



Software Process Modeling and Simulation

Sadaf Mustafiz
School of Computer Science
McGill University

Winter 2003



Outline

- Software Process Entities
- Example EPM
- Process Model (re-modeled)
- Unconstrained Process Model
- Scheduling Considerations
- Simulation
- Results



Software Process Entities

Some obvious entities are:

- Deliverable code
- Users' installation and operation manuals
- Requirements documents
- Design
- Test cases and procedures



Example EPM (1)

- Modeled using a commercially available software system called STATEMATE.
- Focuses on behavioural modeling perspective
- Approach to behavioural modeling utilizes statecharts



Example EPM (2)

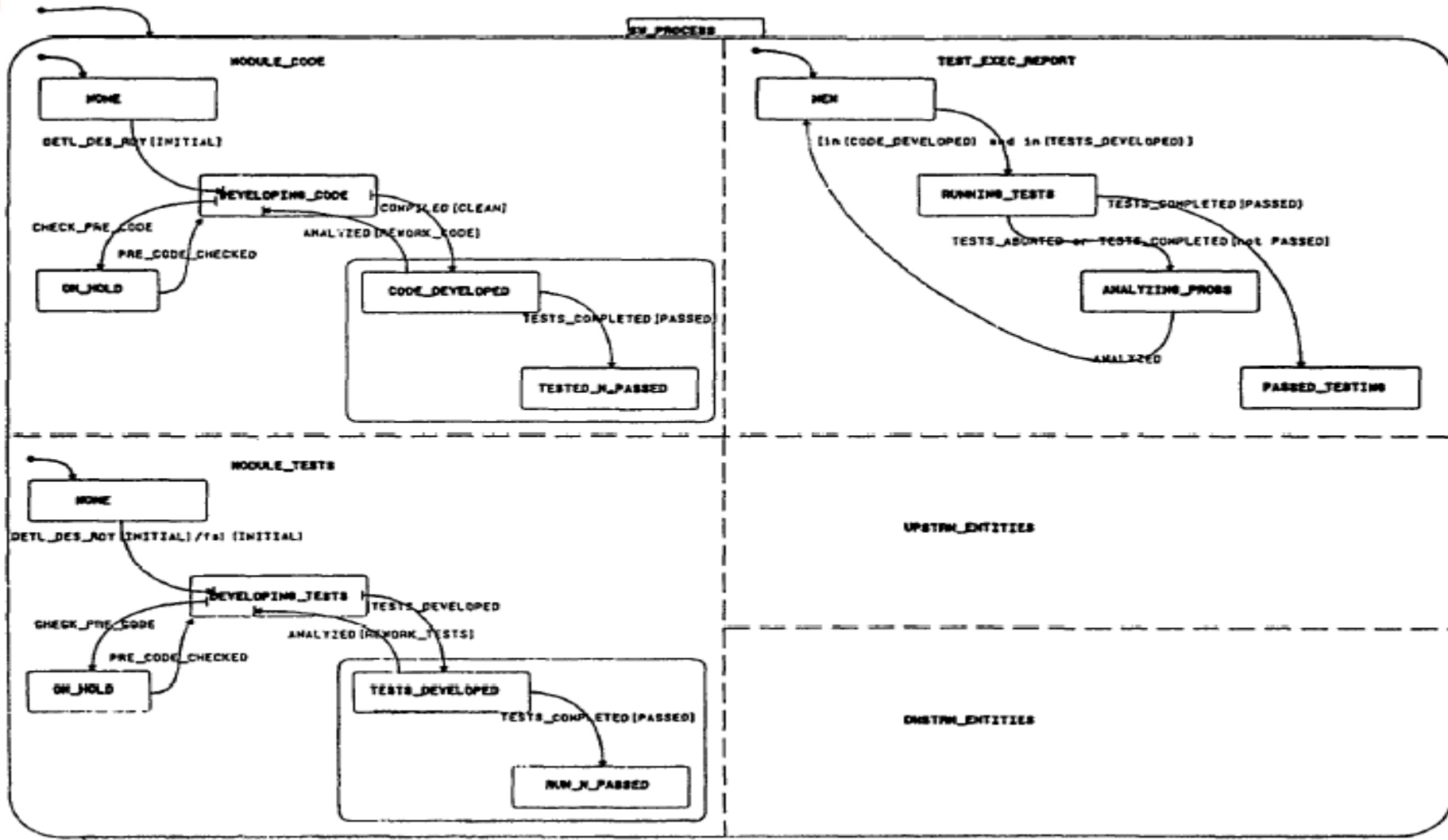
Considering the activities occurring between the time when

1. detailed design for the module has been developed, and
2. the module has successfully passed unit testing.

Three entities of interest:

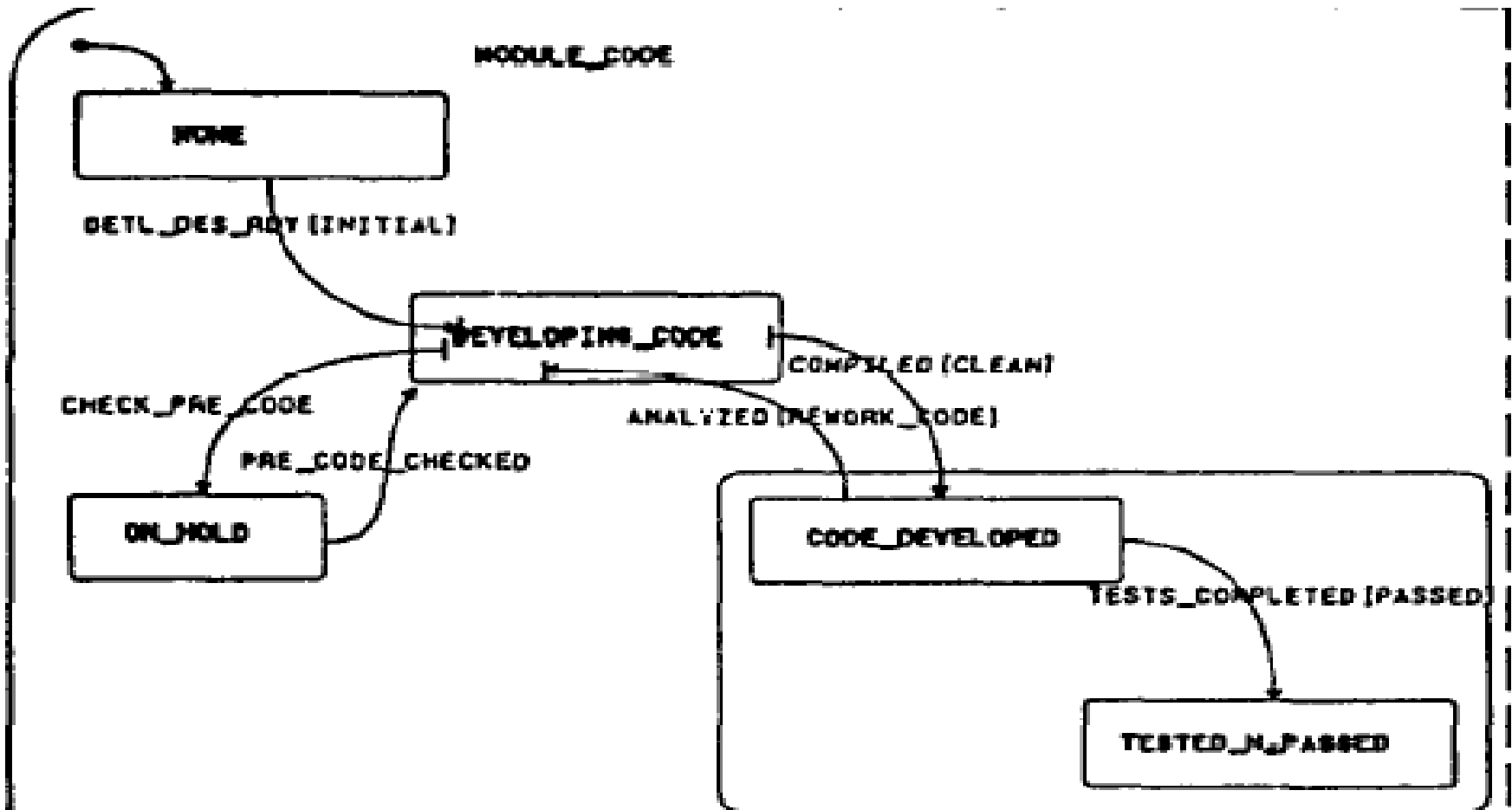
1. Module code
2. Unit tests for the module
3. Test execution and analysis results

Basic EPM Example (3)



