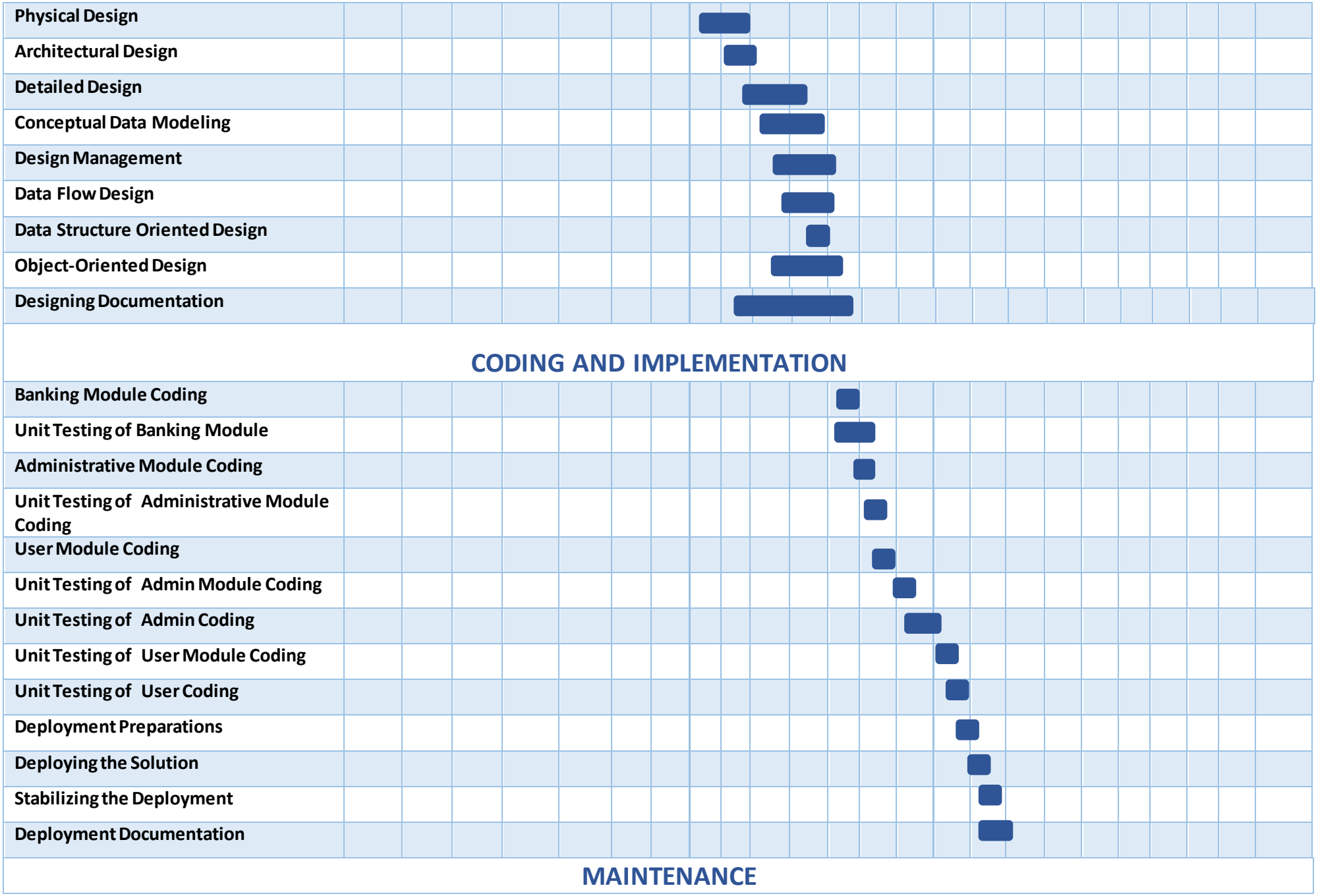
Grady Booch Method

Start Date : August 1st, 2020

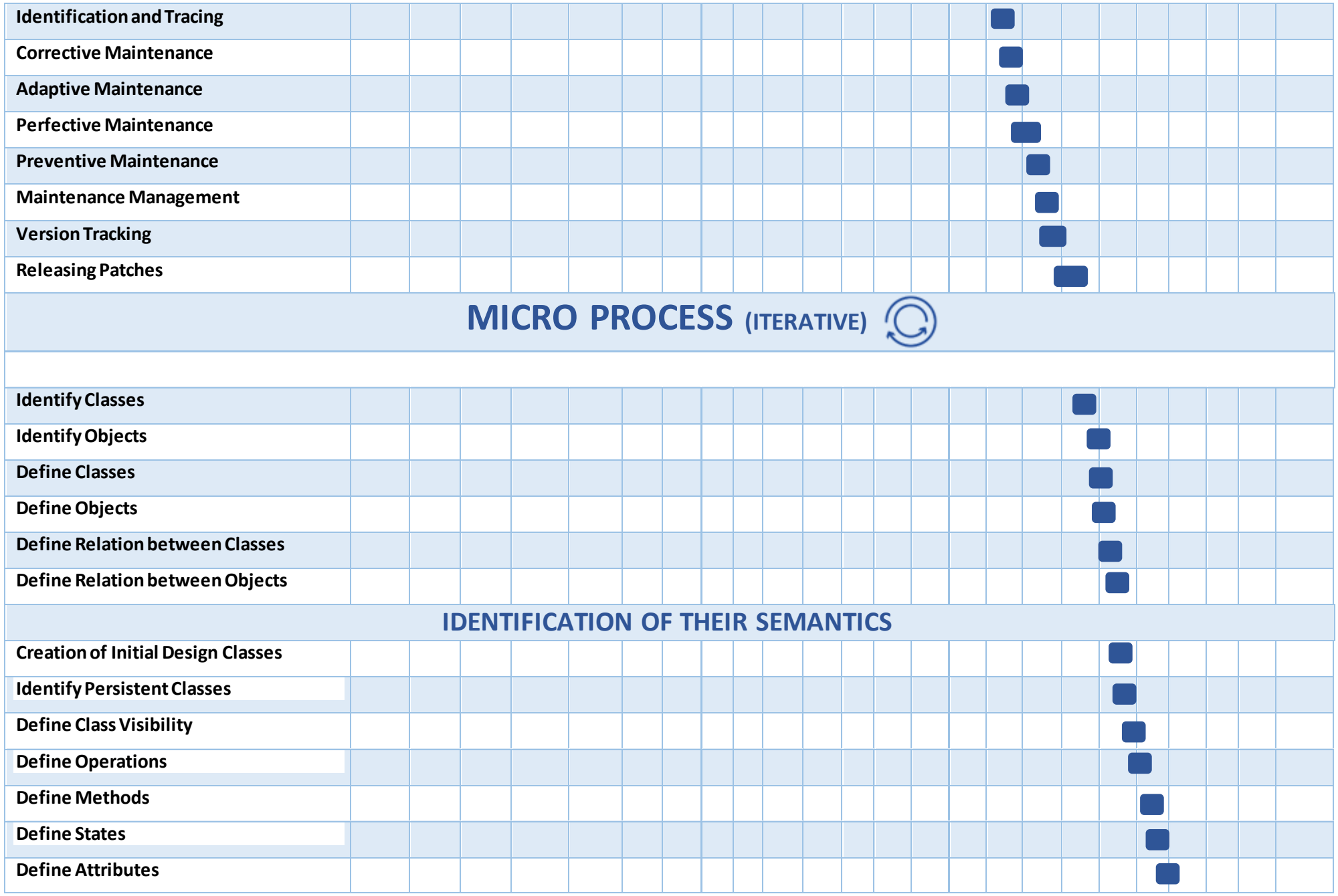
End Date : January 31st, 2021

Plan Activities	August	September	October	November	December	January
MACRO PROCESS						
CONCEPTUALIZATION						
Market Research						


Analyze Industrial Trends
Research on Predefined Models
Analyze Previous Data
Analyze Previous Design
Creating a New Model
Passing Test Cases
Calculation of Model Success
Conceptualization Documentation
ANALYSIS
Feasibility Study
Economic Feasibility
Technical Feasibility
Operational Feasibility
Negotiation and Discussion
Software Requirement Verification
Software Requirement Validation
SRS Documentation
Brainstorming
Analyzing requirements
Requirements modeling
Security Requirements
DESIGN
Identify Design Goals
System Decomposition
Logical Design

