Beta vs. Triangular Distribution

A Case Study for Time Estimations

ABSTRACT

The three-point estimation method in project management offers a strategy for gauging the duration or effort needed for a task or project. Recognizing the limitations of a single estimate, particularly in intricate or uncertain contexts, this approach integrates three distinct estimates: optimistic, pessimistic, and most likely.

-Donell Adams-Welch

Contents

Overview 1

Procedural Guide to Time Estimates and Resource Allocation 2

Beta Distribution Versus Triangular Distribution 4

Beta Distribution 4

Triangular Distribution 4

Case Study: Three-Point Estimations Using the Beta Distribution Formula 5

Gather Inputs 5

Quality Tasks 8

Determine Estimates Using the Beta Distribution 9

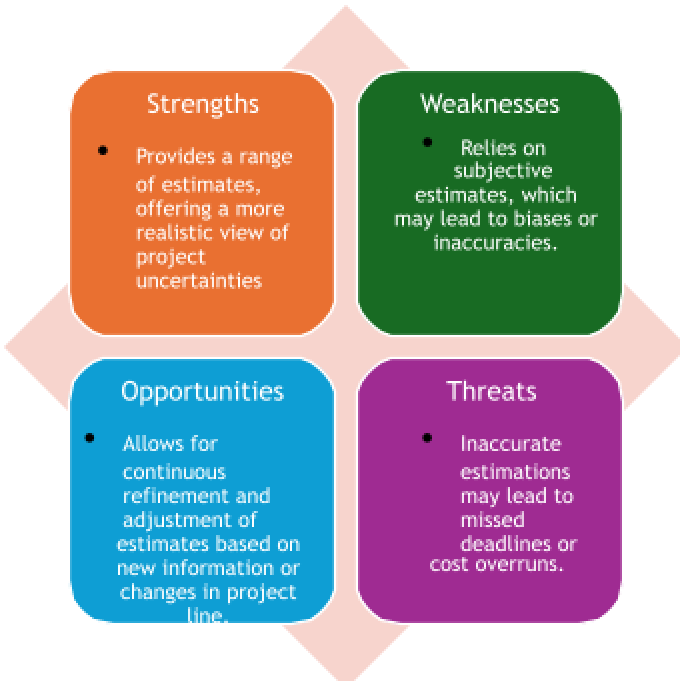
Overview

The three-point estimation method is a technique used in project management to estimate the duration or effort required for a task or project. It is based on the idea that using a single estimate might not accurately reflect the true range of possible outcomes, especially in complex or uncertain situations. The three-point estimation method incorporates three estimates for each task: optimistic, pessimistic, and most likely.

- Optimistic Estimate (O): This estimate represents the most optimistic scenario where everything goes smoothly without any obstacles or delays. It assumes the best-case scenario.
- Pessimistic Estimate (P): This estimate represents the most pessimistic scenario where everything that could go wrong does go wrong. It considers all possible obstacles, delays, and complications.
- Most Likely Estimate (M): This estimate represents the most realistic assessment based on the available information and past experiences. It considers normal circumstances and typical challenges that may arise.

Once these three estimates (optimistic, pessimistic, and most likely) are determined, they are used to calculate a weighted average or expected value using one of several methods, such as the triangular distribution or the beta distribution. The formulas for calculating the expected duration or effort vary depending on the distribution method chosen.

The purpose of using the three-point estimation method is to provide a more accurate and reliable estimate by considering a range of possibilities rather than relying solely on a single point estimate. It helps project managers account for uncertainty and variability in their planning, leading to more realistic schedules, budgets, and resource allocations. Additionally, it allows for better risk management and decision-making throughout the project lifecycle.



Procedural Guide to Time Estimates and Resource Allocation

1. Gather Inputs:

- Review project requirements, line, and deliverables.
- Collect past performance data, including historical effort and completion times.
- Identify potential risks, constraints, and dependencies affecting all tasks.
- Assess team capabilities, including skill levels and availability of resources.

2. Identify Quality Tasks:

- Break down QA activities into individual tasks or subtasks.
- List all tasks involved in the process, including test planning, test case development, test execution, defect tracking, regression testing, and reporting.

3. Determine Three Estimates:

- For each task, determine three estimates: optimistic (O), pessimistic (P), and most likely (M).
    - Optimistic Estimate (O): Best-case scenario assuming smooth execution.
    - Pessimistic Estimate (P): Worst-case scenario considering potential obstacles.
    - Most Likely Estimate (M): Realistic assessment based on available information.

