

# *Cost Estimation Guidance for AI Software Development Projects*

**Arlene F. Minkiewicz**

**ICEAA, Workshop, May 2025, Atlanta, GA**

## Contents

Introduction.....	1
AI through the years .....	2
Traditional Estimation Methods – Challenges faced with Estimating AI Software Developments .....	2
Key Factors Influencing Cost and Effort in AI Projects .....	4
Real World Use Cases of AI Development Projects. ....	10
Cost Estimation Framework for AI Software Developments.....	13
Next steps and Conclusions .....	15
Glossary of Terms .....	17
References .....	19

## Introduction

Artificial Intelligence (AI) has played an evolving role in software development, dramatically advancing over the years. Initially, AI applications were limited to rule-based systems, but modern AI now powers applications such as game-playing, medical diagnostics, natural language processing (NLP), self-driving cars, and complex military systems. The ability of machines to mimic human-like behavior has transformed industries worldwide, making AI a core component of modern software applications.

Cost estimation is an critical aspect of program management for software development projects of any kind. Software has become ubiquitous in almost every important system being deployed today. From smart phones, smart homes., smart cars to fighter jets, tanks, satellites and spaceships - more and more hardware functionality is being replaced by software. Consider the F-35 Joint Strike Fighter military aircraft as compared to its predecessor, the F-16, the F-35 has....

- 177x more computer code
- 300x software development costs
- 90% of the functionality is delivered by software <sup>1</sup>

Inaccurate cost estimation is one of the leading causes of project failures in software development. Studies show that over 50% of IT projects exceed their budgets due to underestimations in

complexity and resource allocation (Standish Group, 2022). AI software development requires a more nuanced approach to cost estimation, given its iterative nature, unpredictable outcomes, and high computational demands.

This paper will delve into the unique cost estimation considerations when a software development project delivers AI capability or features. The first section provides a brief history of the evolution of AI through the time of its inception to the present. The next section speaks to the challenges faced when traditional estimating techniques are applied to AI intensive software development. Following that, the paper will address the key factors that influence cost and effort in AI related projects. Use cases will be presented highlighting the potential cost challenges within real world AI developments. The next section proposes a framework for incorporating consideration of the unique AI aspects into a software estimate. Implications for this work and future steps will then be discussed.

## AI through the years

In the 1950s, AI research centered around rule-based systems and symbolic reasoning in an attempt to mimic human-like thought processes through logic-based programming. During the 60's and 70's, programs were developed that were capable of playing chess, proving theorems and mimicking some forms of human reasoning. Early research in AI was stymied by limited computational power and a lack of real-world data. In the 80's and 90's, expert systems emerged as a new application of AI. These systems were focused on simulating human decision-making skills through the codification of specialized knowledge. While expert systems demonstrated the potential value of AI in software development, they were limited in adaptability and failed to provide generalized intelligence. In the late 1990s and early 2000s, enabled by increases in computing power and accessibility of large amounts of digital data, Machine Learning (ML) techniques began to improve and gain momentum. ML proved to be a game changer for AI, leading to applications like fraud detection, predictive analytics, finance, medical diagnosis, and more. Deep learning, a subset of ML that uses neural networks with many layers has transformed AI in the software development domain, driving breakthroughs in fields such as computer vision, Natural Language Processing (NLP), and speech recognition. Google, Microsoft, OpenAI, and others have developed sophisticated models that understand language, recognize images and create original content in many forms.

## Traditional Estimation Methods – Challenges faced with Estimating AI Software Developments

Before considering the challenges with traditional estimation methods, a quick review of the most used methods is appropriate. Traditional software estimation methods are well-established and rely on historical data, expert judgement and structured methodologies to predict effort, cost and schedule for software projects. Below is a summary of these common techniques.

- **Expert Judgement** - This method leverages experience and insights from people familiar with similar software applications, market conditions, customers, organizational factors, etc., such as project team members and subject matter experts (SMEs). Delphi techniques are often employed to align expert opinions into a consensus-based estimate.

- **Analogous Estimation**- This method uses historical data (e.g. size, performance, complexity, personnel, actual effort, cost and schedule) from past projects with similar characteristics to estimate new projects.
- **Parametric Estimation** - This approach develops cost estimating relationships (CERs) based on historical data that has been collected from past projects and analyzed using mathematical processes. These CERs are used to generate new project estimates based on relevant parameters such as size, complexity, performance, personnel, organizational factors, etc.
- **Bottom-Up Estimation** - This method involves the decomposition of a project into smaller tasks, estimating cost and effort for each component, and then aggregating these estimates to produce an overall project estimate.

While these traditional estimation techniques work well for conventional software development, there are significant challenges when applied to AI-intensive projects due to the dynamic and unpredictable nature of AI development. The following limitations apply across all estimation methods.