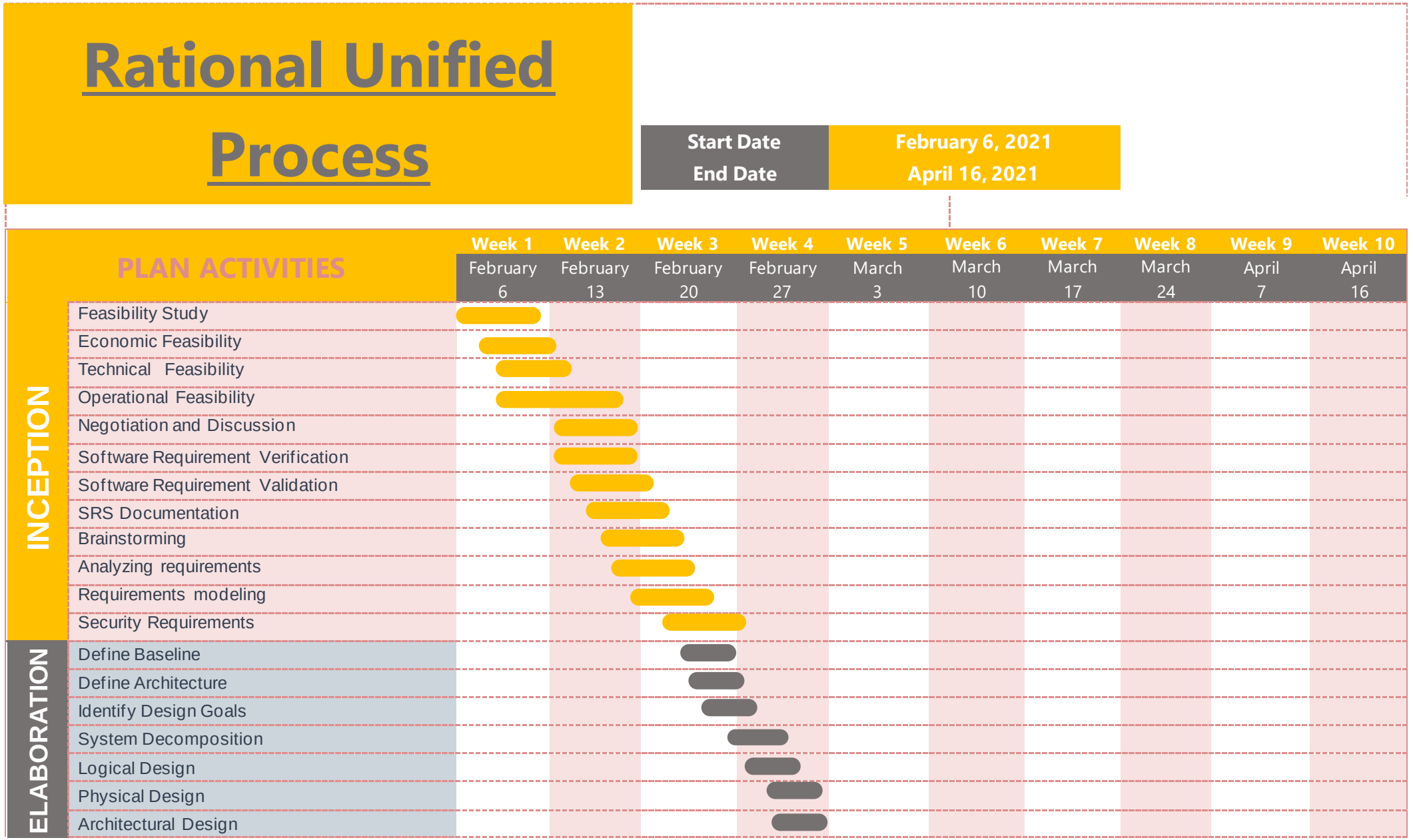

Physical Design
Architectural Design
Detailed Design
Conceptual Data Modeling
Design Management
Data Flow Design
Data Structure Oriented Design
Object-Oriented Design
Designing Documentation
CODING AND IMPLEMENTATION
Banking Module Coding
Unit Testing of Banking Module
Administrative Module Coding
Unit Testing of Administrative Module Coding
User Module Coding
Unit Testing of Admin Module Coding
Unit Testing of Admin Coding
Unit Testing of User Module Coding
Unit Testing of User Coding
Deployment Preparations
Deploying the Solution
Stabilizing the Deployment
Deployment Documentation
MAINTENANCE

Identification and Tracing
Corrective Maintenance
Adaptive Maintenance
Perfective Maintenance
Preventive Maintenance
Maintenance Management
Version Tracking
Releasing Patches
MICRO PROCESS (ITERATIVE)
Identify Classes
Identify Objects
Define Classes
Define Objects
Define Relation between Classes
Define Relation between Objects
IDENTIFICATION OF THEIR SEMANTICS
Creation of Initial Design Classes
Identify Persistent Classes
Define Class Visibility
Define Operations
Define Methods
Define States
Define Attributes