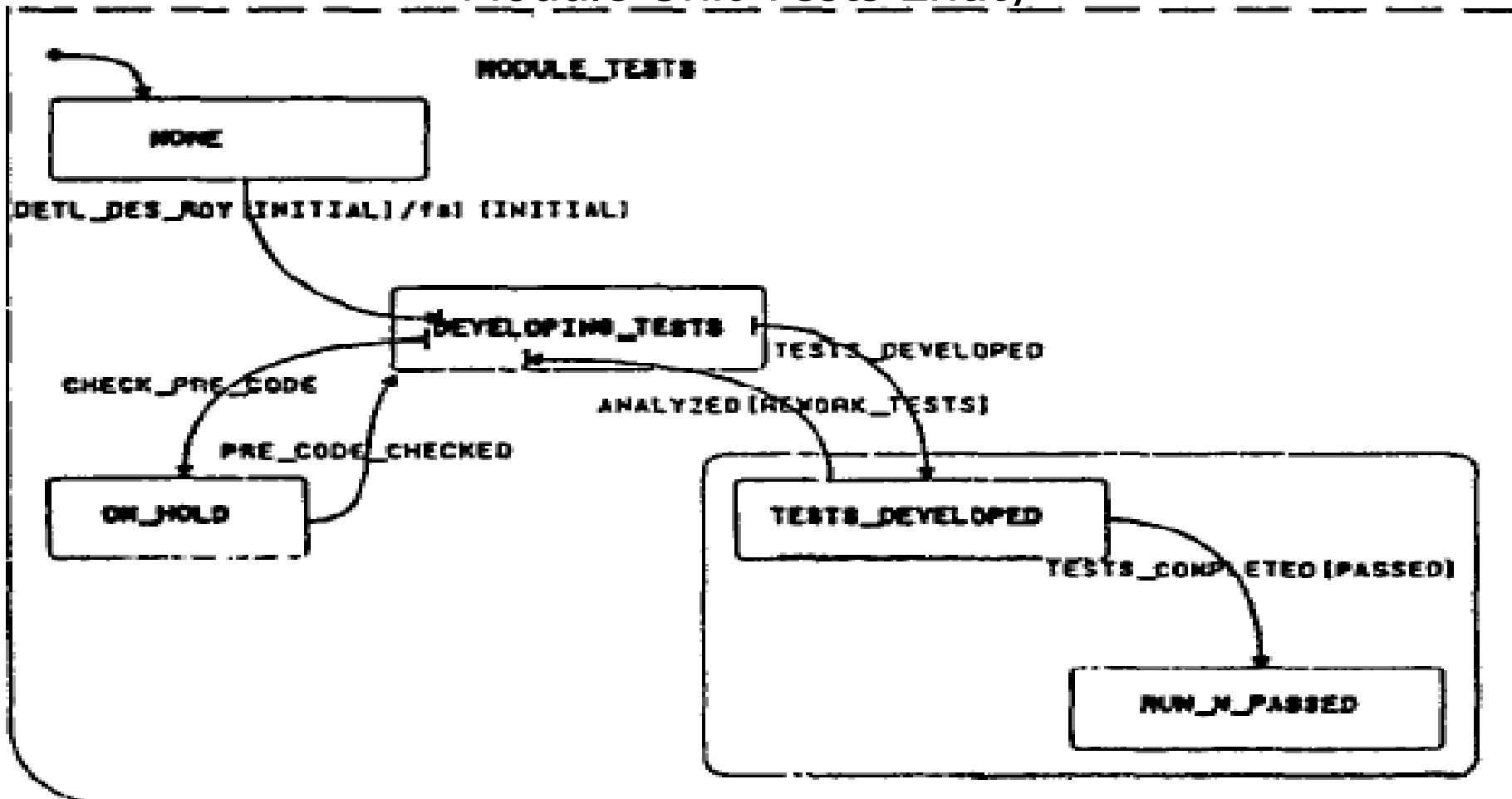
Example EPM (4)

Module Code Entity



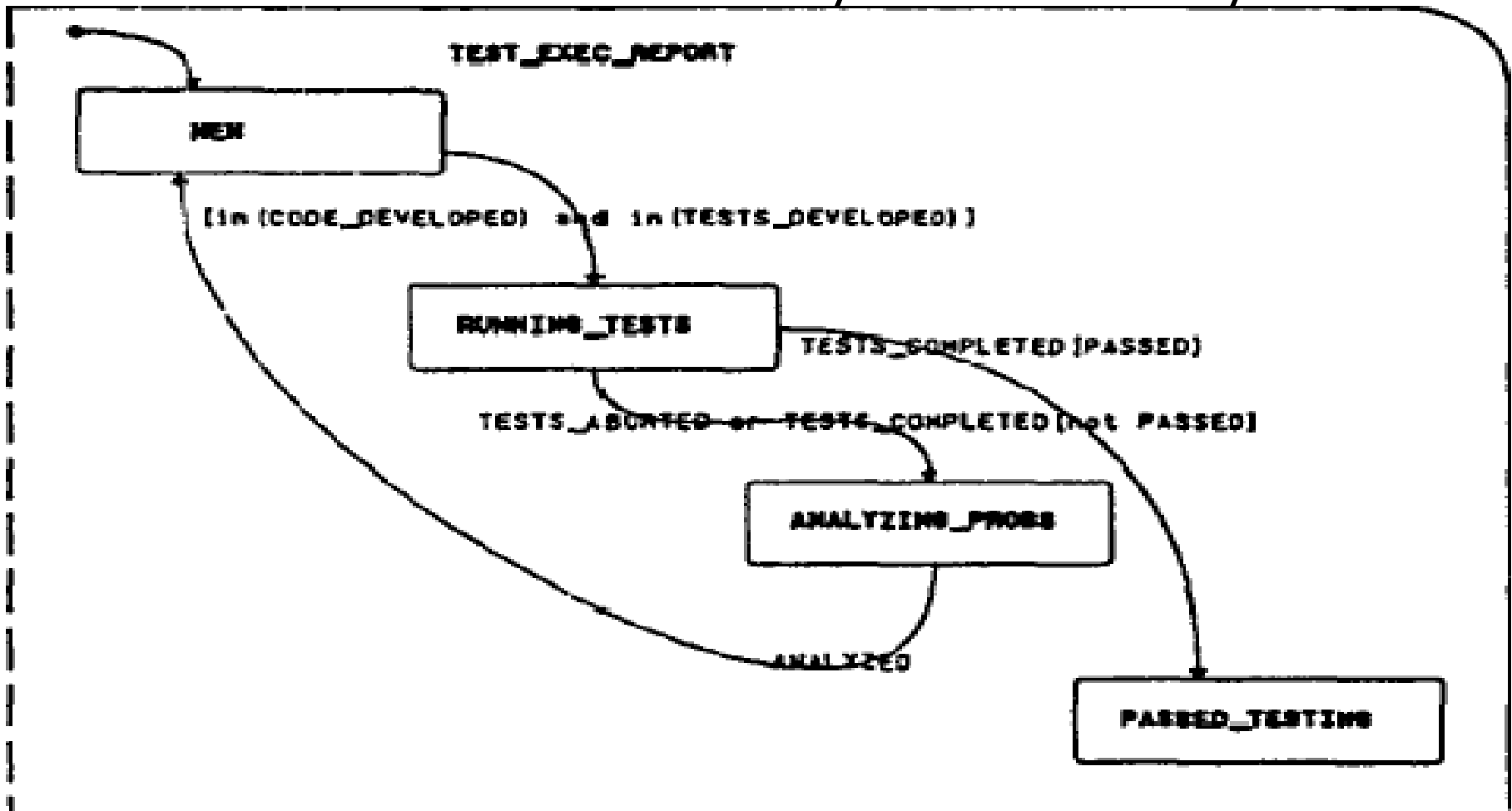
Example EPM (5)

Module Unit Tests Entity



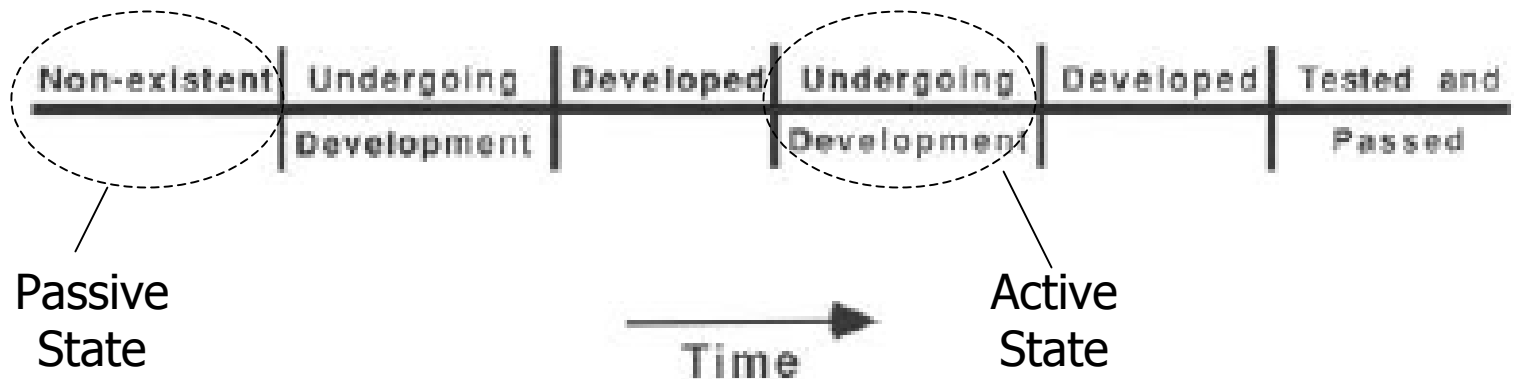
Example EPM (6)