- **Lack of Comparable Historical Data** – AI is a rapidly evolving field, making many AI projects unprecedented to some extent. In addition to this fact, many AI applications do not have reliable historical data, making it difficult to derive accurate estimates from past projects.
- **Rapid Technological Advancement** – AI models, tools, and best practices evolve and change frequently so even in cases where there is historical data from previous projects, it may quickly become outdated and lack relevance.
- **High Variability in Data Requirements** – Unlike traditional software, AI projects are heavily data-driven. The effort required for data acquisition, cleansing, labeling and augmentation varies widely from project to project and is challenging to estimate upfront.
- **Iterative and Experimental Nature of AI Development** – AI projects involve extensive experimentation, model tuning and retraining cycles. Unlike traditional software development, where tasks are generally clearly defined, AI workflows often evolve unpredictably, making estimating challenging (this is especially true when performing a Bottom-Up estimation)
- **Computational Resource Demands** – AI projects, particularly those involving deep learning, require high performance computing resources (e.g. GPUs, TPUs). The computational requirements fluctuate based on model complexity, dataset size, and tuning needs, creating additional estimation challenges.
- **Skill Set and Expertise Constraints** – AI projects require specialized expertise in machine learning, data science, deep learning and other advanced skills. The availability and cost of these professionals vary significantly, potentially pushing the estimator out of their comfort zone with respect to the cost of staffing an AI project
- **Deployment and Maintenance Complexity** – AI models require continuous monitoring, retraining and adaptation to prevent model drift as real-world data changes. This ongoing maintenance introduces long-term costs atypical of more traditional applications that are challenging to estimate.

- **Scope Creep and Evolving Requirements** – Based on their experimental nature, AI projects often experience scope adjustments. An example of this might be the need to collect additional data if model performance does not meet requirements.
- **Complexity of AI Quality Assurance and Testing** – Unlike traditional software, AI models produce probabilistic outputs, requiring extensive validation frameworks to ensure fairness, bias mitigation and reliability. These testing requirements are often outside the purview of most traditional test scenarios and may be underestimated or completely overlooked.

## Key Factors Influencing Cost and Effort in AI Projects

### Data Preparation

The quality and availability of data significantly influences the cost and timeline of AI projects. Data collection can be expensive, especially if large, high-quality datasets are required. The cost increases further when data must be manually labeled or annotated, a common requirement in supervised learning. Additionally, data cleansing and preprocessing require substantial effort and computational resources (examples of this activity include handling missing values, normalizing parameters and features, removing biases) The complexity grows when dealing with unstructured data such as images, video, or text, necessitating advanced processing techniques and specialized tools. According to (Avinash, et al, 2022)<sup>3</sup>, data preprocessing is one of the most crucial starting points for creating a model that provides quality results and can take up to 60% of the total time in the machine learning process.

#### Key Cost Considerations in Data Preparation

- **Data Collection** – Data can be acquired from public sources, can be purchased from third party vendors or can come from generating or acquiring proprietary datasets. In all these cases, data acquisition can be expensive. Costs will vary depending on the domain. Open-source datasets are free, but collection of data requires effort. Proprietary and third party (commercial) datasets may be acquired for a price, which must be included in an estimate (along with any ongoing maintenance or licensing costs) or, in the case where the data set is internal (or internally generated), may still require some degree of manipulation and massaging to get into a usable format.
- **Data Cleaning and Preparation** – Data wrangling may consume significant amounts of an AI project's time and cost. There are several factors that help quantify the complexity of this task; the most critical of which are:
  - **Data Imbalance and Missing Values** – this task requires cleaning imbalanced data which if not addressed could lead to biased model predictions, skewed performance metrics or the failure to learn meaningful patterns for the minority class. Missing values in a dataset generally require some sort of interpolation or annotation or could potentially result in the necessity for additional data collection
  - **Format Standardization** - this task involves converting data from various formats (CSV,JSON,XML,etc.) into a unified format

- Noise-Reduction and Deduplications – the noise reduction tasks involve identifying and removing irrelevant, erroneous, or misleading data which may arise from sensor errors, human errors, environmental factors or other issues with systems or processes. Deduplication is the process of identifying and removing duplicate records in a dataset which may interject bias in the model.
- **Data Annotation and Labeling** – this activity is essential if supervised learning techniques are employed but can be one of the largest hidden costs in AI projects. This task can either be a manual process or an automated process. The manual process can be labor intensive with cost ranging from \$0.02-\$15 per data point<sup>4</sup>, while the automated process may require investment in tools but generally tends to reduce the overall cost of the activity.

### **Key Cost Drivers for Data Preparation**

The most important factors to consider for data preparation in an AI project are:

- The volume of data required (for context consider that deep learning models need millions of samples while more traditional ML models can work with thousands)
- Annotation costs per data point (for context a complex medical imaging model is more expensive than one that labels social media interactions)
- Cleaning overhead is driven primarily by the quality of the dataset. AI models trained on clean, well annotated data achieve up to 30% higher accuracy; reducing rework and training iterations. Poor data quality can increase project costs by 50-100%<sup>5</sup>.

### **Algorithm Selection**

Choosing the most appropriate machine learning or deep learning algorithm before the training begins is crucial, as it directly affects development costs and system performance. More complex models, such as deep learning networks, often require extensive computation and data, increasing both hardware costs and energy consumption. Conversely, simpler models, like decision trees or linear regression may be less expensive but might not produce the desired accuracy. The trade-off between model complexity and performance is an important consideration in cost estimation. According to the research done in (Kus, 2024)<sup>6</sup>, choosing the wrong algorithm can increase training costs by 2-5x due to inefficiency. The choice of algorithms also affects explainability, scalability and deployment costs, making it a critical factor in project planning.