4. Calculate Expected Time:

- Use the three estimates to calculate the expected time for each task.
    - For the beta distribution: Expected Time (TE) = (O + 4M + P) / 6
    - For the triangular distribution: Expected Time (TE) = (O + M + P) / 3

5. Allocate Resources:

- Based on the expected time estimates for each task, allocate resources such as engineers, testing environments, tools, and equipment.
- Consider the skill level and availability of resources when making allocations.

6. Factor in Contingencies:

- Account for potential risks and uncertainties by adding contingency buffers to the time estimates and resource allocations.
- Allocate additional time and resources to handle unexpected events or delays.

7. Monitor and Adjust:

- Continuously monitor the progress of activities against the estimated timeframes.
- Analyze deviations and identify the reasons behind them.
- Adjust future estimates and resource allocations based on lessons learned and past performance data.

8. Documentation and Reporting:

- Document all time estimates, resource allocations, and actual effort expended for all tasks.
- Provide regular reports to stakeholders on the progress of activities and any adjustments made to plans.

9. Review and Improve:

- Conduct regular reviews of the three-point estimation process to identify areas for improvement.
- Solicit feedback from team members and stakeholders to enhance accuracy and efficiency.

10. Conclusion:

- By following this process, managers can effectively use the three-point estimation technique to determine time estimates and resource allocation for all activities, leading to improved project planning and execution.

Beta Distribution Versus Triangular Distribution

Beta Distribution:

- The beta distribution is a probability distribution commonly used for estimating uncertain variables where the minimum, maximum, and most likely values are known or can be estimated.

- It uses a continuous probability distribution function with shape parameters alpha (α) and beta (β), providing more flexibility and precision in modeling uncertainty by adjusting these parameters to fit the data.

- The beta distribution is suitable for situations where the uncertainty follows a specific pattern or has known characteristics, allowing for more accurate estimation.

- However, it requires a larger sample size and more complex calculations, especially when shape parameters need to be estimated from data.

- It is often used in Monte Carlo simulations and statistical modeling to simulate various scenarios and outcomes, such as project scheduling, inventory management, financial modeling, and risk analysis.

Triangular Distribution:

- The triangular distribution is a probability distribution commonly used in risk assessment and project management for representing uncertainty in variables where the minimum, maximum, and most likely values are known or can be estimated.

- It forms a triangle-shaped distribution, with the most likely value at the peak and the minimum and maximum values forming the base.

- The triangular distribution provides a simpler approach and is easier to understand and apply, especially for simple or less precise data.

- It is suitable for situations where there is limited information or data available about the uncertainty, and assumptions about its distribution are less certain.

- The triangular distribution requires less computational effort and can be quickly calculated with basic statistical techniques, making it more accessible for quick estimations.

- It is commonly used in project scheduling, risk analysis, decision-making processes, project planning, resource allocation, cost estimation, and scenario analysis in project management.

In summary, while both distributions are used for estimating uncertainty, the beta distribution offers more flexibility and precision but requires more complex calculations, whereas the triangular distribution provides a simpler approach suitable for situations with limited information or data.

Case Study: Three-Point Estimations Using the Beta Distribution Formula

Gather Inputs

Requirements Specification: MyProject Enhancements

1. Introduction

This document outlines the requirements for enhancing the MyApp Template functionality within MyApp tool. The goal is to improve usability, efficiency, and accuracy of the estimation process.

2. Background

Estimators currently rely on manual copy-pasting between MyApp Templates and the actual line items. This specification proposes changes to enable direct utilization of templates within the package creation process. Additionally, future functionalities like automated pricing and estimates are envisioned.

3. Terminology

- **MyApp Template:** Predefined set-up used as a basis for creating MyApps.
- **Line Item:** Individual item added to a work order based on a MyApp Template.
- **Work Order:** Represents a maintenance, turn, or initial rehab project.
- **MAPT Code:** Code used for potential future integration with MAPTT pricing.

4. Requirements

4.1 MyApp Template Updates

- **Trade:**
  - Add a new field "Trade" to the MyApp Template.
  - This field is a pull-down menu with a required selection.
  - It will be used to restrict vendor selection based on their associated trades.
- **Locks (Editability):**
  - Introduce three toggles ("(i)Editable in Line") for Description, Quantity, and Price fields.
  - By default, these toggles are off, preventing the estimator from editing these values while adding a MyApp.
- **MAPTT Codes:**
  - Add a field for entering multiple MAPTT codes (multi-select).
  - Users can add, remove, or delete MAPTT codes associated with the template.
- **Estimated Days to Complete:**
  - Add a new required field "Estimated Days to Complete".
  - This field accepts integer values between 0 and 90, representing the estimated days needed to complete the task.
- **Dependencies:**
  - Add a "Dependencies" field using chip selection format.
  - This allows selecting existing MyApp Templates as dependencies for the current template.
  - Dependencies define tasks needing completion before starting this specific task.