Define Dependencies	
Define Associations	
Define Generalizations	
IDENTIFICATION OF THEIR RELATIONSHIP	
Identify Inheritance	
Single inheritance	
Multi-level inheritance	
Multiple inheritance	
Multipath inheritance	
Hierarchical Inheritance	
Hybrid Inheritance	
IDENTIFICATION OF THEIR INTERFACES AND IMPLEMENTAION	
Identify Interface	
Identify Abstract Classes	
Define Interface	
Define Abstract Classes	

(b)


Rational Unified Process
Start Date
February 6, 2021
End Date
April 16, 2021
PLAN ACTIVITIES
Week 1 February 6
Week 2 February 13
Week 3 February 20
Week 4 February 27
Week 5 March 3
Week 6 March 10
Week 7 March 17
Week 8 March 24
Week 9 April 7
Week 10 April 16
INCEPTION
Feasibility Study
Economic Feasibility
Technical Feasibility
Operational Feasibility
Negotiation and Discussion
Software Requirement Verification
Software Requirement Validation
SRS Documentation
Brainstorming
Analyzing requirements
Requirements modeling
Security Requirements
ELABORATION
Define Baseline
Define Architecture
Identify Design Goals
System Decomposition
Logical Design
Physical Design
Architectural Design


Detailed Design
Conceptual Data Modeling
Design Management
Data Flow Design
Data Structure Oriented Design
Object-Oriented Design
Designing Documentation
CONSTRUCTION
Banking Module Coding
Unit Testing of Banking Module
Administrative Module Coding
Unit Testing of Administrative Module
User Module Coding
Unit Testing of Admin Module Coding
Unit Testing of Admin Coding
Unit Testing of User Module Coding
Front-end Coding
Back-end Coding
Database Coding
Unit Testing of Database Coding
Coding Documentation
TRANSITION
Identification and Tracing
Corrective Maintenance
Adaptive Maintenance
Perfective Maintenance
Preventive Maintenance
Installation and Activation
Implement enhancements
Interface with other systems.
Maintenance Management
Version Tracking
Releasing Patches
Retirement of Old Version.

(c)

Waterfall Project Plan

Start Date	January 6, 2020
End Date	March 16, 2020

	PLAN ACTIVITIES	Week 1 January 6	Week 2 January 13	Week 3 January 20	Week 4 January 27	Week 5 February 3	Week 6 February 10	Week 7 February 17	Week 8 February 24	Week 9 March 7	Week 10 March 16
REQUIREMENT GATHERING & ANALYSIS	Feasibility Study										
	Economic Feasibility										
	Technical Feasibility										
	Operational Feasibility										
	Negotiation and Discussion										
	Software Requirement Verification										
	Software Requirement Validation										
	SRS Documentation										
SYSTEM DESIGN	Identify Design Goals										
	System Decomposition										
	Logical Design										
	Physical Design										
	Architectural Design										
	Detailed Design										
	Conceptual Data Modeling										
	Design Management										
IMPLEMENTATI	Banking Module Coding										
	Unit Testing of Banking Module										
	Administrative Module Coding										