### **Key Cost Considerations for Algorithm Selection**

- **Model Complexity and Compute Requirements** – the selection of an algorithm leads directly to the compute intensity requirements, which will directly impact the cloud/GPU/TPU costs. Lightweight models, suitable for structured data will require minimal compute power, Deep learning models will demand high-end GPUs/TPUs along with extensive tuning (human resources)
- **Algorithm Optimization Needs** – some algorithms require more tuning than others, driving cost and time of the project up. Hyperparameter tuning can increase training time exponentially, though through pruning and quantization, inference costs can be

reduced, but not without expertise during data preparation (which comes with its own costs) According to (Liang, et al, 2021)<sup>7</sup>, Pruning reduces cloud inference costs by 30-50% and quantization can shrink model size by up to 75% - cutting memory and compute costs by 50-80%

- **Model Training Strategy** - there are several strategies to train a model – consideration of the best strategy is a crucial part of the algorithm selection process necessitating a critical cost vs performance analysis to determine the best strategy for a particular project. Training from scratch involves building and training a new model from random initial weights, full training involves training the model from scratch using an established dataset, while fine-tuning involves the adaptation of a pre-trained model to a specific task (Kumari, 2024)<sup>8</sup>. As one goes from a from-scratch to a full training to a fine-tuning strategy, one trades complete control for efficiency and cost.

## **Model Training**

Training an AI model is one of the most resource intensive phases of development. The cost is influenced by factors such as dataset size, computational power and the number of iterations required for optimization. Cloud based AI training can help mitigate the infrastructure costs, but pricing varies based on the type and duration of GPU or TPU usage. Studies have indicated that training a large deep learning model, such as GPT-like architectures, can cost millions of dollars due to hardware, power consumption and specialized engineering efforts. Furthermore, hyperparameter tuning and model experimentation can further add to the development effort, requiring both automated and manual interventions.

### **Key Cost Considerations for Model Training**

- **Training Duration and Compute Costs** Training an AI model requires specialized hardware. Small ML models, such as decision trees, linear regression, etc. can be trained on regular CPUs in a relatively short period of time while deep learning models (CNNs, Transformers) require powerful GPUs or TPUs and extensive compute times with potentially large resource costs per iteration. Figure 1 from The AI Index 2024<sup>9</sup>, (using Epoch 2023 dataset) shows the estimated training costs for select frontier models between 2016-2023. Note that GPT-4 and Gemini have estimated training costs in the range of \$100M.

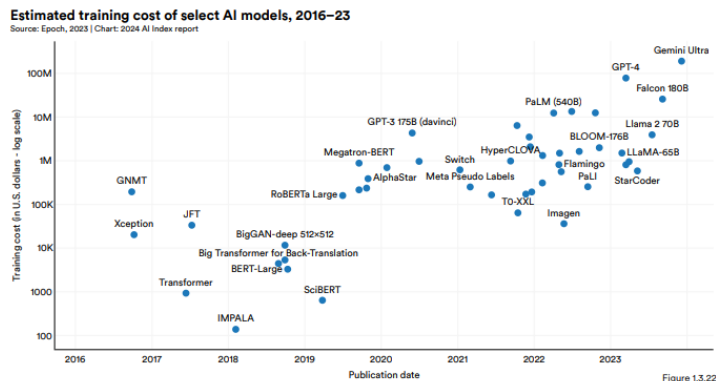


Figure 1 – Estimated training costs of select AI Models, 2016-2023 (Epoch Data)

- **Parallelism and Optimizations** – There are several optimization strategies that can be applied to impact the efficiency, speed, and cost of training deep learning models through the management of computation resources, reduction of memory constraints and improving model convergence. Batch size tuning refers to a method of model training that processes batches of training samples before the model updates its weights (as opposed to updating model weights after each sample is processed). Another strategy, distributed training, splits the computations across multiple GPUs, TPUs or cloud instances, significantly reducing training time but requiring additional GPUs and additional expertise.
- **Energy Consumption.** – AI training is energy intensive; a single Transformer training session can emit as much-CO2 as five cars over their lifetimes. (Hao, 2022)<sup>10</sup> Large companies like Google DeepMind optimize costs by using TPUs, which are 10-30x more efficient than GPUs
- **Cloud vs On-premises training** – Cloud based AI training can reduce upfront costs but incurs ongoing fees throughout the training process. On-premises infrastructure has high upfront costs but can save long term operational expenses
- **Experimentation and Retraining Costs** – In order to optimize hyperparameters, AI models require multiple iterations where each iteration increases cloud computing costs. Training cycles can be reduced through transfer learning. Transfer learning is the process of taking a pre-trained model (trained with a large dataset) and adapting it to a different but related task. This approach can significantly reduce training time and computational costs as well as the dataset size.

## Deployment and Maintenance

Deploying an AI model into production is only the beginning of its lifecycle. Unlike traditional software, AI models require continuous monitoring, retraining and updates to ensure they remain accurate and effective over time. According to (Sculley, et al, 2015)<sup>11</sup> maintaining an AI system often incurs higher costs than initial development due to issues such as data changes, unexpected model behavior and security risks. Model drift occurs when real world data distributions change; this drift necessitates regular fine-tuning or retraining, adding substantial operational costs. Additionally, AI driven applications must be optimized for latency, scalability, and cost efficiency. Edge AI

deployment, which involves running AI on local devices rather than the cloud, can reduce inference costs but requires additional hardware investment.

### **Key Cost Considerations for Deployment and Maintenance**

- Cloud vs Edge Deployment – use of the cloud reduces initial investment but incurs on-going usage-based costs. Edge AI reduces cloud costs but requires specialized hardware.
- Retraining and Maintenance - because AI models decay over time due to model drift, they need constant monitoring and care. Depending on the size and complexity of the AI model and the scope of the drift, the cost of regular retraining could become a significant part of the operational costs of the AI model.
- Scalability Costs – Real time inference can increase cloud costs by as much as 10x as compared to batch processing. Real time inference involves processing data and generating predictions instantaneously as new data arrives. This approach is essential for applications requiring immediate responses such as fraud detection, autonomous vehicles and live customer interactions. It also requires maintenance of the infrastructure necessary for constant resource availability, driving operational costs up. Batch processing involves collecting data over a period of time and processing it at scheduled intervals. Batch processing allows for optimization of resource utilization because tasks can be scheduled during off-peak hours, reducing costs.

