# Project 1 TEST PLAN (IEEE 829 FORMAT)

1. Test Plan Identifier
2. References
3. Introduction
4. Test Items
5. Software Risk Issues
6. Features to be Tested
7. Features not to be Tested
8. Approach
9. Item Pass/Fail Criteria
10. Suspension Criteria and Resumption Requirements
11. Test Deliverables
12. Remaining Test Tasks
13. Environmental Needs
14. Staffing and Training Needs
15. Responsibilities
16. Schedule
17. Planning Risks and Contingencies
18. Approvals
19. Glossary

# **PREDICTION OF HEART ATTACK**

## HEART ATTACK TEST PLAN 001

## REFERENCES

* + 1. SRS Document
    2. Project Plan
    3. SOW

## INTRODUCTION

This project is called the Prediction of Heart Attack; it is used to predict the occurrence of heart attack based on existing patient datasets, which consists of factors such as cholesterol, age, gender, blood pressure etc. and the “target” value for that particular patient providing the status of heart attack (occurred or not occurred). This product contains four modules.

1. ML algorithm
2. DL algorithms
3. Pandas (SQL), Dataset handler.
4. Database Driver.

The user provides real-time data of the given health factors and the model predicts the occurrence of a heart attack.

## TEST ITEMS(FUNCTIONS)

1. Testing of ML (Machine Learning) Algorithm
2. Testing the accuracy of the Machine learning algorithm.
3. Testing of DL (Deep Learning) Algorithm.
4. Testing the accuracy of the Deep learning algorithm
5. GUI: Warning of predicted Heart Attack.

## SOFTWARE RISK ISSUES

* + 1. Overfitting of model (ML/DL).
    2. Underfitting of model (ML/DL).
    3. Data quality of provided dataset.
    4. Feature selection.

## FEATURES TO BE TESTED

Features mentioned in section 4 will be tested here.

## FEATURES NOT TO BE TESTED

Not applicable to this project.

## APPROACH (STRATEGY)

* + 1. Invoke the “Heart Attack Prediction software”.
    2. Check the GUI screen; the GUI screen will be popped up.
    3. Using the GUI, screen, test ML model prediction, Test DL model prediction, Heart attack warning, and verify output with the provided dataset to calculate accuracy.

## ITEM PASS/FAIL CRITERIA

* For every test case, we give an input and expect the output.
* The input contains 13 features; factors related to heat conditions.
* If the output or label is similar to the expected output then the test case is passed.
* Ex: If given text data is; X\_test = [[56, 1, 1, 130, 221, 0, 0, 163, 0, 0, 2, 0, 3]] and if the prediction matches y\_test=[1], then the test case has passed.

## SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS

If 25% of test case fail to pass or meet requirements, then the testing is suspended.

## TEST DELIVERABLES

* Test plan
* Test script
* Test script automation
* Test cases

## REMAINING TEST TASKS

Not applicable for this project.

## ENVIRONMENTAL NEEDS

Two engineers for one month

## STAFFING AND TRAINING NEEDS

Training for two engineers.

## RESPONSIBILITIES

Who is in charge?

This issue includes all areas of the plan. Here are some examples:

* 1. Setting risks.
  2. Selecting features to be tested and not tested.
  3. Setting overall strategy for this level of plan.
  4. Ensuring all required elements are in place for testing.
  5. Providing for resolution of scheduling conflicts, especially, if testing is done on the production system.
  6. Who provides the required training?
  7. Who makes the critical go/no go decisions for items not covered in the test plans?

## SCHEDULE

**1. Accurate Estimations are Crucial**

* Project plans should be based on realistic and validated estimates.
* Inaccurate development estimates can lead to project delays.
* Testing is an integral part of the project plan.

**2. Risk of Cutting Testing**

* Testing often gets sacrificed when project deadlines loom.
* Rushed releases can result in subpar quality.
* Handling schedule slippage should be addressed proactively.

**3. User Awareness and Tolerance**

* Users should be informed that development delays affect testing and system delivery.
* User tolerance may increase if they understand the benefits of thorough testing.
* Discussing potential effects in advance can lead to user agreement on certain defects if schedules slip.

**4. Milestones and Tracking**

* Identify all relevant milestones and their relation to the development process.
* Tracking slippage caused by testing becomes more manageable.
* Link test dates directly to related development activity dates to prevent blaming the testing team for delays.
* Use dependent or relative dating to ensure testing starts in tandem with delivery, even if delivery is delayed.

## PLANNING RISKS AND CONTINGENCIES

**Overall Risks to the Project with Emphasis on Testing:**

1. **Lack of Personnel Resources at Testing Phase:**
   * Risk: Insufficient testing team members when testing is scheduled to start.
   * Mitigation: Resource allocation adjustments, including adding more team members if possible.
   * Consideration: Impact on team morale due to potential overtime work.
2. **Insufficient Availability of Required Resources:**
   * Risk: Unavailability of necessary hardware, software, data, or tools for testing.
   * Mitigation: Prioritize and secure required resources well in advance to avoid delays.
3. **Late Delivery of Software, Hardware, or Tools:**
   * Risk: Delays in receiving essential software, hardware, or tools for testing.
   * Mitigation: Proactive communication and coordination with suppliers to ensure on-time delivery.
4. **Delays in Training:**
   * Risk: Delays in training for the testing team on the application and tools.
   * Mitigation: Schedule training sessions early, consider backup training resources, and allocate additional time if needed.
5. **Changes to Original Requirements or Designs:**
   * Risk: Changes to project requirements or designs impacting testing.
   * Mitigation: Define a change management process and its impact on schedules:
     + If requirements change post-defined date, adjust test and development schedules.
     + Be prepared to reduce the number of tests or increase acceptable defects (careful consideration for product quality).
     + Add resources or consider overtime if required (monitor impact on team morale).
     + Be cautious about altering the project scope, and avoid unnecessary optimization of resources.
     + Extreme measures like project abandonment should only be considered as a last resort.
6. **Management Reluctance:**
   * Observation: Management is often reluctant to accept extreme scenarios.
   * Reminder: Inaction typically results in cutting or omitting testing, which is an undesirable outcome.

## APPROVALS

**Approval Process for Advancing to the Next Project Level:**

1. **Master Test Plan Level:**
   * Approval: Typically involves all relevant parties.
   * Consideration: This level requires broad consensus as it impacts the overall project.

**Important Factors to Consider in Determining the Approval Process:**

1. **Audience Variances:** -
   * Different plans have different audiences. - Consider the specific audience when defining the approval process.
2. **Knowledge Differences at Various Levels:** -
   * Varying levels of knowledge exist at different plan levels. - Recognize these differences when seeking approvals.
3. **Technical Expertise vs. Business Understanding:** -
   * Programmers possess technical expertise but may lack comprehensive understanding of the project's business processes. - Users may have varying business acumen and limited technical skills. - Be cautious of individuals who overstate their technical skills or their grasp of business processes, as this can lead to misunderstandings and issues.
   * The key takeaway is that the approval process should be tailored to the specific plan level and the expertise of the individuals involved. A comprehensive understanding of both technical and business aspects is essential to ensure the successful progression of the project.Top of FormBottom of Form

## GLOSSARY

Used to define terms and acronyms used in the document, and testing in general, to eliminate confusion and promote consistent communication.