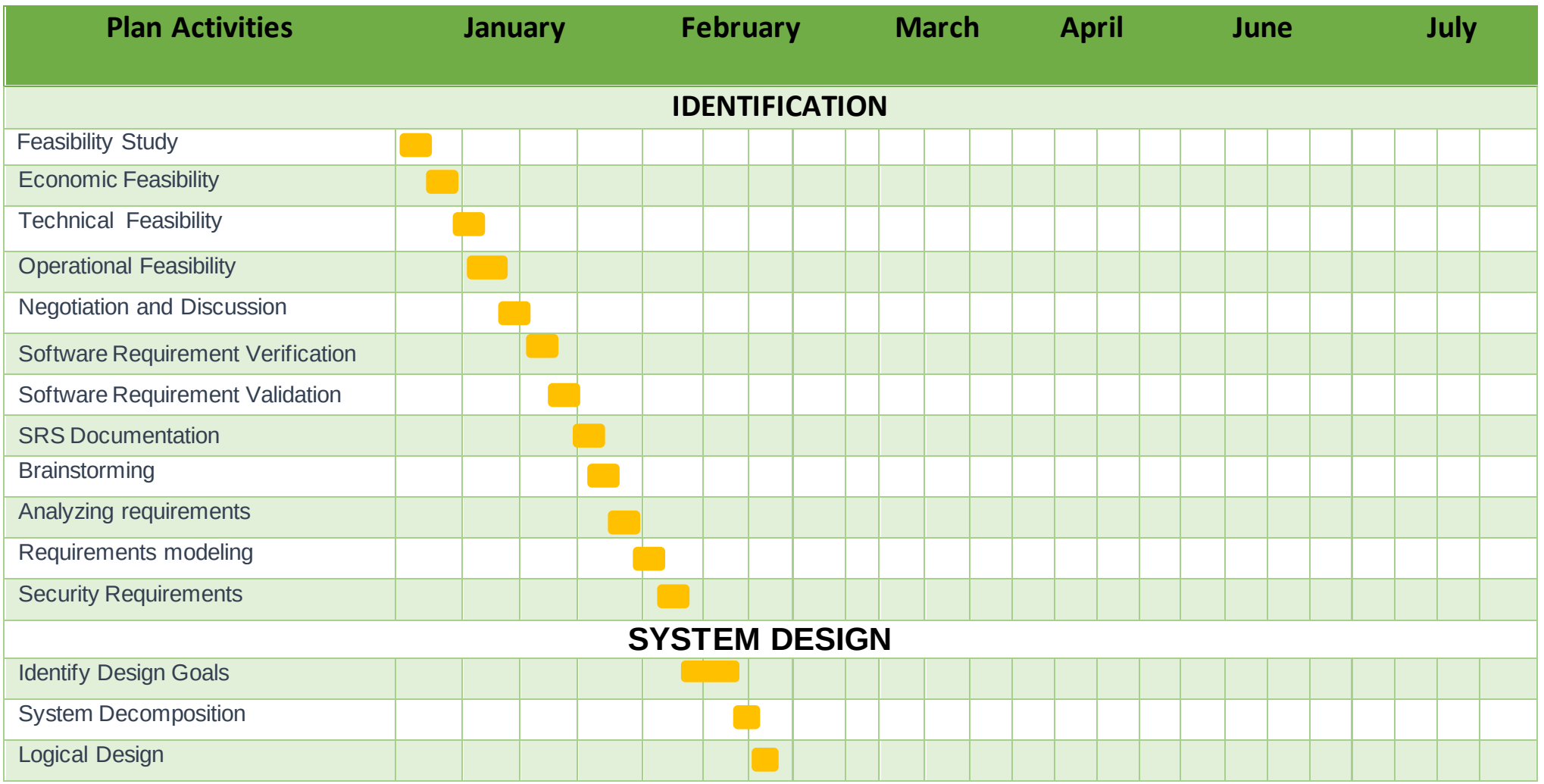

Unit Testing of Administrative Module
User Module Coding
Unit Testing of Admin Module Coding
Unit Testing of Admin Coding
Unit Testing of User Module Coding
Coding Documentation
INTEGRATION & TSETING
Integration of Components
Integration of Modules
Integration Testing
System Testing
Bottom-up Integration Testing
Incremental Testing
Big Bang Testing
Acceptance Testing
DEPLOYMENT OF SYSTEM
Creating Operation Procedures
Deployment Preparations
Deploying the Solution
Core Technology Deployment
Site Deployments
Stabilizing the Deployment
Transferring Ownership to Operations Team
Closing the Deploying Phase
MAINTENANCE
Identification and Tracing
Corrective Maintenance
Adaptive Maintenance
Perfective Maintenance
Preventive Maintenance
Installation and Activation
Maintenance Management
Version Tracking
Releasing Patches

