Towards a Fast Cost Estimation Supported by Large Language Models

Sandro Hartenstein[1,2], Sidney Leroy Johnson[1], Andreas Schmietendorf[1,2]
sandro.hartenstein@hwr-berlin.de, s_johnson20@stud.hwr-berlin.de, andreas.schmietendorf@hwr-berlin.de

[1] *Berlin School of Economics and Law, Alt-Friedrichsfelde 60, 10315 Berlin, Germany*

[2] *Otto-von-Guericke-University Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Germany*

Abstract

This elaboration describes the conceptual identification of the capabilities of large language models for cost estimation. For this purpose, the possibilities of the models for size estimation according to COSMIC Function Point are explained and examined prototypically. A comparison of a manual measurement with a trained and untrained model shows the current state of AI capabilities in a concrete example. Finally, the next steps within the study are briefly discussed.

Keywords

Artificial intelligence, size measurement, function points, cost estimation

1. Introduction

Embrace the power of Artificial Intelligence (AI) in software development cost estimation and revolutionize the way you plan your projects. Traditional cost estimation methods often rely on manual effort and subjective judgment, leading to inconsistencies, inaccuracies, and delays. By integrating AI into your estimation process, you unlock a world of possibilities.

AI is able to bring objectivity, efficiency, and predictive capabilities to the table. Through advanced algorithms and machine learning techniques, AI analyzes historical data, project parameters, and industry trends to generate accurate cost estimates. It eliminates human biases and enhances decision-making by considering a multitude of factors simultaneously.

With AI, you can benefit from faster estimation cycles, saving valuable time and resources. Its ability to process vast amounts of data in seconds means you can generate reliable cost estimates in a fraction of the time it would take using traditional methods. This efficiency allows you to allocate your resources more effectively, optimize project planning, and make informed decisions swiftly.

Moreover, AI-powered cost estimation provides a proactive approach. By leveraging machine-learning algorithms, AI continuously learns and adapts to new data, refining its estimation accuracy over time. It identifies patterns, recognizes project risks, and provides valuable insights that help you mitigate challenges before they arise. This proactive approach empowers you to make strategic adjustments, improve cost control, and increase the likelihood of project success.

The aim is to determine the possibilities of AI for automating and optimizing the COSMIC FP method. For this purpose, we have formulated the following research questions, which are to be answered in the proposal:

1. How can the cost estimation process supported by AI?
2. Can the software size be accurately determined with AI and COSMIC FP?

2. State of the Art

This section briefly explains the most important terms in the context of the subject area and presents current interesting studies and approaches.

2.1. Important terms

AI, short for Artificial Intelligence, refers to the simulation of human intelligence in machines that are programmed to think, learn, and perform tasks that typically require human intelligence. It is a branch of computer science that focuses on developing intelligent systems capable of replicating or emulating human cognitive abilities.

AI encompasses a wide range of techniques, algorithms, and approaches, including machine learning, deep learning, natural language processing, computer vision, expert systems, and more. These techniques enable AI systems to analyze and interpret complex data, recognize patterns, make decisions, solve problems, and even engage in natural language conversations.

Machine learning is a core component of AI and involves training models with large amounts of data to recognize patterns and make predictions or decisions. Deep learning, a subset of machine learning, utilizes artificial neural networks inspired by the structure of the human brain to process and learn from complex data.

AI systems can be categorized into two types: narrow AI and general AI. Narrow AI refers to systems designed to perform specific tasks within a defined domain. For example, image recognition, voice assistants, and recommendation systems are all examples of narrow AI applications. On the other hand, general AI aims to possess human-like intelligence across a broad range of tasks and exhibit characteristics such as reasoning, learning, and adaptability.

AI has widespread applications across various industries and domains. It is used in healthcare for diagnosing diseases and assisting in treatment plans, in finance for fraud detection and algorithmic trading, in transportation for autonomous vehicles, in customer service for chatbots, in manufacturing for process optimization, and in many other fields.

The field of AI continues to advance rapidly, with ongoing research and development exploring new techniques, algorithms, and applications. As AI systems become more sophisticated, they hold the potential to revolutionize industries, improve efficiency, and solve complex problems, making a significant impact on society as a whole.

Software Requirements are a description of the features and functionalities that a software system must have. They specify what the software should do and how it should perform.