- **Reorganize Fields:**
  - Move the "Name" field above the "Description" field within the MyApp Template modal.

4.2 Select Line Estimate Template - New "Add" Process

- **Capture Link between Templates and MyApps:**
  - When clicking "Add" on the vNext "Line" screen, display a screen based on the existing MyApp Template search.
- **Read-Only Fields:**
  - Market (populated based on the property's market).
  - Specification Type (populated based on the relevant spec type).
  - User Inputs (required, populated from the selected template, and flow back to the MyApp).
  - Location (required, populated from the template, and carried over after clicking "Add").
- **Search:**
  - Implement a global search functionality.
- **Selecting a Template:**
  - Clicking "Add" on a chosen template populates the current MyApp view with its contents.

4.3 MyApp Screen Updates

- **Line Line Item Estimate Section:**
  - Pre-populate "Area" and "Location" fields based on selections made during template search.
  - "Name", "Category", and "Trade" fields are read-only and populated from the selected template.
- **Labor and Materials:**
  - "Description" is editable only if the corresponding toggle is enabled.
  - "Quantity" and "Price" are editable only if their respective toggles are enabled.
  - "Measure" is always read-only.

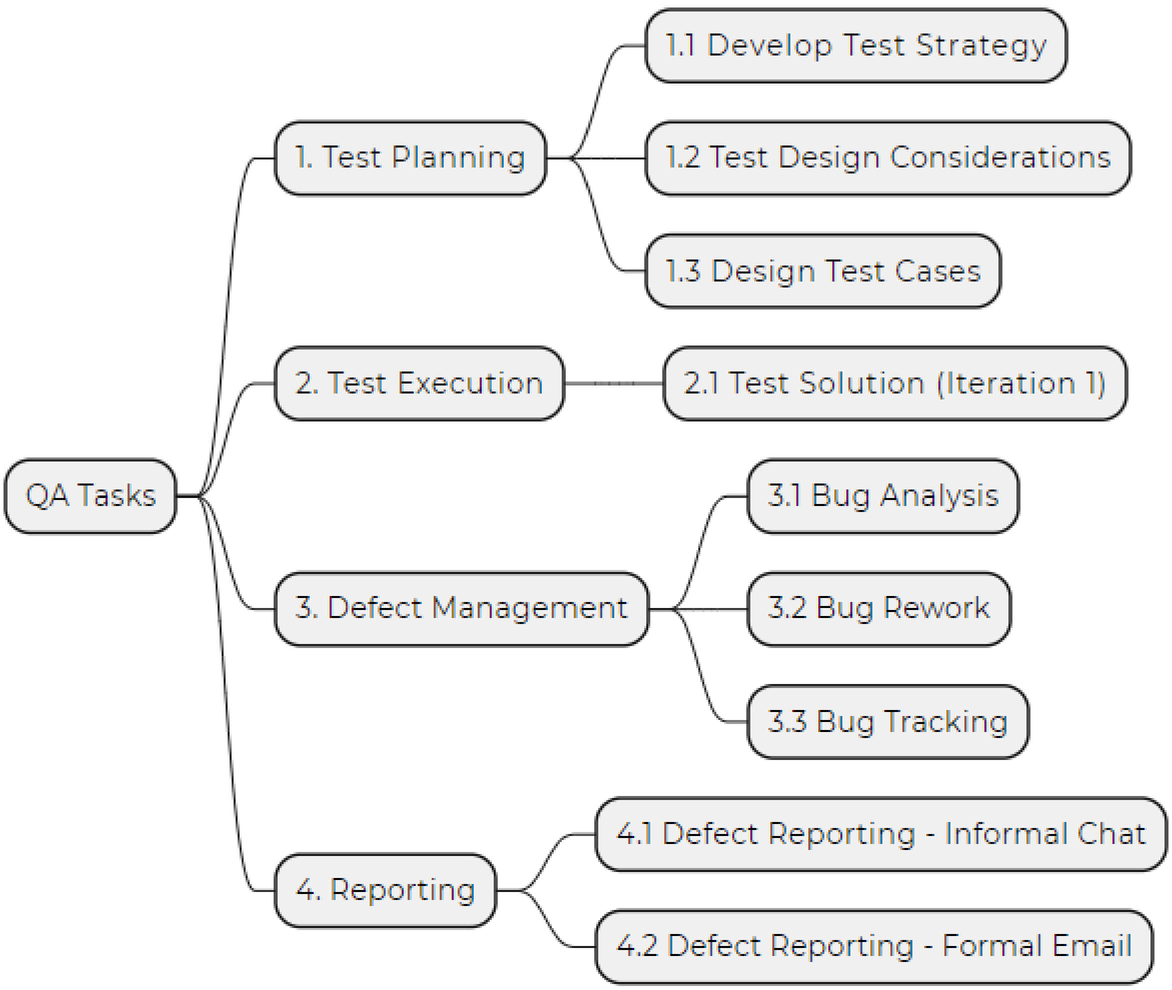
- "Total" is always read-only.