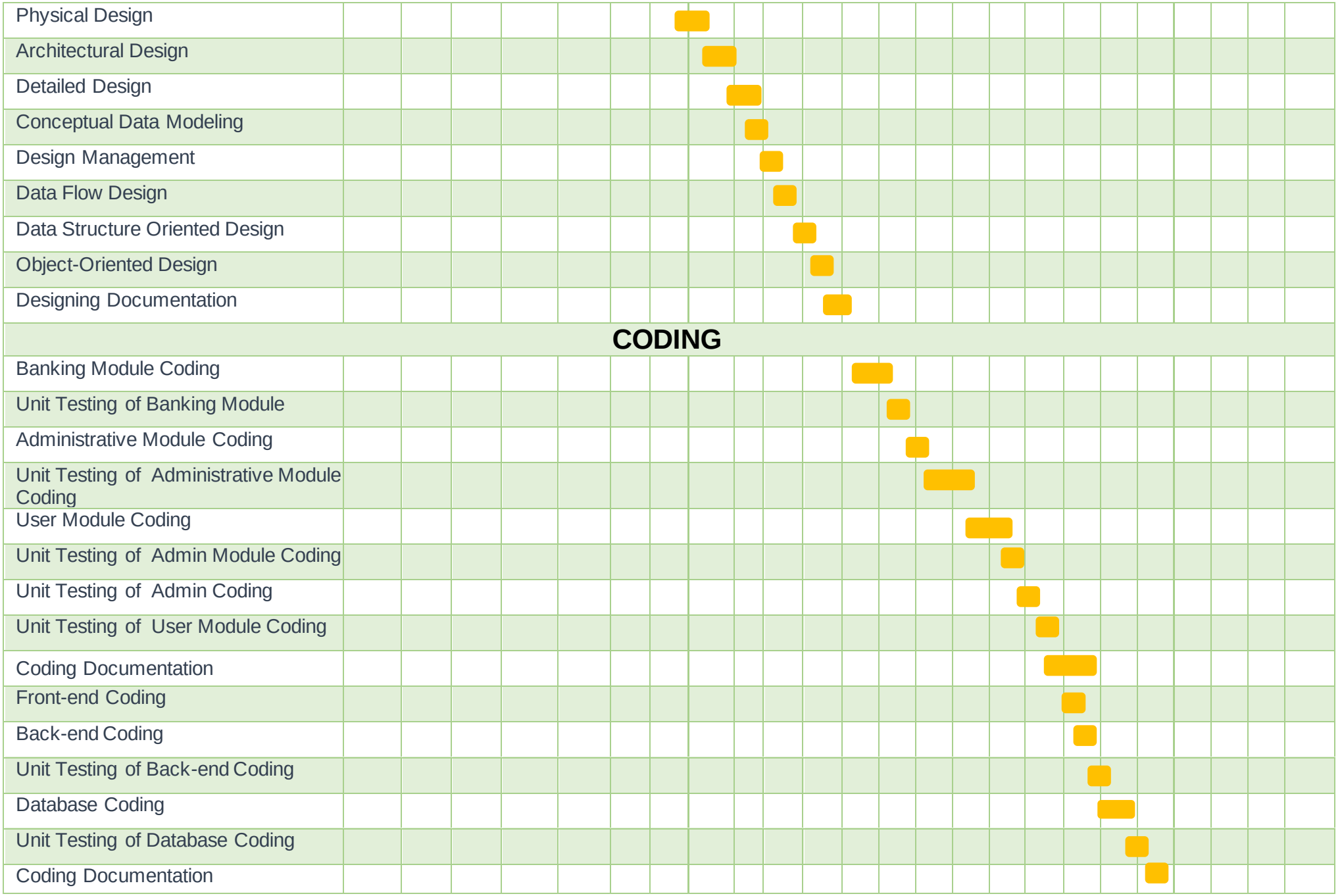
(d)

Spiral Model (ITERATIVE)