### **Team Experience**

The expertise of the AI development team directly impacts project costs and success. AI projects demand a diverse set of skills, including machine learning engineering, data science, cloud computing along with domain expertise. Salaries for AI professionals are among the highest in the tech industry. Companies often face a dilemma between using in-house talent (through training or hiring) versus outsourced AI development professionals. While outsourced resources can reduce short term costs, it may limit customization and lead to longer term dependency costs. Investing in AI training and upskilling existing employees can be a cost-effective alternative.

### **Tool Selection and Maturity**

Selecting the right AI development tools and frameworks is crucial for balancing cost, efficiency, and scalability. Open-source tools such as TensorFlow, PyTorch, and Scikit-learn offer cost-effective and flexible solutions, eliminating licensing fees while benefiting from active community support and frequent updates. These frameworks do require in-house expertise for setup, optimization and maintenance.

Enterprise grade solutions like IBM Watson, Google Vertex AI and Microsoft Azure AI provide pre-configured environments, managed services and enterprise support, making them easier to deploy and scale. Such platforms often include automated ML tools, security compliance features, and integration with cloud ecosystems, leading to reductions in engineering effort and operational risk. They do, however, also come with subscription-based pricing models that can lead to high long-term costs.

Tool maturity is also an important consideration. Mature frameworks have extensive documentation, stable APIs, and widespread adoption, making debugging and troubleshooting more efficient. According to a Gartner report (2023), organizations leveraging mature, well supported AI platforms experience significantly lower development overhead compared to those relying on experimental tools.

### **Key Cost Considerations in Tool Selection and Maturity**

- **Licensing Fees** – Open-source tools reduce costs, while enterprise platform charges can be substantial. Licensing fees are on-going so a long-term ROI analysis while selecting an enterprise platform is wise.
- **Setup and Maintenance** – Open-source tools most likely require some inhouse support for the setup and maintenance activities, while enterprise solutions provide managed services
- **Integration Costs** – mature tools integrate easily while experimental/custom tools may increase debugging and engineering time significantly
- **Scalability** – Enterprise solutions scale easily while open-source tools may require manual intervention for performance optimization
- **Long-Term cost Tradeoffs** – Enterprise solutions are not always the most expensive solution. A large-scale deployment could easily become cost prohibitive if open source or experimental tools were chosen.

### **Application Platform**

While touched on in several of the previous paragraphs in this discussion, the deployment platform plays a role significant enough to warrant a summary of the issues alluded to earlier. The deployment platform plays a critical role in the performance, cost and scalability of AI applications. Organizations must choose between on-premises, cloud-based or edge computing solutions based on factors such as security, operational cost, and real-time processing needs. Developers of AI solutions have several options as well as the option to select a platform strategy that is a hybrid of more than one of the platforms discussed. According to (McKinsey's State of AI 2022)<sup>12</sup> companies that optimize their platform strategy can reduce AI operating expenses by up to 30% by choosing the right mix of cloud, on-premises and edge computing.

### **Key Cost Considerations for Application Platform**

- **Cloud-based AI Solutions** (AWS, Google Cloud, etc.) provide high scalability, flexible pricing models, and managed AI services, reducing the need for heavy upfront infrastructure investment. Cloud costs are recurring costs and will accumulate over time, especially for high-frequency inference workloads
- **On-premises deployment** offers more control over data security and compliance, which is particularly important in regulated industries like finance, healthcare, and government. While it requires initial capital investment in servers and GPUs, it can reduce long term operational costs for companies with predictable workloads. Enterprises training large AI models may find on-premises infrastructure cheaper in the long run compared to cloud-based solutions

- Edge AI deployment allows AI models to run directly on IoT devices, mobile applications or embedded systems, reducing latency and cloud dependency. This is particularly beneficial for autonomous vehicles, industrial automation and real-time video analytics. Edge AI often requires specialized hardware, increasing initial investment in development and hardware costs.

### **Integration Challenges**

AI systems often need to integrate with existing enterprise applications, databases or IoT devices, which can introduce compatibility issues and impact costs. Legacy systems, built with outdated architectures, may require extensive modifications to accommodate AI functionalities. There have also been issues cited associated with ensuring API compatibility, data pipeline synchronization, and real-time processing capabilities that require custom engineering, impacting development costs.

#### **Key Cost Considerations for Integration Challenges**

- **Compatibility with Legacy Systems** – Often AI applications will need to integrate with one or more of an organization’s legacy systems. It is important to ensure that the AI application aligns with the existing system’s architecture; modifications required to achieve alignment will increase development cost and time. Data formats and communication protocols must also be compatible between the legacy system(s) and the AI application to ensure seamless interaction between the two. Extensive testing may be required to ensure that the integrated system performs correctly.
- **Data Integration and Management** – Often the integration with legacy systems requires efforts to integrate and manage the datasets between the disparate systems. Data consolidation involves the effort associated with aggregating data sources to create a unified dataset for the AI application. Post integration, it is often necessary to implement processes to ensure data accuracy, consistency and completeness. All of these data related activities associated with integration will increase the amount of effort associated with deploying and delivering the AI Application.
- **Scalability and Performance Optimization** - Effort is likely to be required to ensure that the integrated system can handle increased loads without performance degradation and to optimize resource usage to maintain system efficiency. This is likely to involve upgrades to hardware or cloud resources to support scalability needs and the addition of time to the schedule for performance optimization tasks to ensure operational efficiency.