There are several types of software requirements, including functional requirements, non-functional requirements, and domain requirements. Functional requirements describe the specific behaviors or functions of the software system. They specify what the system should do. Non-functional requirements describe the qualities or characteristics of the system. They specify how well the system should do what it does. Domain requirements are specific to the domain or industry in which the software will be used. They may include legal or regulatory requirements, industry standards, or other constraints that must be met by the software.

COSMIC FP (Function Point) is a software development methodology that measures the size of a system in terms of its functional requirements. It provides a standardized way to measure and compare the size of different systems, regardless of their technology or programming language. COSMIC FP is based on the concept of function points, which are used to quantify the functionality of a software system. The methodology involves analyzing the system's functional requirements and mapping them to specific function points, such as user inputs, outputs, and inquiries. By using this approach, organizations can better understand the size and complexity of their systems, identify areas for improvement, and make more informed decisions about software development projects. [1]

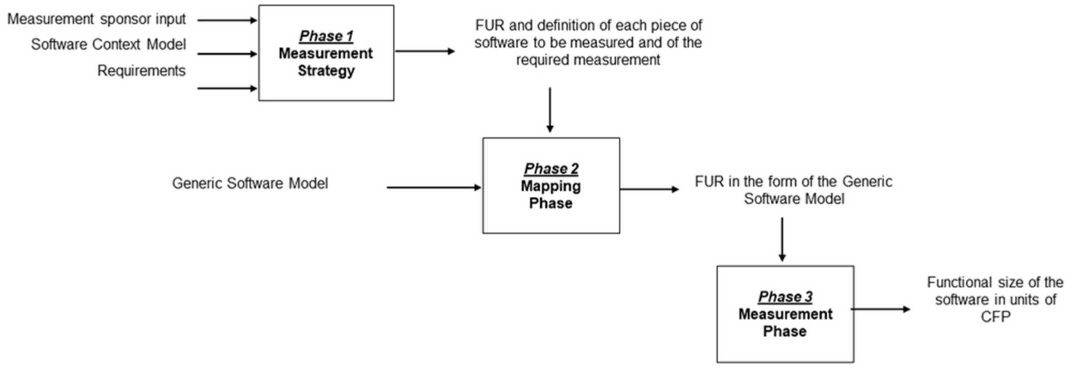

Measurement sponsor input
Software Context Model
Requirements
Phase 1
Measurement Strategy
FUR and definition of each piece of software to be measured and of the required measurement
Generic Software Model
Phase 2
Mapping Phase
FUR in the form of the Generic Software Model
Phase 3
Measurement Phase
Functional size of the software in units of CFP

Figure 1 Measurement process of COSMIC FP [2] S. 4

The measurement is carried out in three phases, which are shown in Figure 1. In the first phase, the functional user requirements, short FUR, are formed from the software context, requirements and measurement objectives. This is visualized in detail in Figure 2. In the second phase, these are mapped to the Generic Software Model, which is shown in Figure 3, so that the data movements become visible. In the third phase, the CFP of the individual data movements of the FUR are added together and result in the project size as CFP.

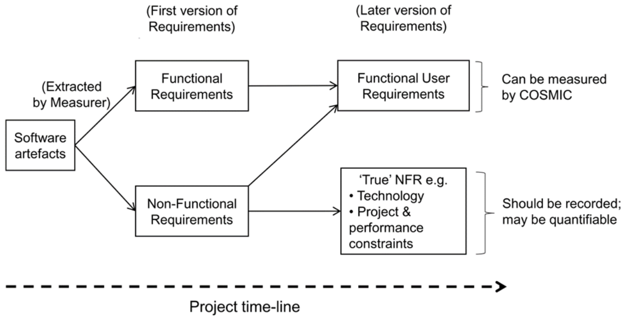

(First version of Requirements)
(Later version of Requirements)
(Extracted by Measurer)
Software artefacts
Functional Requirements
Functional User Requirements
Can be measured by COSMIC
Non-Functional Requirements
'True' NFR e.g.
• Technology
• Project & performance constraints
Should be recorded; may be quantifiable
Project time-line