- **Materials Section:**
  - Involves adding material selection functionality to the template. Quality Tasks

1. **Test Planning:**
   - **Think before you test!** This stage involves creating a strategy for testing the features of your real estate app. It's like making a checklist of things to examine.
     - **Develop Test Strategy:** Here, the Quality team decides **what** functionalities to test, **how** to test them, and **when** to test them based on the app's development stages. It's like figuring out which parts of the app to investigate and in what order.
     - **Test Design Considerations:** The team thinks about factors that might affect testing, like different user types (property managers, tenants) or potential issues with various devices (phones, tablets). Imagine checking the app for user- friendliness on different screen sizes or considering how different user roles might interact with it.
     - **Design Test Cases:** Specific scenarios are created to test different functionalities of the app. This is like creating a detailed list of things to try out, like submitting a maintenance request, viewing lease agreements, or searching for available units.
2. **Test Execution:**
   - **Time to put the plan into action!** This stage involves running the tests and seeing how the app performs.
     - **Test Solution (Iteration 1):** The Quality team starts testing the app, one test case at a time. It's like trying out each feature of the app to see if it works as planned. This might be the first of several rounds of testing, depending on any updates or new features added.
3. **Defect Management:**
   - **Uh oh, something isn't right!** This stage involves finding and fixing bugs (errors) in the app.
     - **Bug Analysis:** The Quality team investigates any problems they encounter during testing. They figure out what went wrong and how to replicate the issue. It's like finding a glitch in the app and understanding why it's happening.
     - **Bug Rework:** Once the bug is understood, developers fix the app to address the problem. This is like patching up the glitch in the app.

- **Bug Tracking:** The Quality team keeps track of all the bugs they find and their resolution status. It's like creating a log of all the app issues discovered and ensuring they're fixed.

4. **Reporting:**
   - **Sharing the results!** The Quality team documents their findings and communicates any issues to the development team. They might also create reports summarizing the testing process and its results. This is like presenting a report to the app developers about any bugs found and suggesting improvements.



Determine Estimates Using the Beta Distribution

To use the beta distribution formula to estimate the time required for the tasks outlined in the Requirements Specification: MyApp Template Enhancements, we need to identify the tasks and determine their optimistic

(O) , pessimistic (P), and most likely (M) time estimates. Let's consider 1 resource for these tasks.

Given values:

- Optimistic (O) = 15 minutes
- Most Likely (M) = 45 minutes
- Pessimistic (P) = 1.5 hours = 90 minutes

Now, let's calculate the expected time (TE) for each task using the beta distribution formula:

TE = (O + 4M + P) / 6

1. MyApp Template Updates:

- Trade: O = 15 min, M = 45 min, P = 90 min
- Locks (Editability): O = 15 min, M = 45 min, P = 90 min
- MAPTT Codes: O = 15 min, M = 45 min, P = 90 min
- Estimated Days to Complete: O = 15 min, M = 45 min, P = 90 min
- Dependencies: O = 15 min, M = 45 min, P = 90 min
- Reorganize Fields: O = 15 min, M = 45 min, P = 90 min

2.Select Line Estimate Template - New "Add" Process:

- Capture Link between Templates and MyApps: O = 15 min, M = 45 min, P = 90 min
- Read-Only Fields: O = 15 min, M = 45 min, P = 90 min
- Search: O = 15 min, M = 45 min, P = 90 min

3.MyApp Screen Updates:

- Line Line Item Estimate Section: O = 15 min, M = 45 min, P = 90 min
- Labor and Materials: O = 15 min, M = 45 min, P = 90 min