## **Real World Use Cases of AI Development Projects.**

### **1. OpenAI’s GPT-5 (Orion) Project**

The Orion Project has encountered significant challenges, particularly concerning escalating costs and resource constraints. The financial demands of training GPT-5 are substantial. Each six-month training run is estimated to cost approximately \$500 million in computing resources alone (Burke 2024)<sup>13</sup>. Despite these investments, the performance improvement over previous models has been marginal, raising concerns regarding the ROI. A significant hurdle has been the scarcity of high-

quality training data. Traditional data sources have proven insufficient, prompting OpenAI to explore alternative strategies, such as generating synthetic data and hiring specialists to create new datasets. These approaches, while potentially effective, introduce additional costs and complexities. The Orion journey exemplifies the multifaceted cost challenges inherent in advancing AI technologies. It underscores the meticulous planning, flexible strategies and proactive risk management in the financial oversight of AI development endeavors.

## **2. Stability AI's Stable Diffusion Model**

The development of Stable Diffusion, an open-source text to image generative model, has been marked by significant cost challenges, prompting innovative strategies to achieve cost efficiency. Initially, training the first version of Stable Diffusion required substantial computational resources, utilizing over ~150,000 GPU hours, which incurred an estimated cost of \$600,000<sup>14</sup>. Further phases of model training introduced innovative techniques to enhance cost efficiency. One approach according to (Kim, et al, 2024)<sup>15</sup> involved the use of block pruning and feature distillation, which reduced the model size by 30-50%. This method maintained competitive performance while significantly lowering computational requirements, making the training process more accessible and less resource intensive. Additionally, the implementation of online and offline preprocessing strategies has been explored to optimize training throughput and cost. By employing distributed computing frameworks, researchers achieved a significant reduction in pre-training costs, demonstrating the potential of scalable solutions in managing large scale datasets efficiently.

## **3. Tesla's Full Self-Driving (FSD) Technology**

Tesla's Full Self-Driving technology represents a significant endeavor in the automotive industry, aiming to achieve fully autonomous vehicles. This project provides an excellent opportunity to highlight cost challenges associated with AI applications with real time implications. The software component of FSD has presented considerable challenges. Tesla's approach relies heavily on computer vision and machine learning algorithms, necessitating the collection and processing of vast amounts of driving data. As their scalability quest includes self-driving on various continents, their datasets need to include driving data and regulatory information from across the world. This data intensive strategy requires significant computational infrastructure, leading Tesla to develop its own supercomputer, known as Dojo, designed to handle the massive datasets involved in training FSD algorithms. The development and deployment of such infrastructure represents a considerable financial commitment. Additionally, Tesla has faced regulatory scrutiny and legal challenges related to the marketing and safety of its FSD technology, which have resulted in increased legal expenses and potential delays in deployment.

## **4. Apple's Project Titan**

Apple's Project Titan was an ambitious initiative launched in 2014 with the goal of developing an electric, self-driving vehicle that would revolutionize the automotive industry. Despite Apple's reputation for innovation, the project faced numerous challenges, leading to its cancellation in February 2024. Though the project failed there are things to be learned from this project. Over its decade-long pursuit, Apple invested more than \$10 billion into Project Titan. This substantial expenditure encompassed research and development, talent acquisition, and prototype development. At its peak, the project employed over 2,000 specialists, including experts poached

from leading automotive companies. The significant financial outlay, coupled with the high costs of developing proprietary automotive technology, placed considerable strain on Apple's resource. A major cost challenge stemmed from internal indecision regarding the project's direction. Initially, Apple aimed to build a fully autonomous vehicle without steering wheels or pedals, a concept that required unprecedented technological advancements. However, as technical and regulatory hurdles emerged, the company oscillated between developing a complete vehicle and focusing solely on autonomous driving software. This lack of a clear, consistent strategy led to repeated redesigns and shifts in focus, resulting in resource misallocation and escalating costs. Entering the automotive manufacturing sector presented significant challenges for Apple, a company traditionally centered on consumer electronics. Efforts to partner with established automakers, such as Hyundai and Volkswagen, were explored but ultimately did not materialize. The absence of a manufacturing partner meant that Apple would have to build production capabilities from the ground up, a venture requiring massive capital investment and expertise. This fact contributed to the project's unsustainable financial trajectory.

#### **5. Lockheed Martin's F-35 Lightning II Program**

The Lockheed Martin F-35 Lightning II program integrates advanced AI components to enhance its operational capabilities. However, the incorporation of these AI systems has introduced significant cost challenges throughout the program's development and sustainment phases. A pivotal AI-driven component of the F-35 is the Autonomic Logistics Information System (ALIS). Designed to streamline maintenance and operational logistics, ALIS manages functions such as mission planning, supply chain oversight, and aircraft diagnostics. Despite its intended benefits, ALIS has faced persistent issues, including inaccurate data outputs and system malfunctions, leading to increased maintenance workloads and aircraft downtime. To address obsolescence and augment the F-35's computational power for future AI applications, the program initiated the Technology Refresh 3 (TR-3) upgrade. This upgrade encompasses enhancements to the aircraft's displays, processing units and memory systems. However, the TR-3 initiative has encountered production delays and hardware shortages, contributing to a projected cost overrun approaching \$1 billion. The integration of sophisticated AI systems has also influenced the F-35's sustainment expenses. The DoD estimates that operating and maintaining the planned fleet of 2,470 F-35 aircraft through 2088 will exceed \$2 trillion. This figure reflects not only the costs of routing operations but also the continuous updates and maintenance required for the aircraft's advanced AI and other software system<sub>16</sub>