Test Execution and Analysis Results Entity



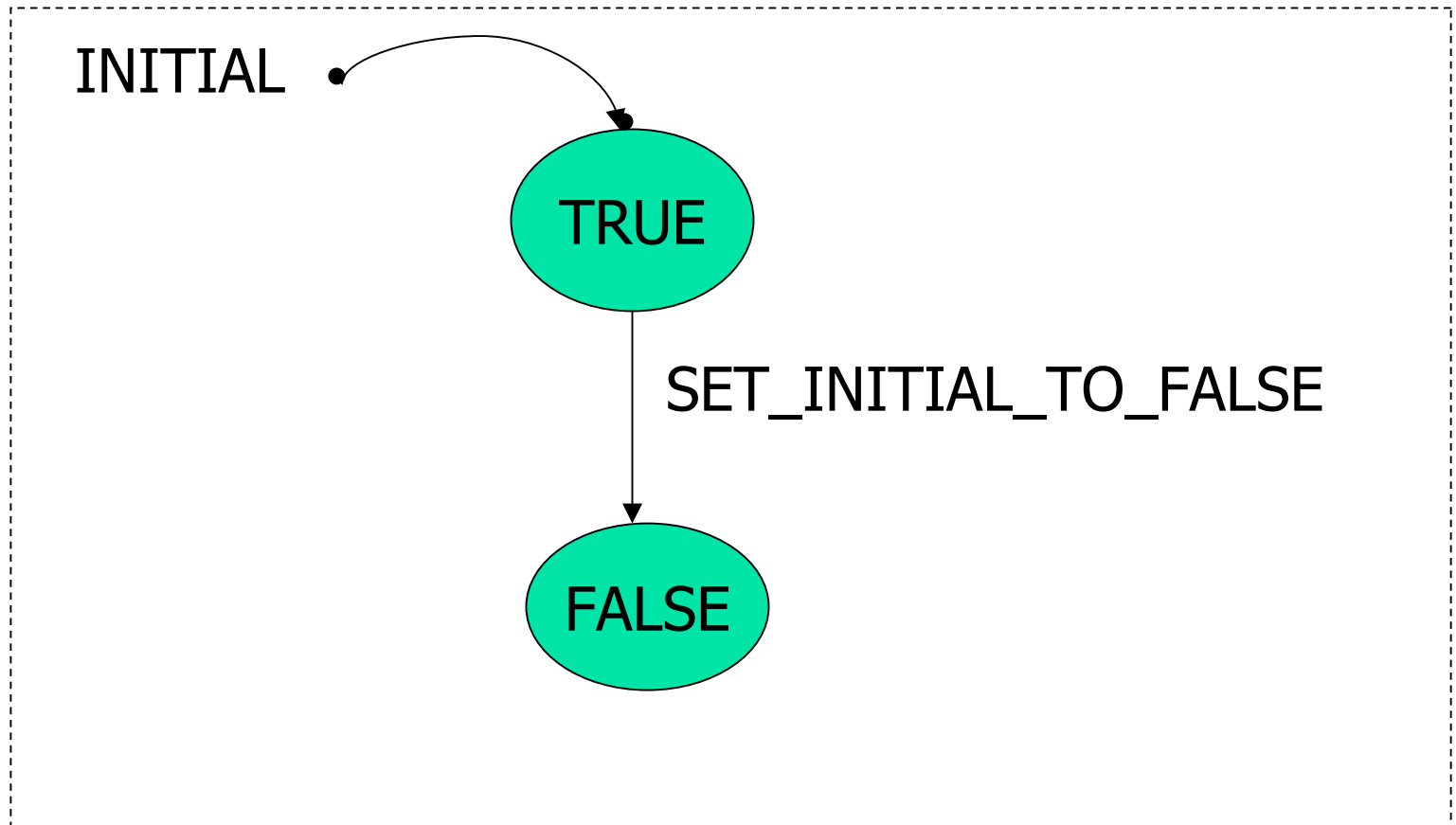
Example EPM (7)

Example Time Line for Module Code Entity

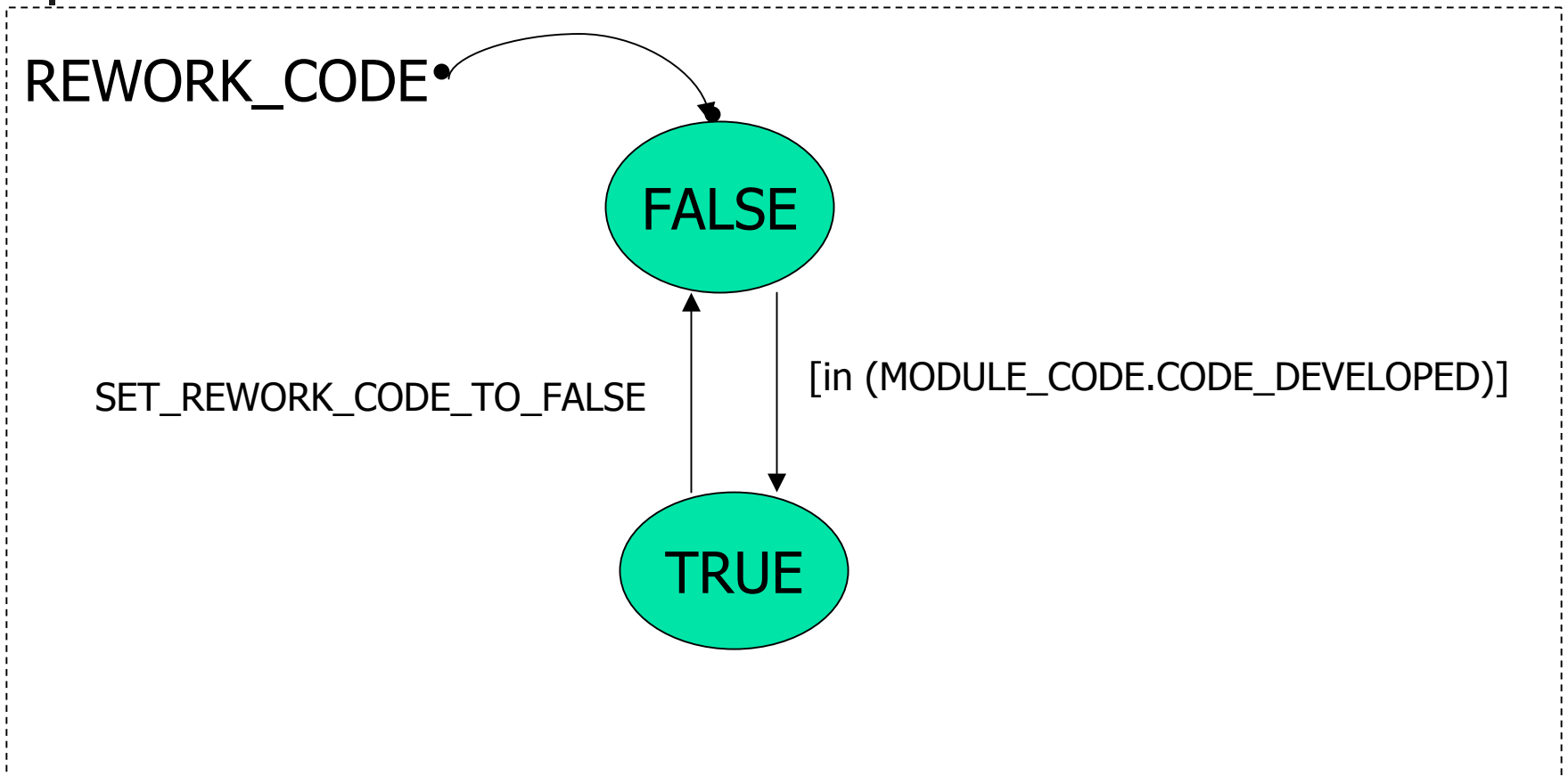


- Entities remain for a non-zero time in each state.
- Transitions take negligible time.
- In the life span of an entity, it must always be in some state.

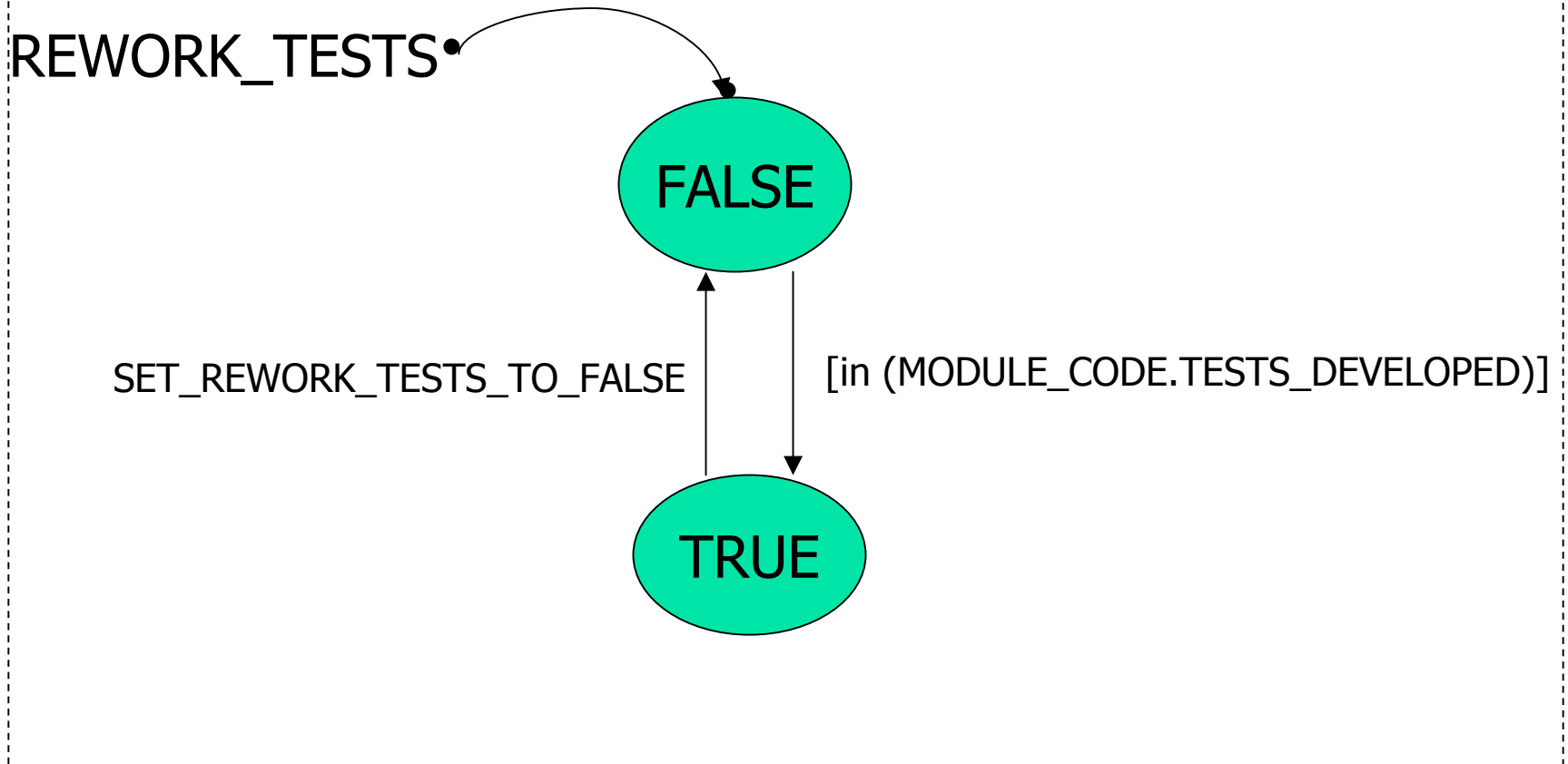
Process Model: Re-modeled (1)