Calculate Time

Now, let's calculate the expected time (TE) for each task using the beta distribution formula:

TE = (O + 4M + P) / 6

Expected Time (TE) for Each Task:

1. Trade: TE = (15 + 4(45) + 90) / 6 = (15 + 180 + 90) / 6 = 285 / 6 ≈ 47.5 minutes
2. Locks (Editability): TE = (15 + 4(45) + 90) / 6 = (15 + 180 + 90) / 6 = 285 / 6 ≈ 47.5 minutes
3. MAPTT Codes: TE = (15 + 4(45) + 90) / 6 = (15 + 180 + 90) / 6 = 285 / 6 ≈ 47.5 minutes
4. Estimated Days to Complete: TE = (15 + 4(45) + 90) / 6 = (15 + 180 + 90) / 6 = 285 / 6 ≈ 47.5 minutes
5. Dependencies: TE = (15 + 4(45) + 90) / 6 = (15 + 180 + 90) / 6 = 285 / 6 ≈ 47.5 minutes
6. Reorganize Fields: TE = (15 + 4(45) + 90) / 6 = (15 + 180 + 90) / 6 = 285 / 6 ≈ 47.5 minutes
7. Capture Link between Templates and MyApps: TE = (15 + 4(45) + 90) / 6 = (15 + 180 + 90) / 6 = 285 / 6 ≈ 47.5 minutes
8. Read-Only Fields: TE = (15 + 4(45) + 90) / 6 = (15 + 180 + 90) / 6 = 285 / 6 ≈ 47.5 minutes
9. Search: TE = (15 + 4(45) + 90) / 6 = (15 + 180 + 90) / 6 = 285 / 6 ≈ 47.5 minutes
10. Line Line Item Estimate Section: TE = (15 + 4(45) + 90) / 6 = (15 + 180 + 90) / 6 = 285 / 6 ≈ 47.5 minutes
11. Labor and Materials: TE = (15 + 4(45) + 90) / 6 = (15 + 180 + 90) / 6 = 285 / 6 ≈ 47.5 minutes

These are the estimated times for each task using the beta distribution formula. Each task is estimated to take approximately 47.5 minutes.

Since each task has the same expected time (TE) of approximately 47.5 minutes, we can calculate the sum total time estimate by multiplying the expected time for one task by the total number of tasks.

Given that we have 11 tasks in total:

Total Time Estimate = Expected Time per Task * Number of Tasks

≈ 47.5 minutes * 11

≈ 522.5 minutes

The sum total time estimate for all the tasks in the project is approximately 522.5 minutes.

To convert 522.5 minutes to hours, you can divide by 60 since there are 60 minutes in an hour:

522.5 minutes / 60 ≈ 8.708 hours.

So, 522.5 minutes is approximately equal to 8.708 hours.

Resource Allocation

Given:

- Total time estimate for all tasks: 522.5 minutes or 8.708 hours.
- QA resource's allocation: 80% per day

Now, we calculate the available time per day for the resource based on an 80% allocation: 80% of a day = 0.80 * 8 hours = 6.4 hours

Finally, we divide the total time required by the available time per day to find the number of days needed:

Number of days = Total time required / Available time per day ≈ 8.708 hours / 6.4 hours ≈ 1.3625 days or 2 days.

Factor in Contingencies

We can incorporate a buffer or contingency factor to accommodate the time required for bug fixing or rework.

Here's how we can modify the formula:

1. Calculate the total time estimate for all tasks including bug rework.
2. Determine the available time per day for the resource considering their allocated percentage.
3. Add a buffer or contingency factor to the total time estimate to account for bug rework.
4. Divide the adjusted total time estimate by the available time per day to find the number of days needed.

Let's add a buffer of 20% to the total time estimate to account for bug rework. Here's how we can calculate the number of days needed:

1. Adjusted total time estimate = Total time estimate * (1 + Bug rework buffer)
2. Number of days = Adjusted total time estimate / Available time per

day Using the values from the previous calculation:

Adjusted total time estimate = 8.708 hours * (1 + 0.20) = 8.708 hours * 1.20 ≈ 10.4496 hours

Now, we divide this adjusted total time estimate by the available time per day (6.4 hours) to find the number of days needed:

Number of days = 10.4496 hours / 6.4 hours ≈ 1.6335 days

Since we can't have a fraction of a day in this context, we would round up to the nearest whole number. Therefore, the QA resource would need approximately 2 days to complete all tasks with an additional buffer for bug rework.