#### **6. US Air Forces Collaborative Combat Aircraft (CCA) Program**

The US Air Force's Collaborative Combat Aircraft (CCA) program is a pivotal initiative aimed at integrating unmanned autonomous aircraft alongside manned fighter jets to enhance combat capabilities and operational flexibility. The CCA program envisions deploying a fleet of AI-enabled, uncrewed, aircraft designed to operate in concert with fifth and sixth generation manned fighters. These collaborative aircraft are intended to perform a variety of roles, including reconnaissance, electronic warfare, and strike missions, thereby augmenting the effectiveness of human pilots and expanding the operational reach of the Air Force. The initiative is a component of the broader Next-Generation Air Dominance (NGAD) strategy, which seeks to maintain US air superiority through advanced technologies and integrated systems. The Air Force aims to procure CCAs at a unit cost

ranging between \$25 and \$30 million, approximately one third to one half the cost of a manned F-35 fighter jet. Achieving this target requires careful consideration of design, production and operational factors to ensure that cost-efficiency does not compromise mission effectiveness (Congressional Research Service, 2025)<sup>17</sup>.

## Cost Estimation Framework for AI Software Developments

The cost estimation framework is intended as a tool to help software cost estimators and software engineering teams to have productive discussions about objectives, goals and challenges of planned AI development projects in order to develop a defensible and credible estimate of the cost and effort associated with said projects. It is also intended to help the estimators create a data collection process to facilitate ease of future AI estimation needs. The steps of the cost estimation framework are described below.

### 1. Project Scope Definition

Defining the project scope is the foundational step in estimating development costs. A well-defined scope ensures that all stakeholders align project objectives, feature sets and data requirements. The first step in project scope definition is objective identification, where project goals and expected outcomes must be clearly stated. To predict costs effectively early on, a realistic assessment of the type of project – predictive analytics, autonomous decision-making, real-time processing, etc., is crucial. Defining objectives upfront helps establish the right budget and resource allocation strategy. Feature set identification is another crucial aspect of scope definition. AI models require functionalities such as computer vision, NLP, reinforcement learning or deep learning, each with different cost implications. It is also important to make sure that data requirements are established early in the development process. AI development is highly data-dependent, and a cogent understanding of the volume, quality and sources of data should be identified as early in the process as possible.

### 2. Phased Development Approach

AI development is inherently uncertain, requiring an iterative approach with multiple checkpoints. The cost estimation process must account for each phase of development, ensuring that budgets remain flexible as insights emerge.

#### Phase 1 – Planning

This phase includes conducting feasibility studies to assess whether AI is the right solution. It is important to assess key risks such as data availability, computational costs, scalability issues, etc. Stakeholders should be identified, and a communication plan should be developed with owners and milestones identified.

#### Phase 2 – Data Preparation

This phase includes a complete assessment of the data needs of the project. It is important to estimate the costs for data collection, cleaning, and preparation, taking into consideration the types of data, unique data characteristics, potential data sources, and any potential obstacles to data collection. An important part of the data preparation phase should also include an

assessment of the methods for data acquisition – will data be collected from open-source domains and what are the human resources associated with collection or will it need to be purchased; are there on-going licensing issues associated with maintaining the data? What are the cost issues associated with storing the data (cloud vs on-premise) and keeping it up to date?

### **Phase 3 – Algorithm Selection and Development**

Choosing the right AI algorithm significantly impacts development costs. Many projects benefit from fine tuning pre-trained models when this makes sense; training a model from scratch requires massive datasets and extensive computing power but is sometimes necessary. It is important to determine up front what the best solution is based on the project under consideration.

### **Phase 4: Model Training and Tuning**

AI training requires intensive computational resources, particularly for deep learning models. Model tuning involves hyperparameter optimization, which can extend training time by 30-50%, increasing cloud costs and/or energy expenditures.

### **Phase 5: Deployment**

Deploying an AI model involves integrating it into an existing enterprise system, mobile app, or cloud infrastructure. Cloud-based AI services such as AWS SageMaker or Google Vertex AI offer scalable solutions but can be high cost for inference workloads. Edge computing, where AI models run on local devices, can be more cost effective but require high initial hardware investments

### **Phase 6: Maintenance and Monitoring**

AI models require continuous updates due to data drift, adversarial attacks and evolving regulatory requirements. Continuous monitoring tools help track model degradation and enable proactive updates.

## **3. Cost Factors and Estimation techniques**

Traditional estimation methods are good but need to be tuned for the unique nature of AI development. There are various methods of estimation, and it is best if a hybrid solution encompassing more than one methodology is applied.

- Expert judgement – experts continue to be valuable, but their opinions should be metered by experience with AI projects
- Analogous estimates – analogy is a tough choice since the likelihood of having a similar AI project is rare – the technology is too new and is rapidly evolving.
- Parametric Estimation – these models are based on historical data analysis, so they are biased based on the projects they were trained on. Size and Complexity are still important parameters for estimation, but it is also important to remember that with AI projects, the dataset size, quality, and uniqueness, along with model training information, are also important parameters that will influence the estimate
- Bottom-Up Estimation – An estimate based on a rolled-up vision of the project is not a bad place to start, but it is important to take into account the iterative nature of an AI

project and incorporate the fact that there is uncertainty with how the project will progress.