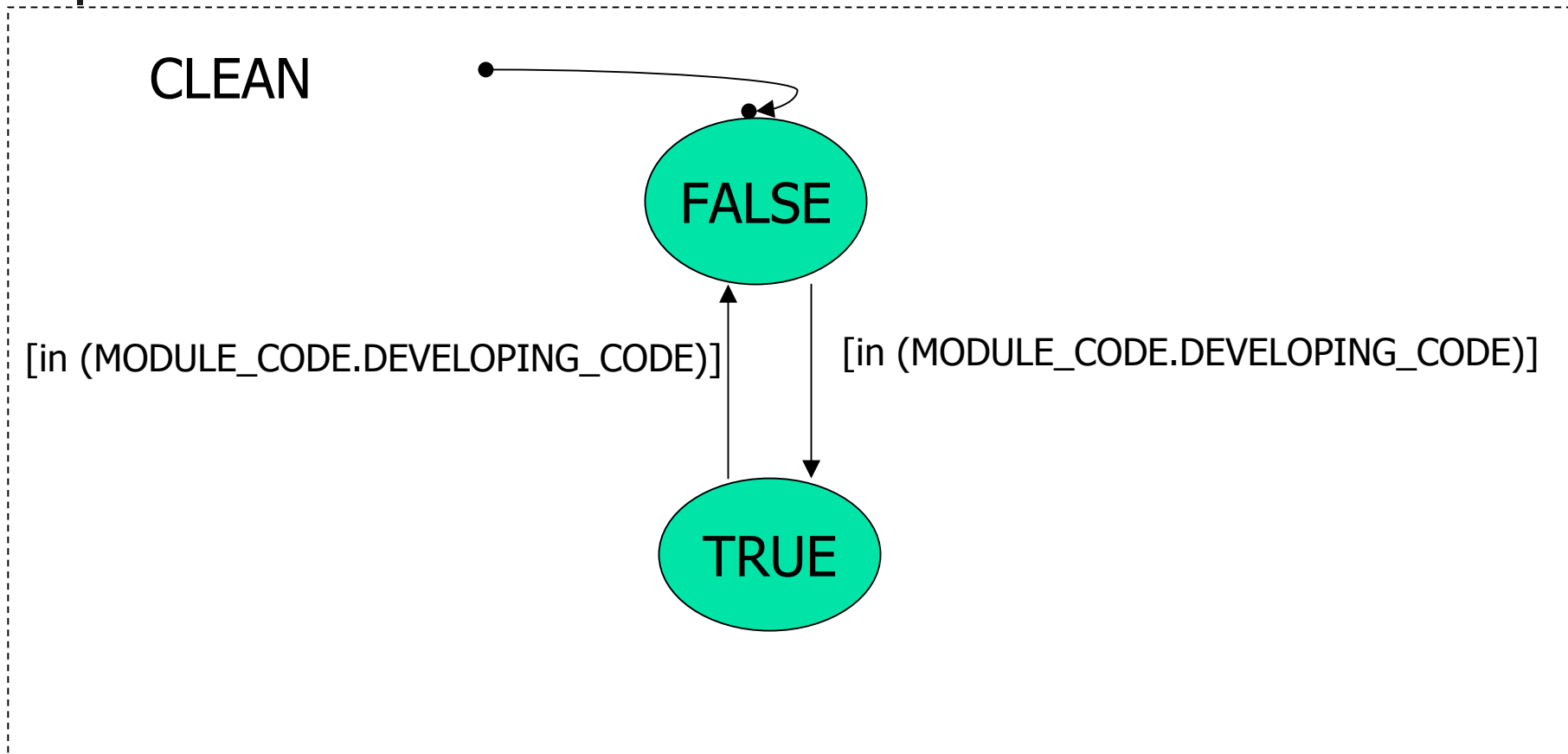
Process Model: Re-modeled (2)



Process Model: Re-modeled (3)

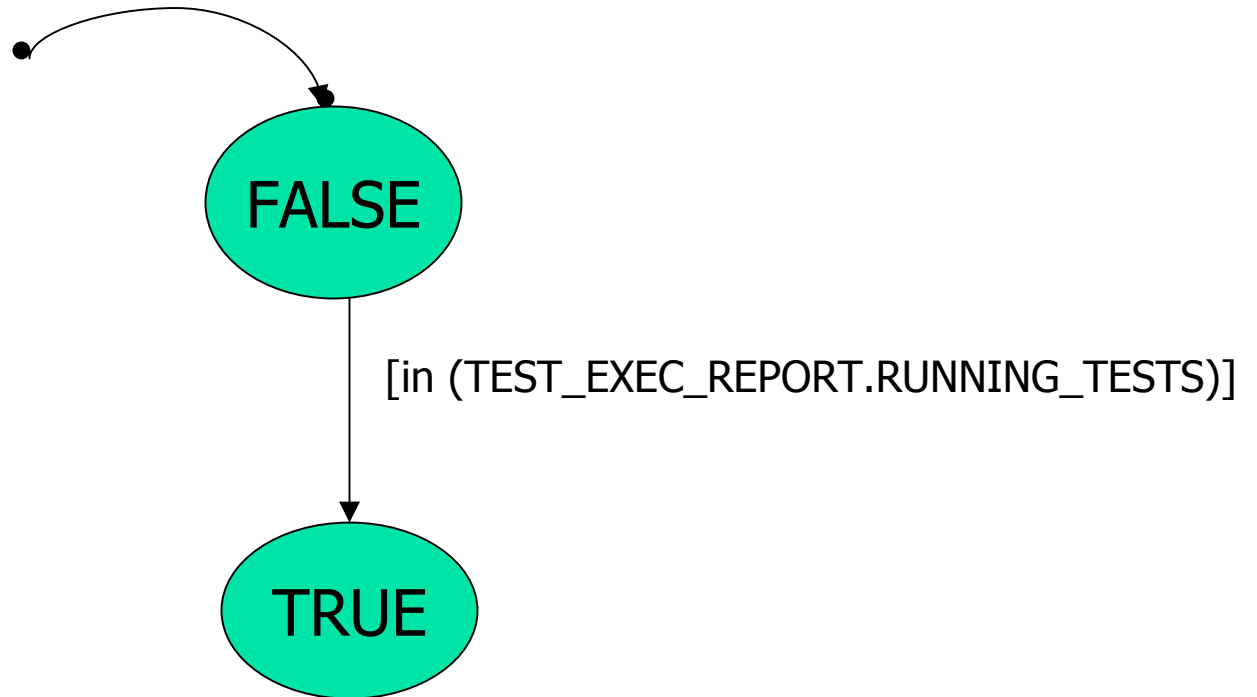


Process Model: Re-modeled (4)

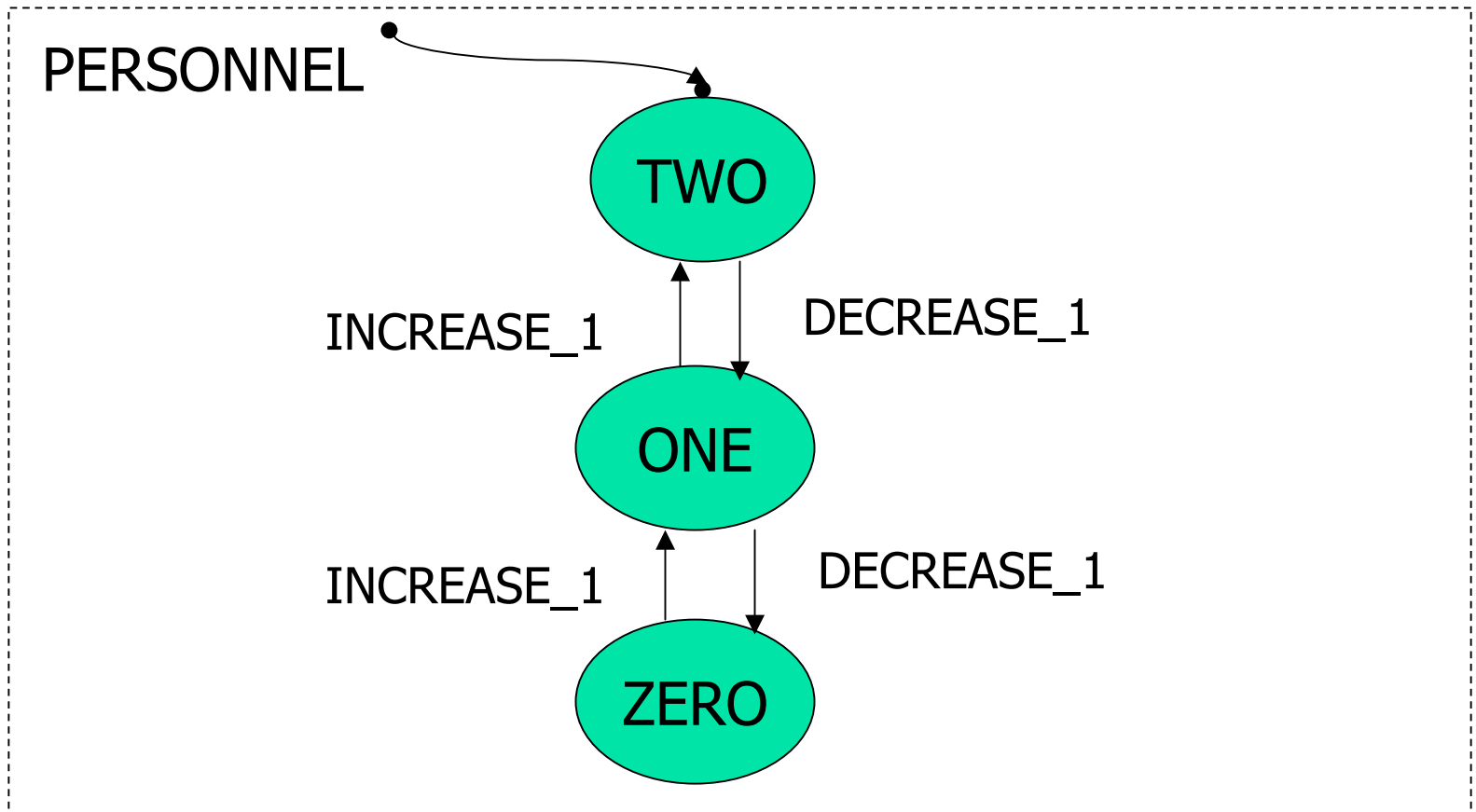


Process Model: Re-modeled (5)

