Engineering Method - FIBA players application

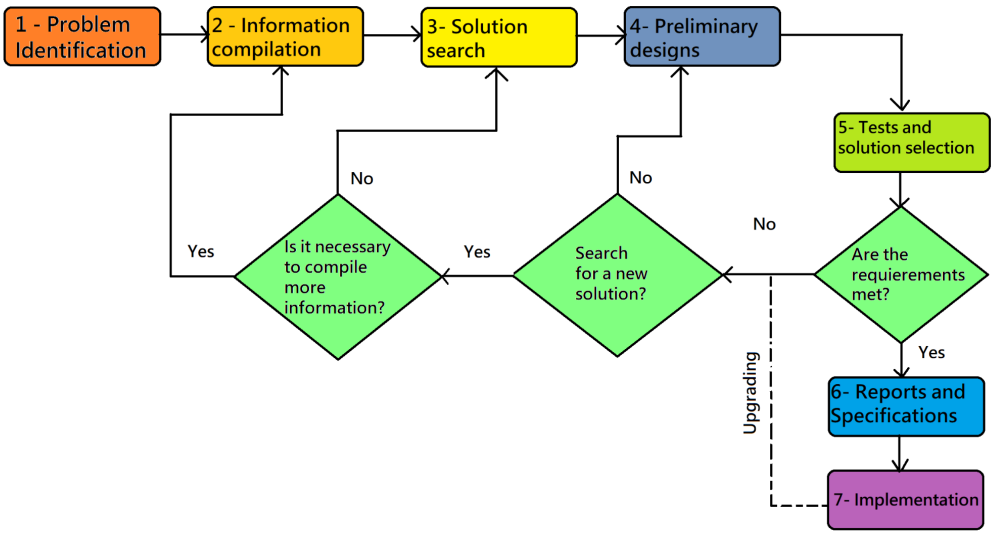
Problem Context[1]:

The International Basketball Federation, also known as FIBA, is the world governing body for basketball, the one that defines the rules of this sport at the international level and the body not only in charge of organizing and coordinating the most important orbital competitions but also of bringing together to all practitioners of this sport at a professional level. FIBA requires a first version of a software solution that handles a large volume of data: the most relevant data of each of the basketball professionals in the planet, so that different queries can be made that allow analysis of these data; the solution must also allow users to know: patterns about the development of the sport, the criteria that take more force or, in general, where the sport is currently heading.

Development of the Solution:

To resolve the above situation, the Engineering Method was chosen to develop the solution following a systematic approach and in line with the problematic situation established.

Based on the description of Paul Wright's “Introduction to Engineering”, the following flowchart was defined, the steps of which we will follow in the development of the solution.



[1] Fictional problem context.

Step 1. Problem Identification

The specific needs of the problematic situation are recognized as well as their symptoms and conditions under which it must be resolved.

Identification of needs and symptoms:

- The solution to the problem must be efficient so that it can make players queries as fast as possible according to desired criteria, with a time complexity of O(log n).

Functional requirements:

The program must be able to:

Name:	**FR1:** Enter player data.
Description:	The system must allow the submission of player data either in mass or through a graphic interface.
Inputs:	Name, age, team, ID, and the 5 following statistical values: true shooting (%), usage (%), assist (%), rebound (%) and defensive (+/-).
Output:	Player data entered.

Name:	**FR2**: Save the players' data.
Description:	The system must allow access to players' data previously submitted.
Inputs:	Nothing.
Output:	Players' data saved.

Name:	**FR3:** Modify player data.

Description:	The system must allow the modification of player data either in mass or through a graphic interface.
Inputs:	Player ID, data to modify, and new value of the data.
Output:	Message about the modification of the player data.

Name:	**FR4:** Delete player data.
Description:	The system must allow the deletion of player data either in mass or through a graphic interface.
Inputs:	ID of the player to be deleted.
Output:	Message about the deletion of the player data.

Name:	**FR5:** Search a player by statistical attributes.
Description:	The system must allow the search of a player through one or two statistical attributes.
Inputs:	Statistical attribute of the player to search.
Output:	Player(s) searched.

Name:	**FR6:** Display search time.
Description:	The system shows the user how long it takes to perform a search (FR5).
Inputs:	Nothing
Output:	Time taken is displayed.