Figure 2 Functional User Requirements [2] S. 7

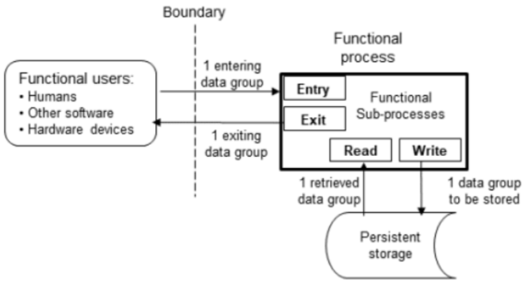


Figure 3 Generic Software Model [1] S. 2

2.2. Related Studies

The requirement to determine the cost of software projects quickly and precisely has existed for a long time, but for an accurate prediction many parameters have to be taken into account. With AI Supported Software Engineering it is now also possible to automate software design. Three current studies serve as a technical preselection of the approach:

- This study from 2020 uses ensemble learning bagging with base learner Linear regression, SMOReg, MLP, random forest, REPTree, and M5Rule to estimate the cost of software development. The dataset is based on 499 projects. The results show that the Mean Magnitude Relative error of Bagging M5 rule with Genetic Algorithm as Feature Selection is 10%, which makes it better than other algorithms. [3]

- This study from 2023 suggests a learning-based cost estimation model that leverages relational databases to improve accuracy. The proposed approach estimates project cost based on the effort required to complete software development, which is a key driver of the project cost. The proposed model is designed to address the challenges posed by the variability in open-source development, including variable team sizes, working hours and expertise. The proposed model is evaluated against 100 open source software repositories and shows its effectiveness in accurately estimating development costs. [4]
- This study from 2021 involves research about software effort estimation using machine learning algorithms. The objective of this research is to use several algorithms of machine learning to estimate the effort of software project development. The best machine learning model is chosen to compare with the Constructive Cost Model (COCOMO) which is one of the well-established software project estimation models but has some weaknesses, but still has some weaknesses, including a lack of accuracy according to software developers) [5]

3. Concept

Our approach is to estimate software size based on requirements catalogues and optimized AI models. The goal is the targeted support of Cosmic FP Analysis by AI for the identification of data movement types. Thus, phase 2 of the measurement process, shown in Figure 1, is to be automated. This information will then be used to carry out phase 3. Figure 4 illustrates the approach schematically.

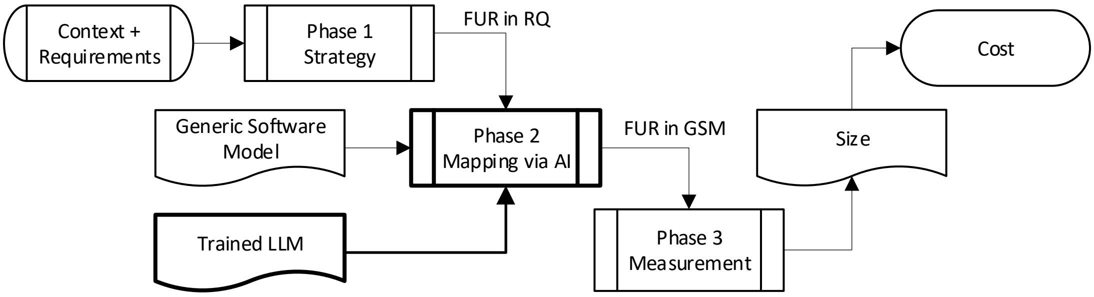


Figure 4 Concept of a fast cost Estimation with AI and COSMIC FP

With the size of the software product as CFP, established conversion factors for general types of software can be used to determine the anticipated costs. This conversion can be fine-tuned later using the software type and context. The conversion into person-months, or PM for short, is:

$$1 \text{ CFP} \approx 0{,}07 \text{ PM [6] S. 399}$$

3.1. Capabilities of AI

In addition to looking at the current literature on the subject, we conducted our own investigations and tests with potential AI technologies. The focus was on the applicability and quality of the results for the concept.

A **large language model** (LLM) is a language model consisting of a neural network with many parameters (typically billions of weights or more), trained on large quantities of unlabeled text using self-supervised learning or semi-supervised learning. LLMs emerged around 2018 and perform well at a wide variety of tasks. They are deep learning neural networks, a subset of artificial intelligence and machine learning. Large language models are first pre-trained so that they learn basic language tasks and functions. Pretraining is the step that requires massive computational power and cutting-edge hardware. LLMs can recognize, summarize, translate, predict and generate text and other content based on knowledge gained from massive datasets3. They are among the most successful applications of transformer models.[7]

In our consideration [8] of LLMs, specifically ChatGPT, in the context of software development, we have potential in terms of generating software architectures, shown at the appendix 6.1. We found shortcomings in quality and accuracy, but the LLM was not trained for this focus. For this reason, LLMs trained specifically for the task could greatly improve cost estimation.