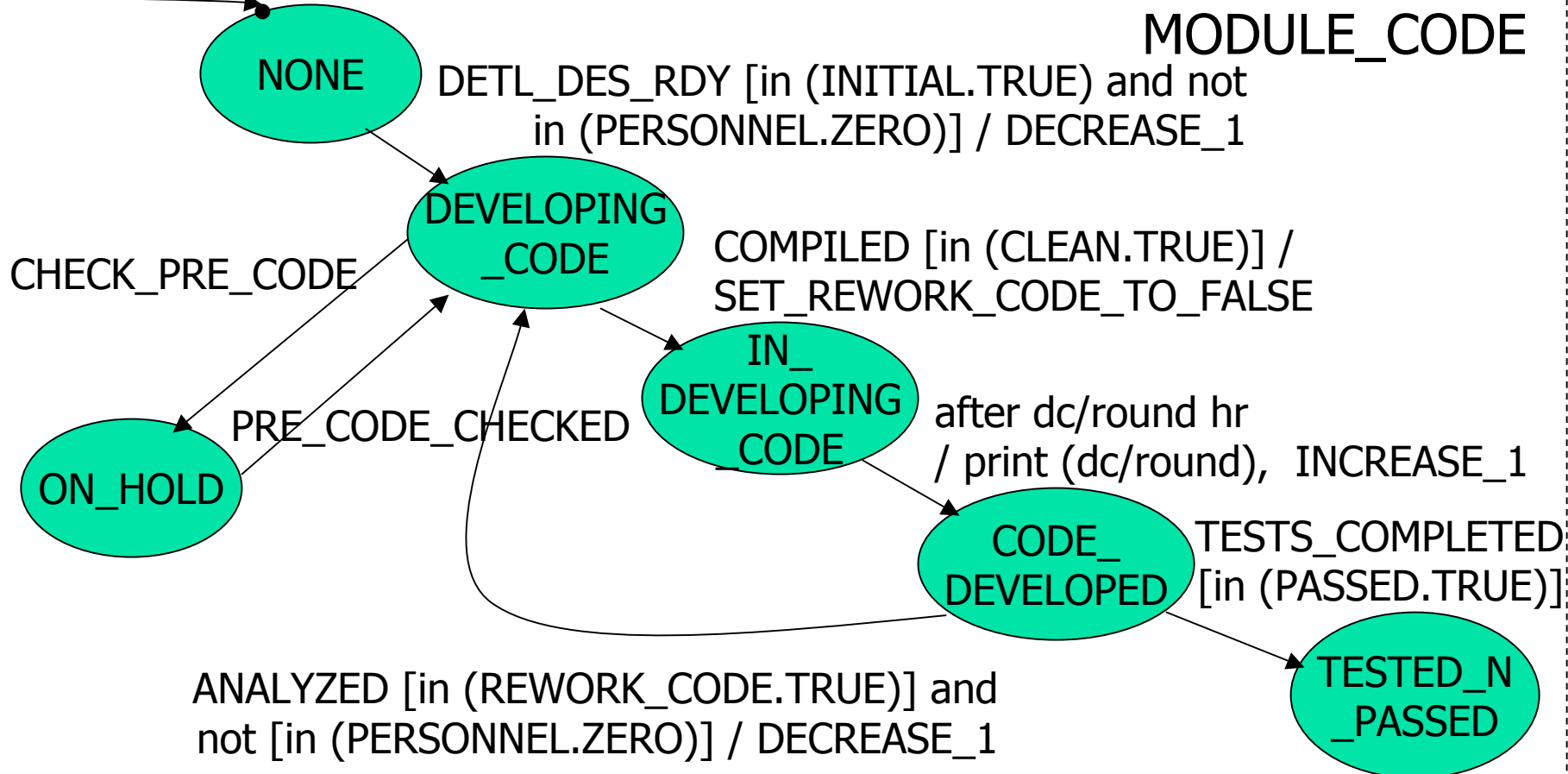
PASSED



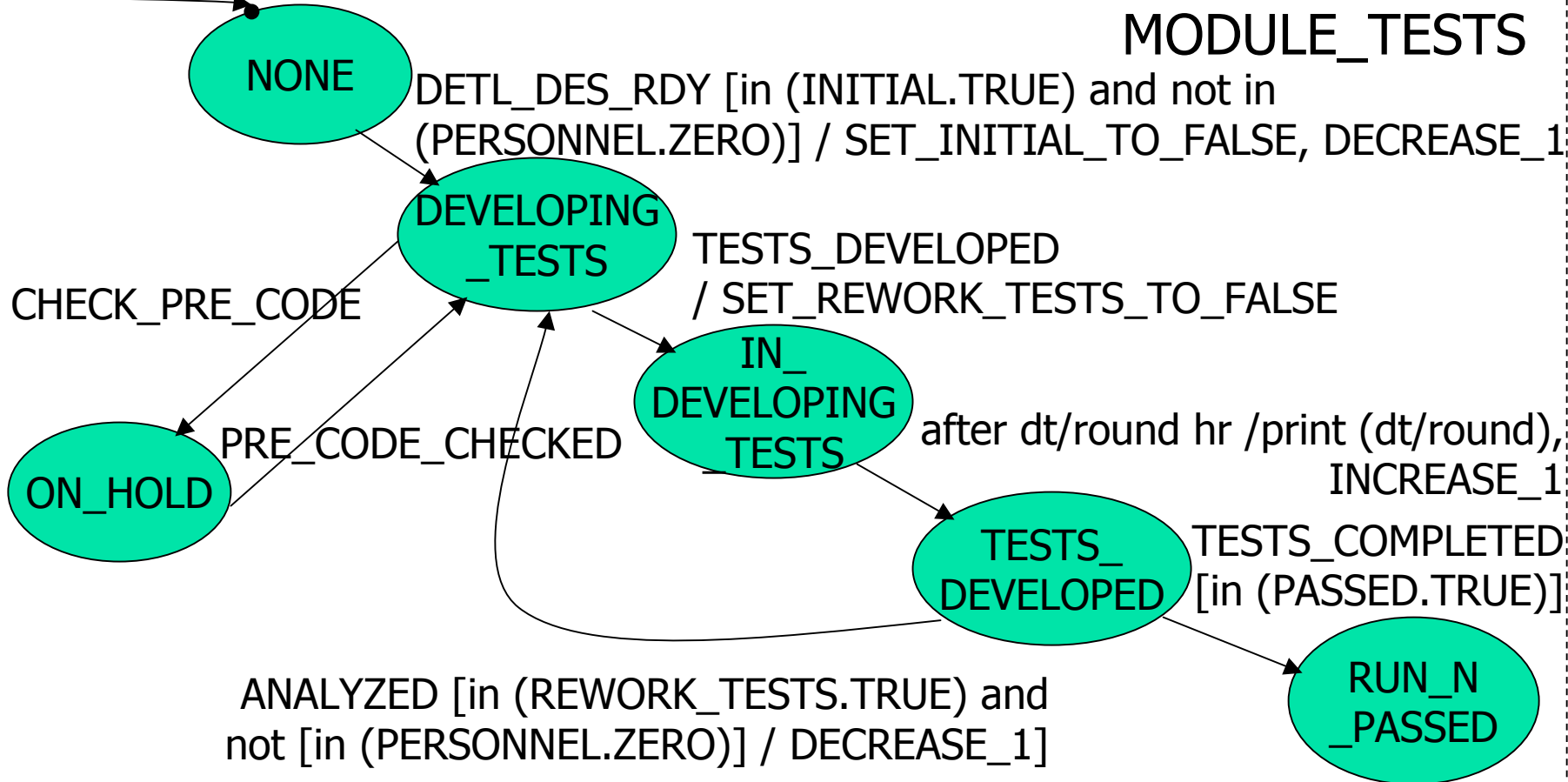
Process Model: Re-modeled (6)



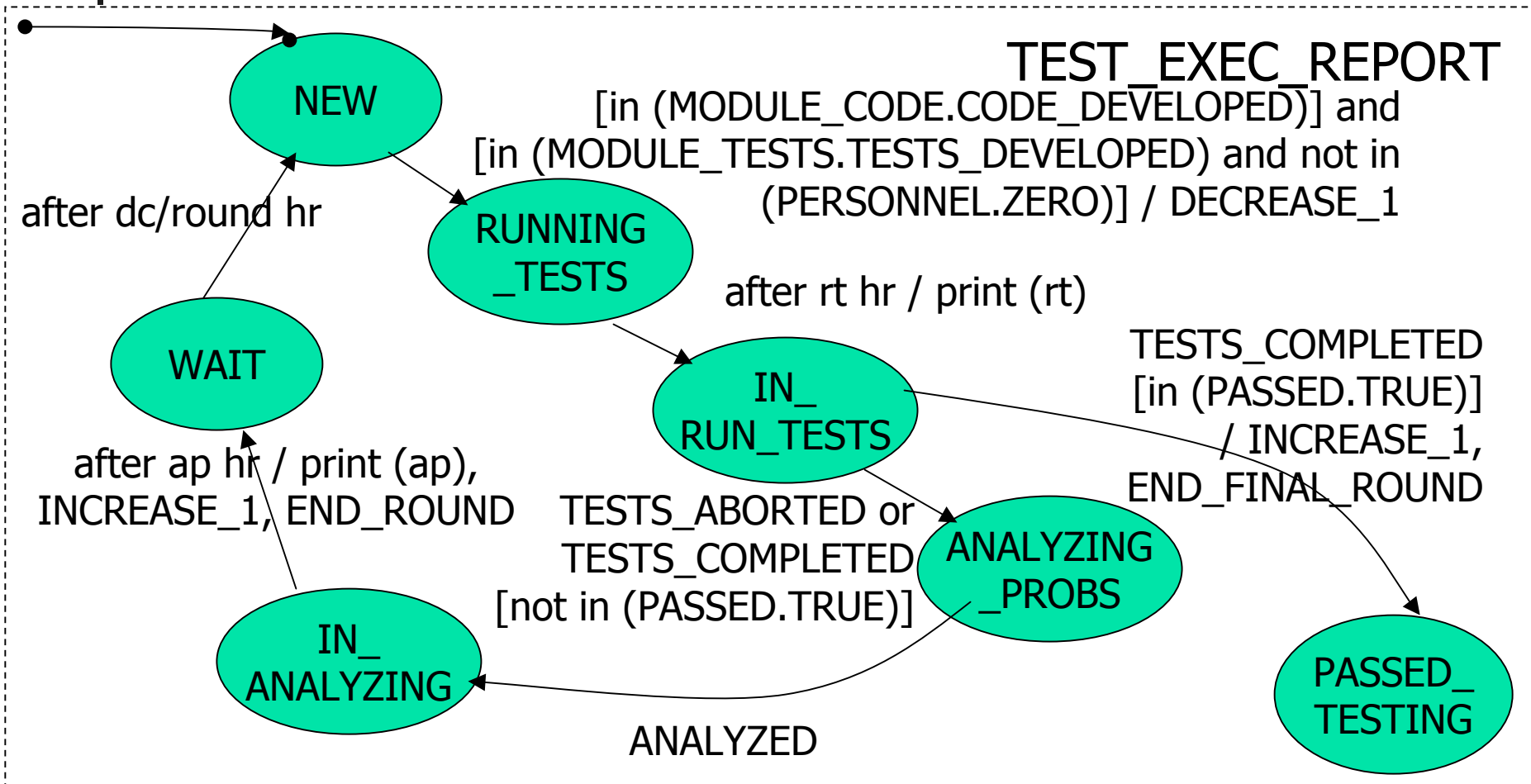
Process Model: Re-modeled (7)