Problem Definition:

FIBA requires the development of a software application that allows users to make queries of basketball players.

Step 2. Information Collection

In order to have complete clarity in the concepts involved, a search is made for the definitions of the terms most closely related to the problem established. It is important to perform this search with recognized and reliable sources to know which elements are part of the problem and which are not.

Definitions:

Sources:

Introduction to Algorithms. Cormen et al. Chapter 3. Growth of Functions.

https://www.basketball-reference.com/about/glossary.html#mp
https://thejumpball.net/2018/01/02/how-the-nba-is-getting-assists-wrong/
https://www.sportslingo.com/sports-glossary/r/rebound/

- Algorithmic Analysis:

  Analyzing an algorithm has come to mean predicting the resources that the algorithm requires. Occasionally, resources such as memory, communication bandwidth, or computer hardware are of primary concern, but most often it is computational time that we want to [1].

- Asymptotic notation:

  The notations we use to describe the asymptotic running time of an algorithm are defined in terms of functions whose domains are the set of natural numbers N '0, 1, 2, ...'. Such notations are convenient for describing the worst-case running-time function T (n), which usually is defined only on integer input sizes. Frequently, asymptotic notation is used to describe the running times of algorithms.

- Θ-notation:

  The Θ-notation asymptotically bounds a function from above and below. For a given function g(n), we denote by $\Theta(g(n))$ the set of functions: $\Theta(g(n))$ = {f(n): there exist positive constants c1, c2 and n0 such that $0 \leq c1g(n) \leq f(n) \leq c2g(n)$ for all $n \geq n0$ }

- O-notation:

  The O-notation provides an asymptotic upper bound function. For a given function g(n), we denote by O(g(n)) the set of functions: O(g(n)) = {f(n): there exist positive constants c and n0 such that $0 \leq f(n) \leq cg(n)$ for all $n \geq n0$ }

- Ω-notation:

  The Ω-notation provides an asymptotic lower bound function. For a given function g(n), we denote by O(g(n)) the set of functions: Ω(g(n)) = {f(n): there exist positive constants c and n0 such that $0 \leq cg(n) \leq f(n)$ for all $n \geq n0$ } [1]

- True shooting (TS%):

  True Shooting Percentage; the formula is PTS / (2 * TSA). True shooting percentage is a measure of shooting efficiency that takes into account field goals, 3-point field goals, and free throws.

- PTS: Points.

- TSA: True Shooting Attempts; the formula is FGA + 0.44 * FTA.

- FGA: Field Goal Attempts (includes both 2-point field goal attempts and 3-point field goal attempts).

- FTA: Free Throw Attempts.

- Usage percentage (Usg%): (available since the 1977-78 season in the NBA); the formula is 100 * ((FGA + 0.44 * FTA + TOV) * (Tm MP / 5)) / (MP * (Tm FGA + 0.44 * Tm FTA + Tm TOV)). Usage percentage is an estimate of the percentage of team plays used by a player while he was on the floor.

- TOV: Turnovers (available since the 1977-78 season in the NBA).

- Tm: Team.

- MP: Minutes Played (available since the 1951-52 season).

- Assist: An assist is a pass that directly leads to a basket. This can be a pass to the low post that leads to a direct score, a long pass for a layup, a fast break pass to a teammate for a layup, and/or a pass that results

in an open perimeter shot for a teammate. In basketball, an assist is awarded only if, in the judgment of the statistician, the last player's pass contributed directly to a made basket. An assist can be awarded for a basket scored after the ball has been dribbled if the player's pass led to the field goal being made

- Rebound: A rebound occurs in basketball when a player gains possession of the basketball after a missed field goal, three-point field goal or free throw attempt.

- Defensive:

Step 3. Finding Creative Solutions

For this step, even if we can think of our own solutions, we look at specialized texts various ways to implement the different abstract data structures/types (ADTs) necessary and that best suit the different cases of each stage of the process of making basketball players queries. The methods considered as alternatives are as follows:

Alternative 1 - Use Self-balancing Binary Search Trees:

Alternative 2 - Use Hash Tables:

Alternative 3 - Use Lists: