# WHAT ARE THE PROBLEMS

In the recent years, the number of projects using machine learning has increased exponentially, as has the amount companies are investing in this technology.

This growth carries out with it a bunch of problems since the first models until now, such as the incorporation of machine learning models in production. Until 2022, up to Deborah Leff, former CTO for data science and AI at IBM, 87% of the data science projects never make it into production ( <https://venturebeat.com/ai/why-do-87-of-data-science-projects-never-make-it-into-production/> ) and among the 90% of companies that have made some investment in AI, fewer than 2 out of 5 report business gains from AI, improving this number to 3 out of 5 when we include companies that have made significant investments in AI (<https://sloanreview.mit.edu/projects/winning-with-ai/>) .

Those are the reasons why only a small percentage of the ML projects manage to reach production, being essential to find out what are the problems which come across since such an extraordinary inversion from the companies should never be wasted.

# STATE OF ART

In order to comprehend the current situation of the software industry, we must understand the previous steps of the software path.

# First software development: structured programming

The first software developments are about the 40s and 50s, when the first computer machines came out. Such was the rise of these new technologies that software´s developers had to give an extra twist to make them useful. So, they started developing software using *Structured Programming.*

Structured Programming is a trend that was born to make the life of the developers easier. It was not until 1966, when Böhm and Jacopini launched the structured program theorem, which says that any program could be wrote using just 3 instructions, when its consolidation began (https://www.edix.com/es/instituto/programacion-estructurada/).

In 1968, Edsger Dijstra published a well-known article, Go To Statement Considered Harmful (<https://homepages.cwi.nl/~storm/teaching/reader/Dijkstra68.pdf>), claiming the use of this new concept and the banish of the Goto sentence. Ending the consolidation of the structured programming.

As it has been said before, it is based in the 3 basic structures: sequence, selection or conditional and iteration. Those three basics made easier to understand the codes, with a more clear structure, better to optimize the testing and debugging and the maintenance expenses were reduced (<https://www.edix.com/es/instituto/programacion-estructurada/>).

# Waterfall methodology

At the beginning of the 70s the projects were complex enough having a long list of requirements, also needing a lot of documentation, alongside a proper design´s planning of the solution. Aiming to solve these new challenges, Waterfall methodology was born.

Based on a correct requirement´ definition, due they must be unchanged during all the process, it is a linear process of project management. Each step on the procedure cannot begin unless the previous phase is finished, and once finished, it is terminal, since Waterfall management does not allow you to return to a previous phase. (<https://www.lucidchart.com/blog/waterfall-project-management-methodology>)

Normally, waterfall methodology varies somewhat depending on the source, but they generally include:

* Requirements gathering and documentation.
* System design
* Implementation
* Testing
* Delivery/deployment
* Maintenance

# Agile methodologies

Due to the slowness and the delays caused by the traditional way of working, which used waterfall methodologies, the software industry ideated the agile methologies. The previous projects were based in fixed requirements, not able to change them once the process started, and the big efforts that were meant to suppose if a change was made drove to not-as-high-as expected quality projects.

In 2001, the Agile Manifest was created by the principal CEOs of the software industry in Utah (<https://agilemanifesto.org>). The Agile manifest was based in four keys:

* Iterations and individuals above processes and tools
* Functional products above exhaustive documentations.
* Partnership above deal negotiations.
* Change with the problem above and strict plan.

Diagrama

Descripción generada automáticamente