Semantic analysis is the process of using natural language processing (NLP), text analysis, and computational linguistics to identify and extract subjective information from source materials. It is used to determine the emotional tone behind a series of words, used to gain an understanding of the attitudes, opinions, and emotions expressed within an online mention. Some popular use cases of sentiment analysis include social media monitoring, customer support management, analyzing customer feedback, brand monitoring and reputation management

In our consideration of sematic analysis of software system requirements in the context of software engineering, we have identified potential. Furthermore, it was found that the quality of the results is very much dependent on the training data set.[9]

3.2. Experiment

In order to determine the suitability of AI in measurement, a practical comparison was chosen as the method. This provides an initial picture of the performance and is manageable in terms of effort. Figure 5 shows the workflow of the approach. The stages are as follows:

1. Context + Requirements
2. Legacy Measurement
3. AI Measurement
4. LLM Training
5. AI Measurement (Trained LLM)
6. Comparison

The first step is to define the context, i.e. the type and goal of the software, and to determine detailed requirements. This was done in a concrete experiment with the support of Chatgpt. The key results of this step are listed in Appendix 6.2.

In the second step, the measurement is carried out manually to obtain reference values. The results are listed in Appendix 0.

In the third step, the requirements are mapped to COSMIC FP using LLM and measured. The outputs are listed in Appendix 6.4 and 0.

The fourth step is the training of the LLM. Here, the LLM is prepared with a few partial results of the manual CFP mapping using promting. This is documented in Appendix 0.

In the fifth step, the requirements are mapped to COSMIC FP using the trained LLM and measured. These results are also shown in Appendix.

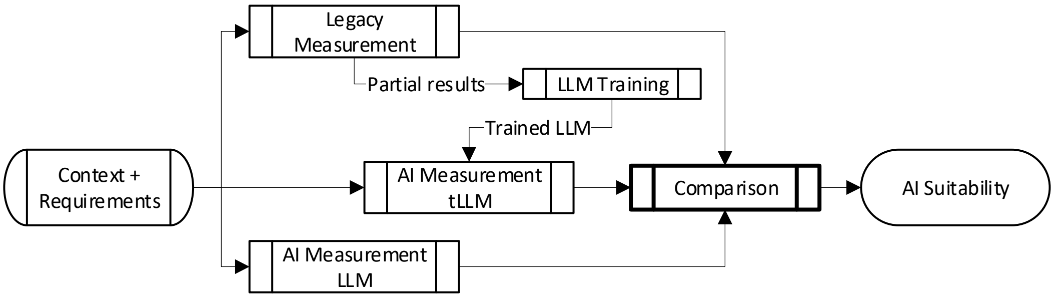


Figure 5 Process for determining the suitability of LLM for sizing software projects.

The criterion is the accuracy and the rough overall effort of the measurement.

3.3. Results

Table 1 compares the results of the individual measurements in order to identify deviations of individual mapping points. An important finding is the close examination of the issue of the LLM. In the experiment, the first run classified and scored according to IFPUG FP, although COSMIC FP was requested. Furthermore, the LLM had to be made aware of missing results, e.g. writes, and the aggregation of the individual partial results was difficult. This could be addressed by using the intermediate results.

Table 1 Comparison of the measurements

	Legacy	LLM	tLLM
User Authentication and Authorization:	6	5	6
Contact Management	11	5	10
Lead and Opportunity Management:	10	5	12
Sales and Pipeline Management:	10	6	9
Customer Interaction Tracking:	6	5	7
Task and Activity Management:	6	5	7
Communication Integration:	7	7	7
Reporting and Analytics:	9	5	9
Integration with Other Systems:	8	6	6
Mobile Accessibility:	8	6	3
CFP	81	55	76
estimated Cost (PM)	5,67	3,85	5,32

The overall result shows a good suitability of LLMs. This is especially relevant against the background of the high effort required for manual mapping and measurement.

4. Conclusion

The results of the experiment with show clear potential for the use of AI in the sizing and cost estimation of software development projects, especially in the requirements phase. Which is at the beginning of the software development cycle and therefore no software artefacts are available yet. The results also show that the quality depends very much on the intensity of the training.