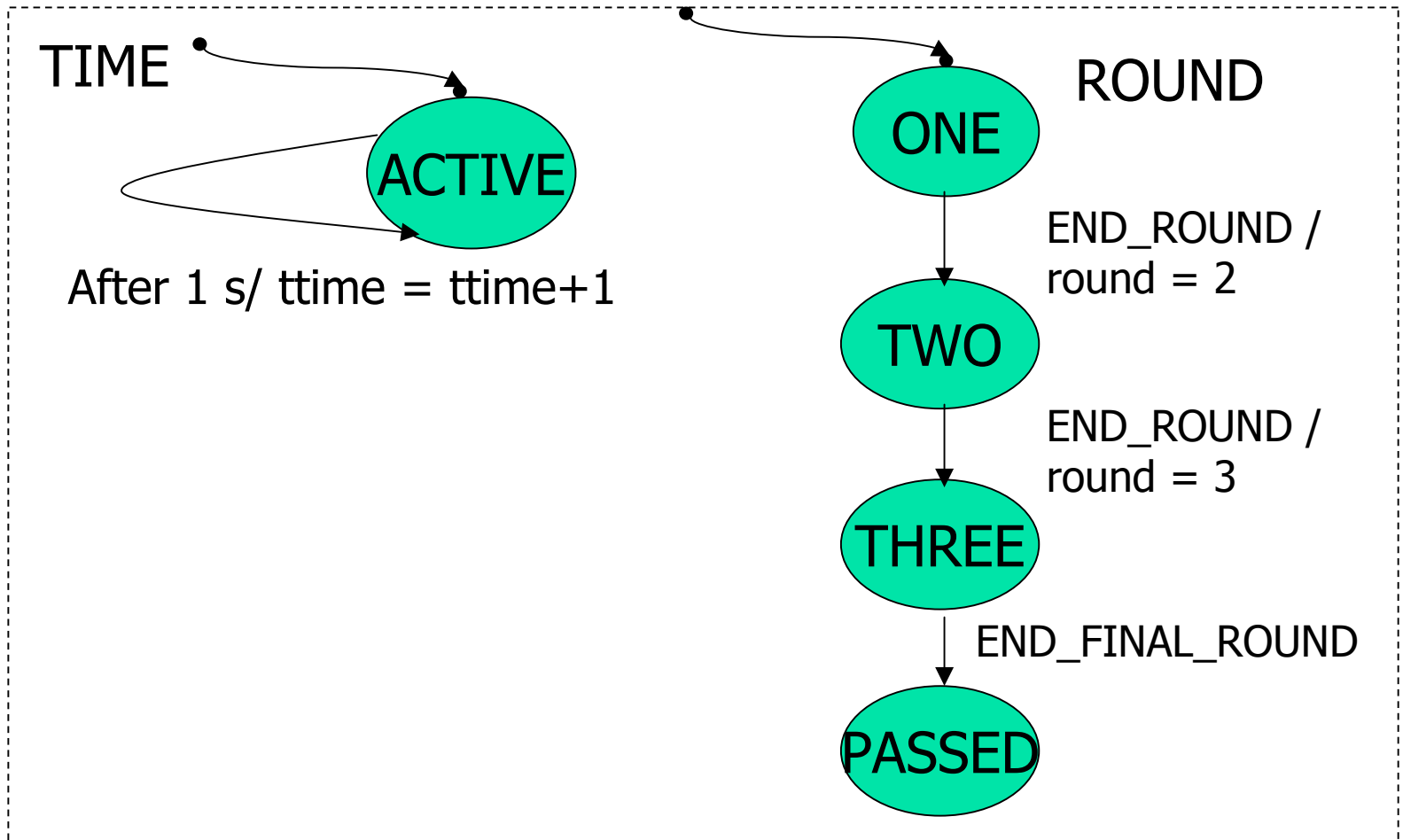
Process Model: Re-modeled (8)



Process Model: Re-modeled (9)



Process Model: Re-modeled (10)



The Unconstrained Process Model (UPM)

- These tasks correspond to the active states in the statechart model.
- The basic plan forecasts that after initial development of code and tests, test execution will uncover errors calling for the rework of both code and tests at half their initial effort level.
- The second round of testing will uncover more errors, but only in the code, requiring one-quarter the initial effort to correct.
- The tests will then be passed on the third round. It has been assumed that each of these tasks is a one-person task that cannot be distributed among multiple workers.

	<u>Hours</u>
Round 1	
Developing Code	12
Developing Tests	8
Running Tests	1
Analyzing Problems	3
Round 2	
Developing Code	6
Developing Tests	4
Running Tests	1
Analyzing Problems	2
Round 3	
Developing Code	3
Running Tests	1
-- Passed --	

Table 1: Resource Requirements for Examples



Scheduling Considerations

- In the first round, development of code will take 12 hours, development of tests will take 8 hours, running the tests will take 1 hour, and analyzing problems in testing will take another 3 hours.
- The second round takes half the time of the first round and the third takes one-third the time.
- Exception: Running the tests will always take 1 hour



Process Simulation

- Statechart Virtual Machine (SVM) has been used to simulate the software process.
- SVM has been developed by Thomas H. Feng (SOCS, McGill University) and was released in Feb 2003.

[SIMULATOR CODE](#)

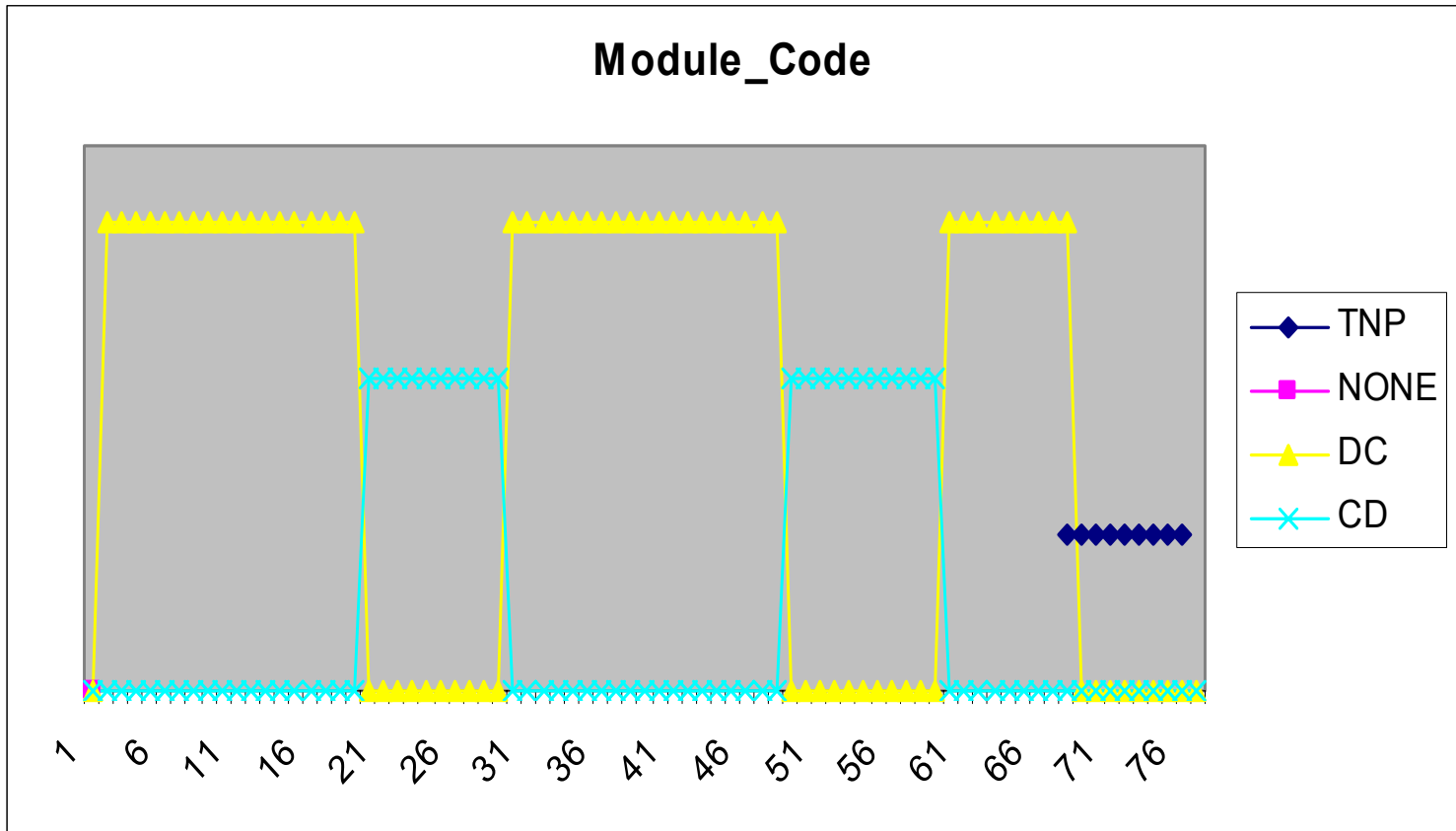
[SIMULATE](#)