#### **4. Dynamic and Iterative Estimation Process**

AI projects require flexibility, as costs often evolve over time. Unlike traditional software, AI models improve with continuous training and retraining, requiring a budget for future model iterations. A dynamic estimation process updates costs based on newly discovered challenges, helping stakeholders adjust budgets proactively. Successful estimators will embrace techniques similar to agile projects; expect things to change, embrace the change, communicate the risks to the stakeholders, remember that flexibility is key to successful projects, don't expect to get it right the first time, and most importantly - establish success metrics and track to learn from each project.

#### **5. Team Expertise and Resource Allocation**

Hiring top AI talent is a major cost driver. Salaries for machine learning engineers are top-end and organizations need to understand that it is important to recognize the value of talent. An organization that lacks AI expertise should consider the value of outsourcing AI development for the short term, though they should consider developing internal talent for the long term as AI is here to stay. It is also important for organizations to assess the maturity and capability of tools and existing hardware. AI solutions require specific tools and expertise; organizations that want to embrace AI should understand what the investment costs are.

#### **6. Risk Management**

All AI projects come with inherent risks. Data quality issues, algorithm bias and regulatory constraints can result in costly rework. Having contingency funds and risk mitigation strategies can prevent cost overruns. Stay away from a point estimate, give a range that recognizes the uncertainty in estimation decisions

#### **7. Case Study Review and Validation**

Case studies from similar AI projects provide valuable insights into budgeting and best practices. Reviewing past projects and incorporating lessons learned from previous projects helps refine cost estimate for future AI initiatives.

## **Next steps and Conclusions**

The framework presented to date is based on lessons learned from recent AI projects as well as common sense, the things that all seasoned estimators consider when they are estimating, especially when what they are estimating is somewhat beyond their experience or expertise. And while common sense tells us that AI projects are not new, we also realize that the speed with which technology is advancing creates a situation where history may not be as relevant as we would like. The proposed framework does not answer all the questions about how much the next AI development project will cost; it does provide a structure for estimators to consider how to estimate such a software project and as importantly, creates a structure for collecting data to support future

estimates. The next step in this process is to identify AI projects where quantitative data can be collected and lessons learned can be used to improve and codify guidance.

AI is here to stay, and the wise among us are going to need to learn how to live side by side with the ChatGPTs and Copilots among us. The challenge is great, but the rewards may be significant if we are true to what we know and willing to learn what we can from a machine in a productive and cautious way. As cost estimators, we are practiced in the art of predicting the future about projects that have yet to be completed. AI presents a new and riveting challenge, while there are significant advances in technology that propel AI into places we are uncomfortable with, at the end of the day, we still need to provide clear, defensible estimates that include our understanding of the past and our knowledge of what we think the future will bring. A framework will not answer all the questions that arise during an estimate, it will however arm the estimator with an additional tool to help guide their thought processes and provide them with some sensible and meaningful questions to pose to the development team in order to inform their estimate with the best information possible.

## Glossary of Terms

**Convolutional Neural Networks (CNNs)** are specialized neural networks primarily designed for processing structure grid data, such as images. They are adept at capturing spatial hierarchies and patterns using convolutional layers.

**Compute** - in the context of AI, the word compute refers to the computational resources required to train and run AI models. This encompasses the processing power, memory, and storage necessary to handle large datasets and perform complex calculations inherent in AI tasks.

**Data imbalance** – in machine learning this refers to a scenario where certain classes or outcomes are significantly underrepresented compared to others within a dataset. This disproportion can lead to models that are biased toward the majority class, often resulting in poor predictive performance for the minority class.

**Deduplication** - Process of identifying and removing duplicate records in a dataset to ensure data quality and prevent redundancy

**Distributed computing frameworks** are software systems designed to facilitate the processing of data and computational tasks across multiple interconnected computers, often referred to as nodes, working collaboratively to achieve a common objective. These frameworks provide the necessary infrastructure, tools and programming models to enable developers to build and execute applications that can efficiently utilize the combined resources of a cluster or network of machines. By distributing workload, these frameworks enhance performance, scalability and fault tolerance, making them essential in handling large-scale data processing and complex computations.

**Feature Distillation** is a specialized technique within the broader framework of knowledge distillation in machine learning. It focuses on transferring the internal representations, or features, learned by a large, complex model (the teacher) to a smaller, more efficient model (the student). This process aims to enhance the performance of the student model by enabling it to mimic the feature extraction capabilities of the teacher model, thereby achieving comparable accuracy with reduced computational requirements.

**GPUs** – Graphic Processing Units (GPUs) are specialized electronic circuits originally designed for rendering graphics in computers, gaming consoles and professional workstations. However, due to their high parallel processing capabilities, GPUs have become essential for AI deep learning, scientific computing and high-performance data processing.

**Hyperparameters** – in machine learning, hyperparameters are external configurations set before the training process begins, guiding the overall learning process of a model. They differ from model parameters, which are internal values learned from training data, such as weights in a neural network. Hyperparameters influence various aspects of model training and architecture, including complexity, learning rate, and the number of iterations.