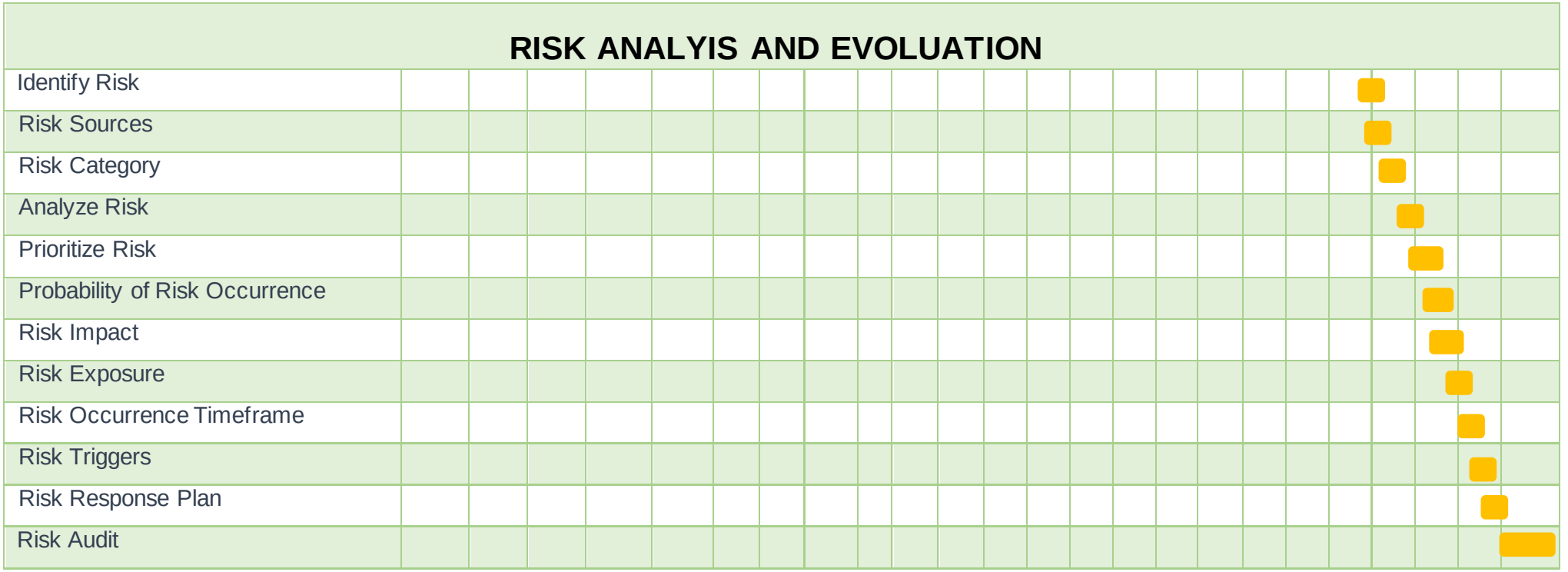
Start Date : January 1st, 2020

End Date : Julv 31st. 2020




Physical Design
Architectural Design
Detailed Design
Conceptual Data Modeling
Design Management
Data Flow Design
Data Structure Oriented Design
Object-Oriented Design
Designing Documentation
CODING
Banking Module Coding
Unit Testing of Banking Module
Administrative Module Coding
Unit Testing of Administrative Module Coding
User Module Coding
Unit Testing of Admin Module Coding
Unit Testing of Admin Coding
Unit Testing of User Module Coding
Coding Documentation
Front-end Coding
Back-end Coding
Unit Testing of Back-end Coding
Database Coding
Unit Testing of Database Coding
Coding Documentation

RISK ANALYIS AND EVOLUATION



Q2. Apply online tools of **COCOMO I** and **COCOMO II** on the following projects: [2.5 marks each]

a. Take every factor as Average and LOC as 50K

b. Take every factor as Simple and LOC as 50K

c. Take every factor as Complex , Highly Influence and LOC as 50K

d. Take every factor as Average and FP = 500

Submit the report of these system from the tool.

(a)

YOUR BASIC COCOMO RESULTS!!

MODE	"A" variable	"B" variable	"C" variable	"D" variable	KLOC	EFFORT, (in person/months)	DURATION, (in months)	STAFFING, (recommended)
organic	2.4	1.05	2.5	0.38	50	145.92501487903888	16.60769315759501	8.786591460615025

Explanation: The coefficients are set according to the project mode selected on the previous page, (as per Boehm,81). The final estimates are determined in the following manner:

effort =a*KLOC^{b}, in person/months, with KLOC = lines of code, (in the thousands), and:

duration =c*effort^{d}, finally:

staffing =effort/duration

For further reading, see Boehm, "Software Engineering Econimics",(81)

WARNING: If you see "NaN" in any field above, you have entered an **INVALID** value for KLOC!! Hit the "BACK" button on your browser, hit the "RESET" button, and enter a **DECIMAL NUMBER** in the KLOC input text box!

Thank you, and happy software engineering!

(b)

YOUR BASIC COCOMO RESULTS!!								
MODE	**"A" variable**	**"B" variable**	**"C" variable**	**"D" variable**	**KLOC**	**EFFORT, (in person/months)**	**DURATION, (in months)**	**STAFFING, (recommended)**
semi-detached	3	1.12	2.5	0.35	50	239.8654292791274	17.018790395683062	14.09415262203188

Explanation: The coefficients are set according to the project mode selected on the previous page, (as per Boehm,81). The final estimates are determined in the following manner:

effort =a*KLOC^{b}, in person/months, with KLOC = lines of code, (in the thousands), and:

duration =c*effort^{d}, finally:

staffing =effort/duration

For further reading, see Boehm, "Software Engineering Econimics",(81)

WARNING: If you see "NaN" in any field above, you have entered an **INVALID** value for KLOC!! Hit the "BACK" button on your browser, hit the "RESET" button, and enter a **DECIMAL NUMBER** in the KLOC input text box!