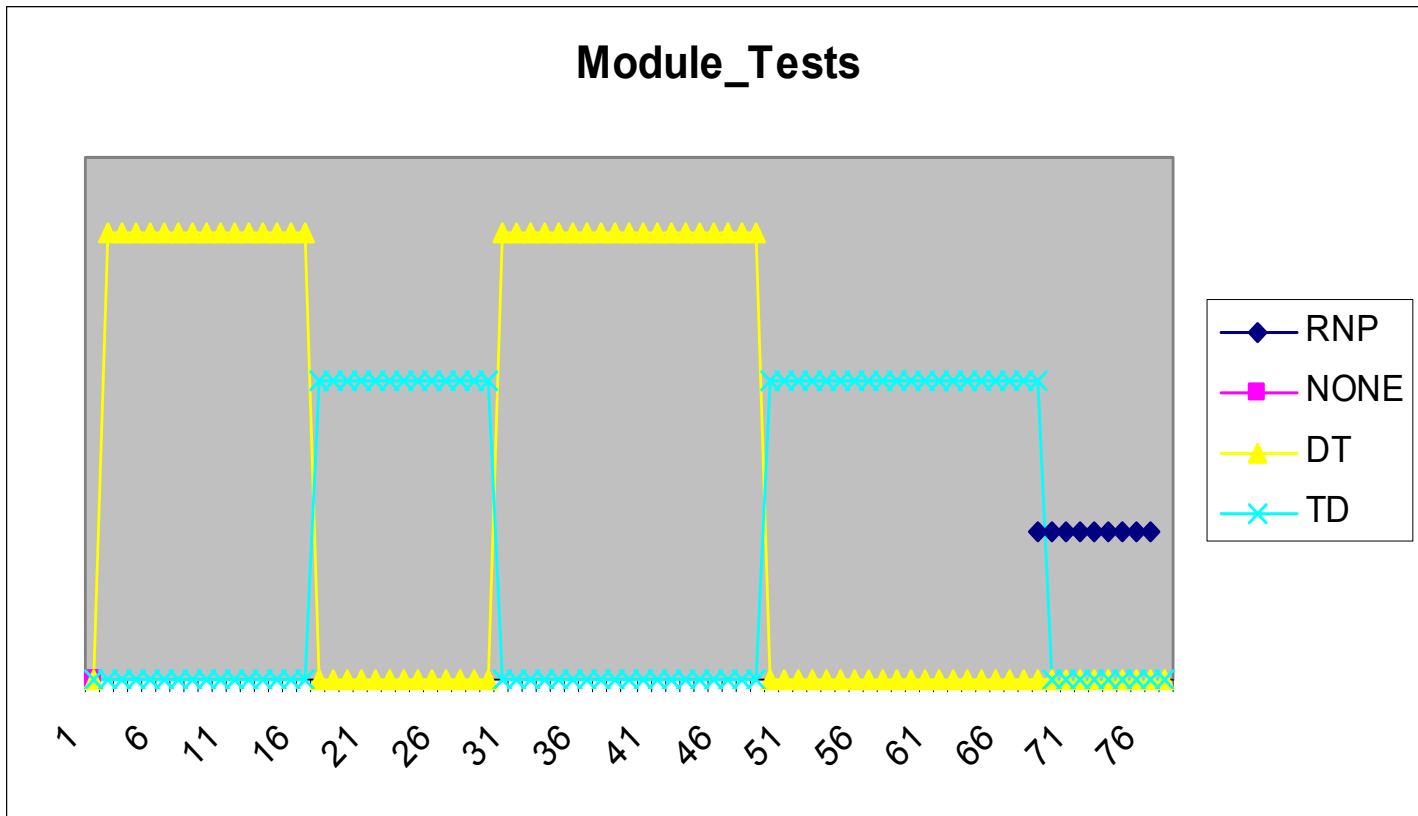
Simulation Results

- Currently, no interface has been defined to generate the graphs automatically in SVM.
- The output of the simulation is printed in text files which have been imported into Excel.

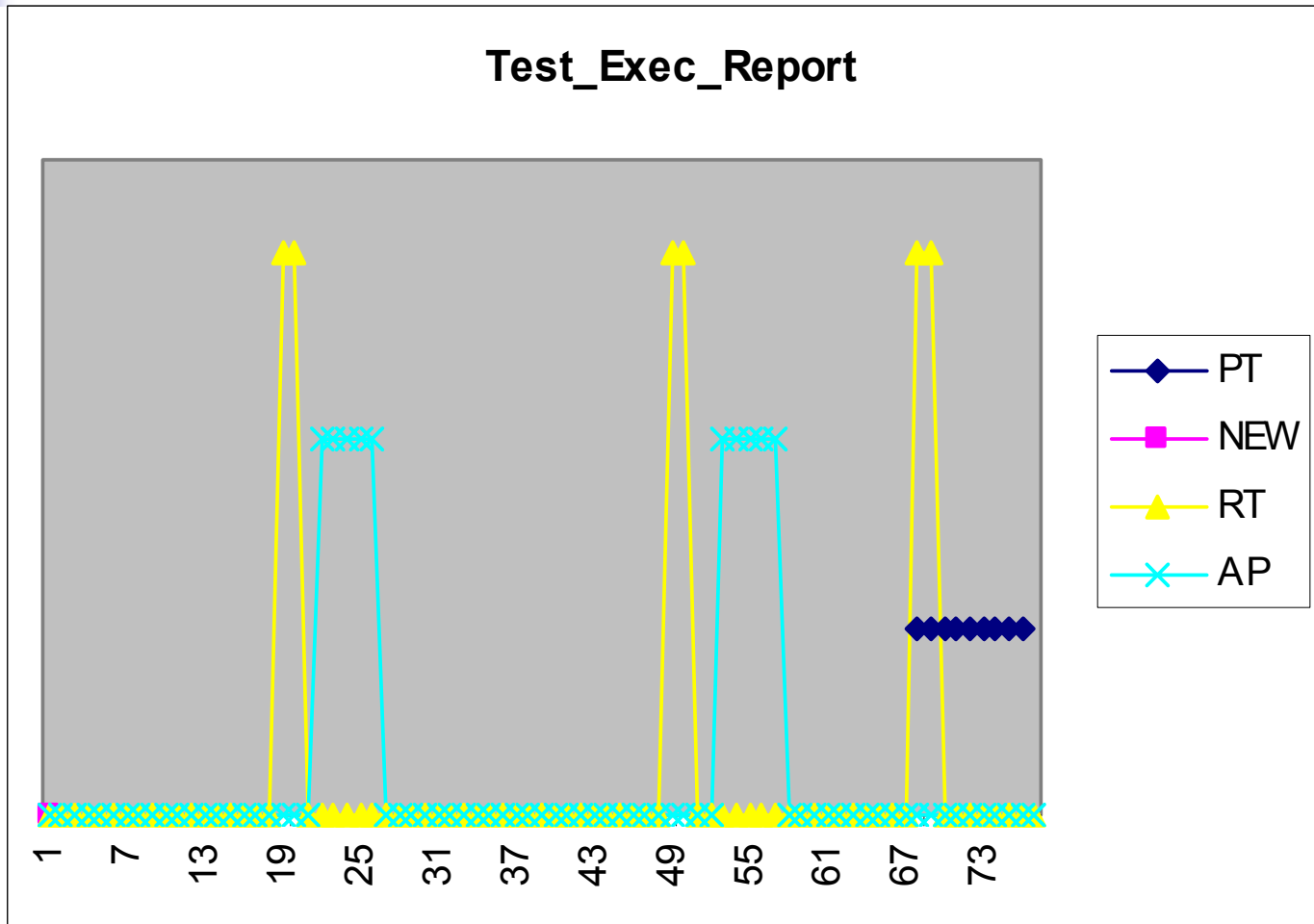
Simulation Results (1)