The findings in relation to the research questions are:

1. How can the cost estimation process be supported by AI?
    a. The cost estimation process can be greatly accelerated by using large scale models. Specifically, activities related to function user requirements, CFP mapping and CFP measurement can be automated.
    b. Communication with LLM is challenging and not always goal-oriented, as methods sometimes have to be demanded emphatically.
2. Can the software size be accurately determined with AI and COSMIC FP?
    a. The accuracy of the size measurement depends very much on the input parameters; if these are pre-trained with the help of promting, the result is much more accurate.
    b. Training the LLMs via promting is also non-trivial, as, for example, inaccurate examples of the LLM are exacerbated in their subsequent executions.

For further research on this topic, we are focusing on training (promting), especially on the development of cross-project datasets. Furthermore, a permanent validation of the input parameters with the AI outputs is helpful to address misunderstandings and false statements at an early stage.

5. Reference

[1] COMMON SOFTWARE MEASUREMENT INTERNATIONAL CONSORTIUM: *WHAT IS A COSMIC FUNCTION POINT?* URL https://cosmic-sizing.org/wp-content/uploads/2018/08/What-is-a-COSMIC-Function-Point-v1.0-1.pdf

[2] ABRAN, Alain ; FAGG, Peter ; LESTHERHUIS, Arlan: *COSMIC Measurement Manual for ISO 19761* : *Part 2: Guidlines*

[3] ALHAZMI, Omar H. ; KHAN, Mohammed Zubair: *Software Effort Prediction Using Ensemble Learning Methods*. In: *Journal of Software Engineering and Applications* 13 (2020), Nr. 07, S. 143–160. URL https://www.scirp.org/pdf/jsea_2020071610583292.pdf

[4] NAGARAJAPPA, Ravi Kumar Begur ; SURESH, Yeresime: *Software Development Effort Estimation Using Relational Database and Optimized Learning Mechanism*. In: *Journal of Computer Science* 19 (2023), Nr. 4, S. 540–553. URL https://thescipub.com/pdf/jcssp.2023.540.553.pdf

[5] ZAKARIA, Noor Azura ; ISMAIL, Amelia Ritahani ; ALI, Afrujaan Yakath ; KHALID, Nur Hidayah Mohd ; ABIDIN, Nadzurah Zainal: *Software Project Estimation with Machine Learning*. In: *International Journal of Advanced Computer Science and Applications* 12 (2021), Nr. 6

[6] DUMKE, Reiner (Hrsg.); SCHMIETENDORF, Andreas (Hrsg.); SEUFERT, Manfred (Hrsg.); WILLE, Cornelius (Hrsg.): *Handbuch der Softwareumfangsmessung und Aufwandschätzung*. Berlin : Logos Verlag, 2014