Thank you, and happy software engineering!

(c)

YOUR BASIC COCOMO RESULTS!!

MODE	"A" variable	"B" variable	"C" variable	"D" variable	KLOC	EFFORT, (in person/months)	DURATION, (in months)	STAFFING, (recommended)
embedded	3.6	1.2	2.5	0.32	50	393.61034661958	16.918477984655127	23.265115631357634

Explanation: The coefficients are set according to the project mode selected on the previous page, (as per Boehm,81). The final estimates are determined in the following manner:

effort =a*KLOC^{b}, in person/months, with KLOC = lines of code, (in the thousands), and:

duration =c*effort^{d}, finally:

staffing =effort/duration

For further reading, see Boehm, "Software Engineering Econimics",(81)

WARNING: If you see "NaN" in any field above, you have entered an **INVALID** value for KLOC!! Hit the "BACK" button on your browser, hit the "RESET" button, and enter a **DECIMAL NUMBER** in the KLOC input text box!

Thank you, and happy software engineering!

(d)

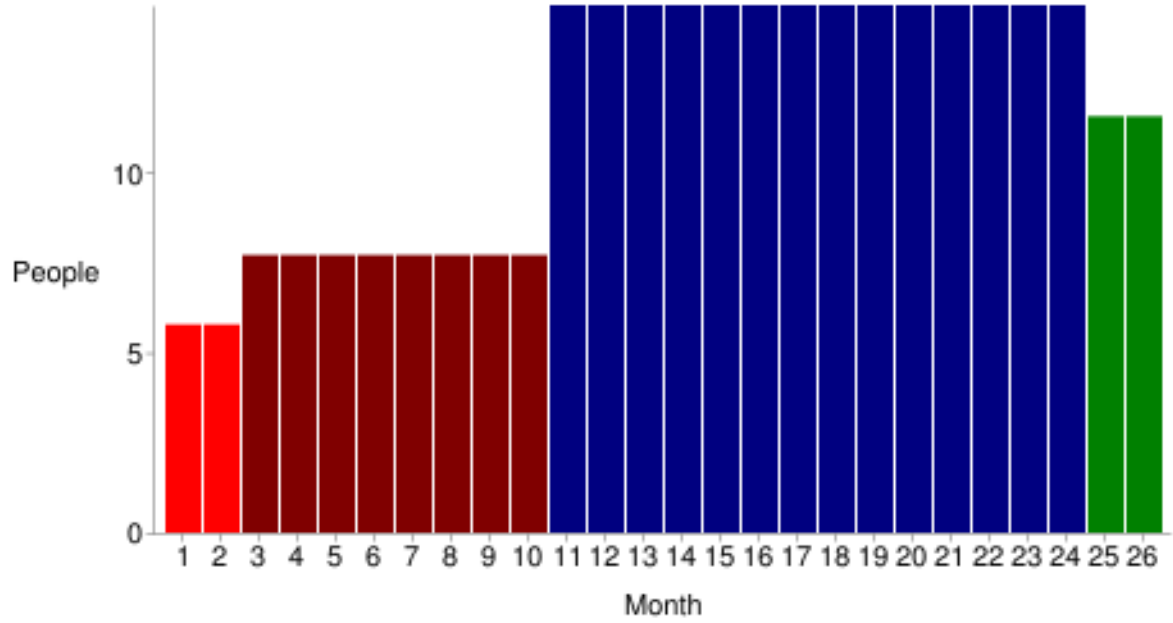
Results

Software Development (Elaboration and Construction)

Effort = 284.8 Person-months
Schedule = 23.7 Months
Cost = $284841

Total Equivalent Size = 64000 SLOC
Effort Adjustment Factor (EAF) = 1.00

Staffing Profile



Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	17.1	3.0	5.8	$17090
Elaboration	68.4	8.9	7.7	$68362
Construction	216.5	14.8	14.6	$216480
Transition	34.2	3.0	11.5	$34181

Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	2.4	8.2	21.6	4.8
Environment/CM	1.7	5.5	10.8	1.7
Requirements	6.5	12.3	17.3	1.4
Design	3.2	24.6	34.6	1.4
Implementation	1.4	8.9	73.6	6.5
Assessment	1.4	6.8	52.0	8.2
Deployment	0.5	2.1	6.5	10.3