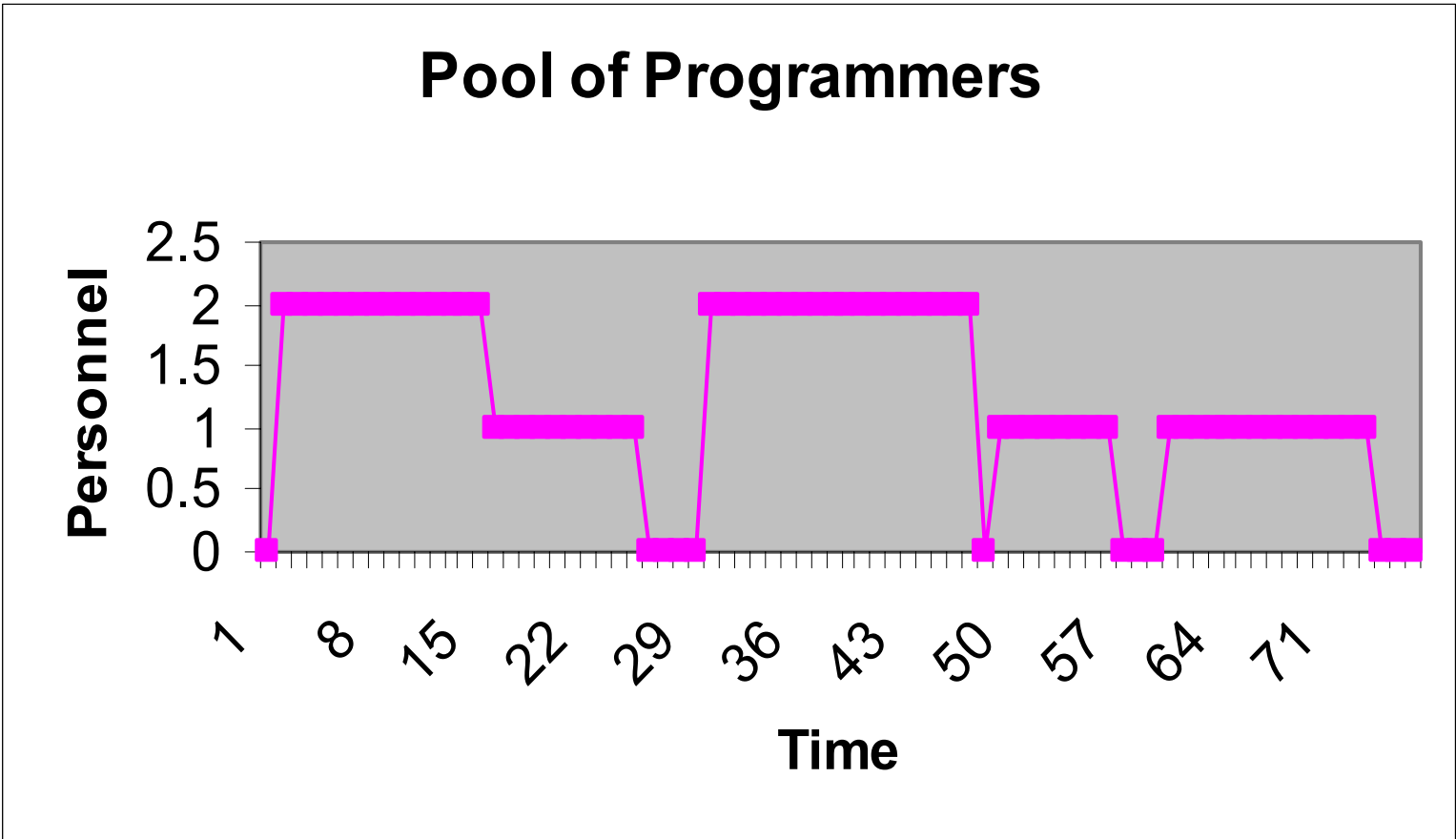
Simulation Results (2)



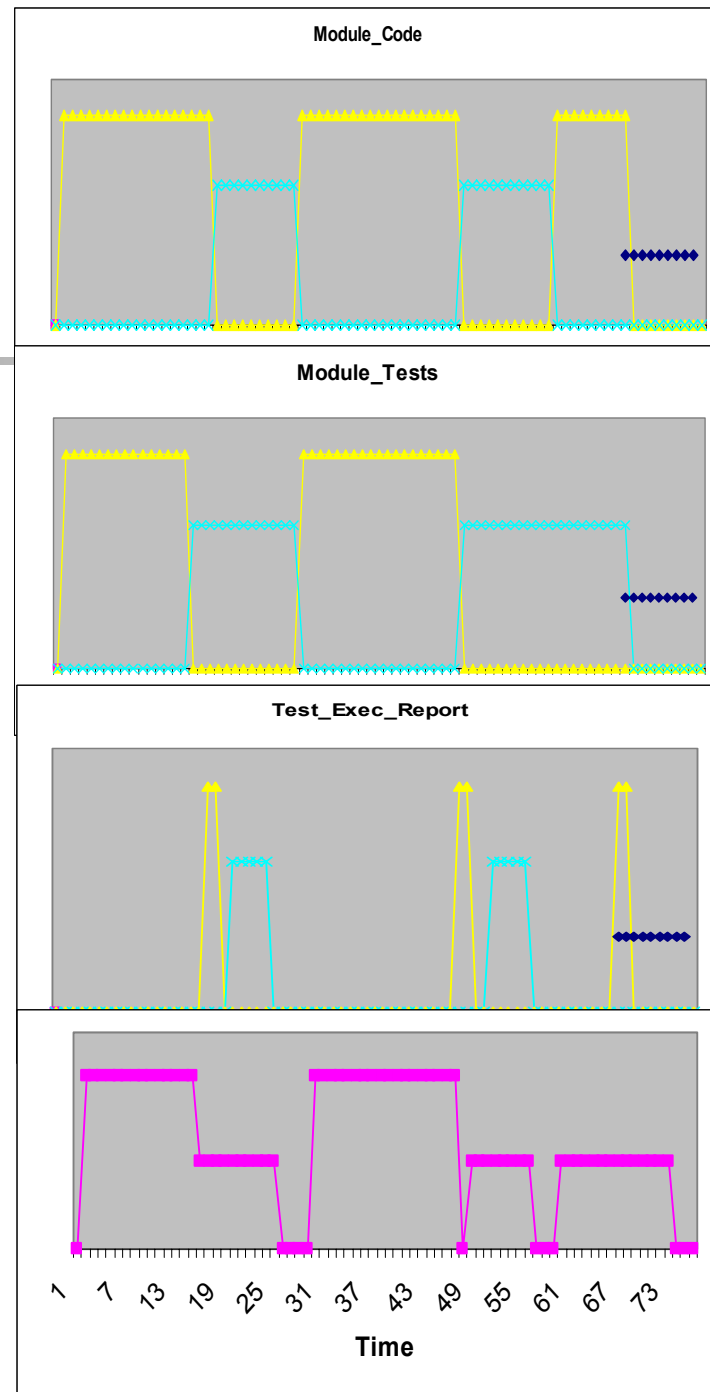
Simulation Results (3)



Simulation Results (4)



Comparison of the Results



Example UPM

Module_Code

None
 Developing_Code
 Code_Developed
 Tested_n_Passed



Module_Tests

None
 Developing_Tests
 Tests_Developed
 Run_n_Passed



Test_Exec_Report

New
 Running_Tests
 Analyzing_Probs
 Passed_Testing



Personnel
 Required



Example CPM

Module_Code

Start

Developing_Code

Code_Developed

Tested_n_Passed

Module_Tests

Start

Developing_Tests

Tests_Developed

Ran_n_Passed

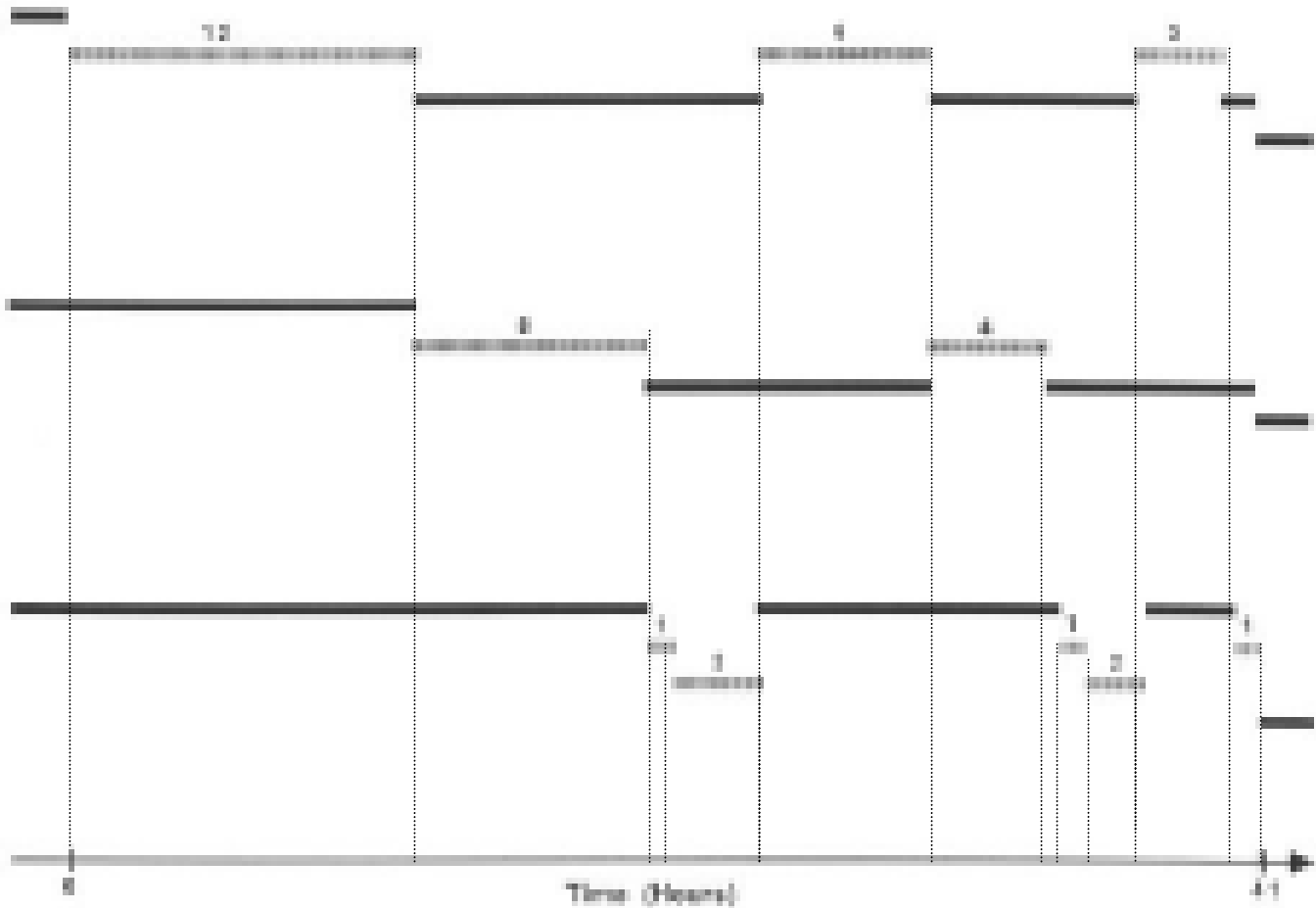
Test_Exec_Report

Start

Running_Tests

Analyzing_Probs

Passed_Testing





References

1. Watts S. Humphrey , Marc I. Kellner, Software process modeling: principles of entity process models, Proceedings of the 11th international conference on Software engineering, p.331-342, May 1989, Pittsburgh, Pennsylvania, United States
<http://portal.acm.org/citation.cfm?doid=74587.74631>