**Inference costs** are the computational, energy and financial expenses incurred when an AI model makes predictions (or inferences) on new data. Unlike training costs (one time) inference costs are ongoing expenses

**Model drift – (or model decay)** refers to the gradual decline in the performance of a machine learning model over time. This degradation occurs as the statistical properties of the input data or the relationship between input features and the target variable change, leading to less accurate predictions. This drift can occur when the underlying relationship between input features and target variables change or when the statistical properties of the input data change.

**Pruning** - the process of removing redundant or less significant parameters in Neural Networks without impacting accuracy.

**Quantization** reduces the numerical precision of model weight and activations to decrease memory usage and improve computational efficiency.

**Supervised Learning** – refers to a form of machine learning where an algorithm is trained using a labeled dataset to predict outcomes or classify data. The algorithm learns to recognize patterns and relationships between inputs and outputs, enabling it to make accurate predictions on new, unseen data.

**TPUs – Tensor Processing Units** (TPUs) are specialized hardware accelerators designed by Google specifically for AI, deep learning and machine learning workloads. TPUs excel at handling large-scale deep learning tasks, particularly those involving extensive matrix computations such as training and inference for complex neural networks

**Transformers** - have revolutionized the handling of sequential data, particularly in NLP. Unlike recurrent neural networks (RNNs), transformers process input data in parallel, significantly improving training efficiency.

## References

- [1] ICEAA Cost Estimating Body of Knowledge – Software, Lesson 1, available at <https://www.iceaaonline.com/cebok-s/>
- [2]. The Standish Group (2022), Chaos Report, ,available at [www.standishgroup.com/files/billboard/2022-Predictions-Layout2.pdf](http://www.standishgroup.com/files/billboard/2022-Predictions-Layout2.pdf), Retrieved December 21024
- [3] Avinash, et al, “Automated *Machine Learning-Algorithm Selection with Fine-Tuned Parameters*”, Proceedings of the Sixth International Conference on Intelligent Computing and Control Systems, 2022.
- [4] Elite Data Labs, “*Breaking Down AI Annotation Costs: What You Should Expect in 2025*”, 2025, Available at <https://aidatalabelers.com/how-much-do-ai-data-annotation-services-cost-in-2025-the-complete-guide/>, Retrieved January 2025
- [5] Anglen,J., “*The Critical Role of Data Quality in AI Implementation*”, Rapid Innovation, available at <https://www.rapidinnovation.io/post/the-critical-role-of-data-quality-in-ai-implementations>, , retrieved Feb 2025
- [6] Kus, et.al., “*Frugal Algorithm Selection*”, Cornell University ArXiv, May 2024, available at <https://arxiv.org/pdf/2405.11059>, retrieved January 2025
- [7] Liang, et. al, “*Pruning and Quantization for Deep Neural Network Acceleration: A Survey*”,2021, available at <https://arxiv.org/pdf/2101.09671>, retrieved Feb 2025
- [8] Kumari, “*Fine-Tuning vs. Full Training vs Training from Scratch in Deep Learning*”, Analytics Vidhya, 2024 available at <https://www.analyticsvidhya.com/blog/2024/06/fine-tuning-vs-full-training-vs-training-from-scratch/>, Retrieved January 2025
- [9] *AI Index Report, 2024, Stanford University, HAI (Human-Centered Artificial Intelligence, available at <http://aiindex.stanford.edu/report/>*, Retrieved January 2025
- [10] Hao, Karen, “*Training a single AI model can emit as much carbon as five cars in their lifetimes*”, MIT Technology Review,2022, available at <https://www.technologyreview.com/2019/06/06/239031/training-a-single-ai-model-can-emit-as-much-carbon-as-five-cars-in-their-lifetimes/> Retrieved Feb 2025.
- [11] Sculley, et al, “*Hidden technical debt in machine learning systems*”, Proceedings neurips.cc, 2015, available at [https://proceedings.neurips.cc/paper\\_files/paper/2015/file/86df7dcfd896fcdf2674f757a2463eba-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2015/file/86df7dcfd896fcdf2674f757a2463eba-Paper.pdf), Retrieved January 2025.
- [12] McKinsey, “*The State of AI in 2022- and a half decade in review*”, Quantum Black, AI by McKinsey, December 2022, available at <https://www.mckinsey.com/~media/mckinsey/business%20functions/quantumblack/our%20insights/the%20state%20of%20ai%20in%202022%20and%20a%20half%20decade%20in%20review/the-state-of-ai-in-2022-and-a-half-decade-in-review.pdf> , Retrieved December 2024

[13] Burke, Bruce, "The Next Great Leap in AI Is Behind Schedule and Crazy Expensive", *Neural News Network*, 2024 available at <https://remunerationlabs.substack.com/p/the-next-great-leap-in-ai-is-behind>, retrieved December 2024

[14] Wikipedia, "Stable Diffusion", available at [Stable Diffusion - Wikipedia](#), retrieved Feb 2025

[15] Kim, B. et al, "BK-SDM: A Lightweight, Fast, and Cheap Version of Stable Diffusion", *arXiv*, Dec 2024, available at <https://arxiv.org/pdf/2305.15798>, , Retrieved Jan 2025

[16] GAO, U.S. Government Accountability Office, "The F-35 Will Now Exceed \$2 Trillion As the Military Plans to Fly It Less", May 2024, available at <https://www.gao.gov/blog/f-35-will-now-exceed-2-trillion-military-plans-fly-it-less>, Retrieved Jan 2025

[17] Congressional Research Service, "U.S. Air Force Collaborative Combat Aircraft", Jan 2025, available at <https://crsreports.congress.gov/product/pdf/IF/IF12740>, retrieved Jan 2025