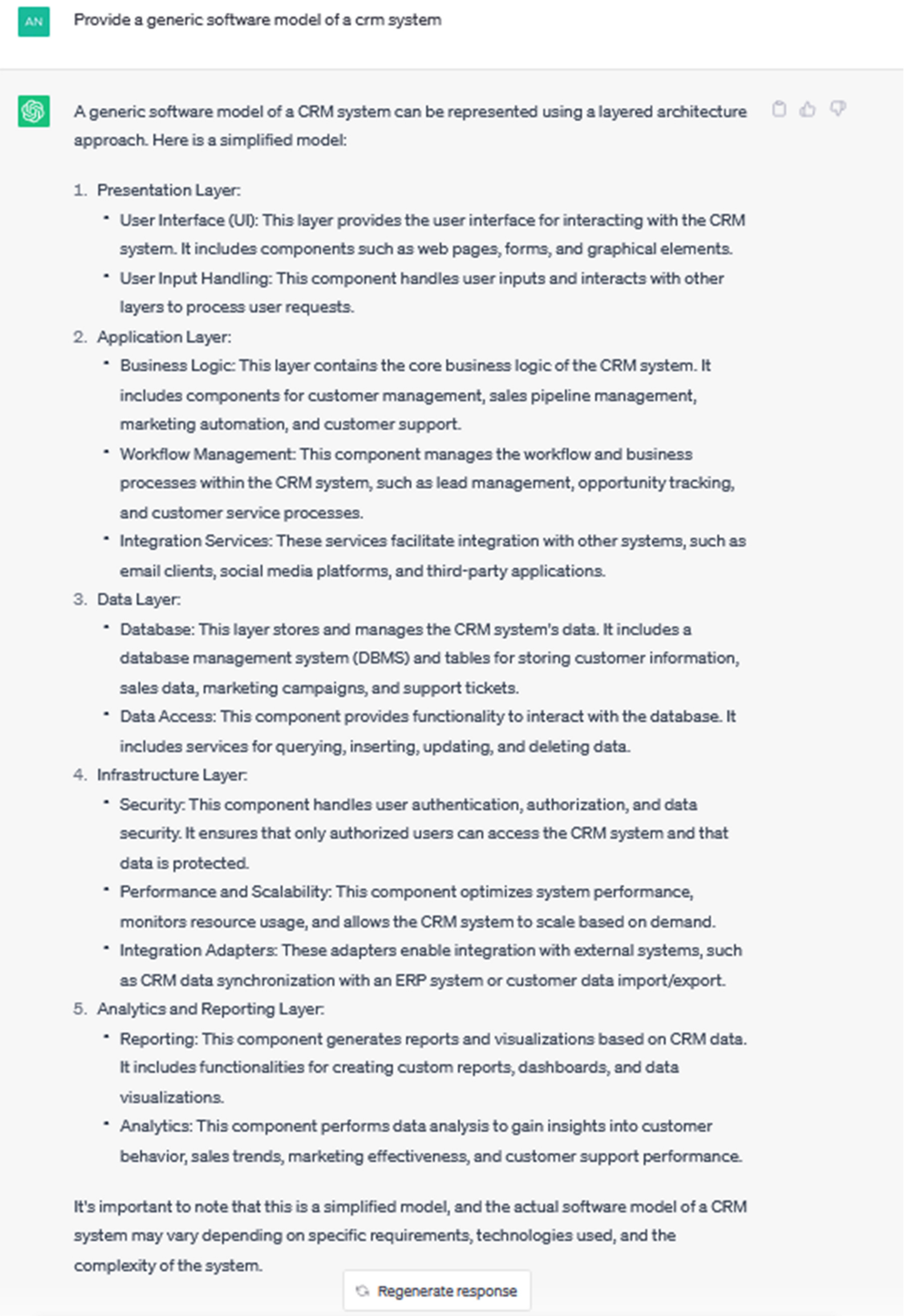
[7] YANG, Jingfeng ; JIN, Hongye ; TANG, Ruixiang ; HAN, Xiaotian ; FENG, Qizhang ; JIANG, Haoming ; YIN, Bing ; HU, Xia: *Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond*. URL https://arxiv.org/pdf/2304.13712.pdf

[8] HARTENSTEIN, Sandro ; SCHMIETENDORF, Andreas: KI basierte Veränderungen der Aufgaben innerhalb des Software Engineerings : Eine Bestandsaufnahme im Zusammenhang mit dem Hype um ChatGPT, Bd. 47. In: GI-Fachausschuß Management der Anwendungsentwicklung und -wartung (Hrsg.): *WI-MAW-Rundbrief* : *GI-Fachausschuß Management der Anwendungsentwicklung und -wartung*, 2023 (47), S. 3–13

[9] SCHMIETENDORF, Andreas ; HARTENSTEIN, Sandro ; JOHNSON, Sidney Leroy: KI-gestützte Modernisierung von Altanwendungen: (Sentiment-) Analysen im Diskurs des Anforderungsmanagements. In: GI-FG SRE (Hrsg.): *WSRE*, 2023 (25), S. 20–21

6. Appendix

6.1. Generic Software Model by ChatGPT



Provide a generic software model of a crm system

A generic software model of a CRM system can be represented using a layered architecture approach. Here is a simplified model:

1. Presentation Layer:
    - User Interface (UI): This layer provides the user interface for interacting with the CRM system. It includes components such as web pages, forms, and graphical elements.
    - User Input Handling: This component handles user inputs and interacts with other layers to process user requests.
2. Application Layer:
    - Business Logic: This layer contains the core business logic of the CRM system. It includes components for customer management, sales pipeline management, marketing automation, and customer support.
    - Workflow Management: This component manages the workflow and business processes within the CRM system, such as lead management, opportunity tracking, and customer service processes.
    - Integration Services: These services facilitate integration with other systems, such as email clients, social media platforms, and third-party applications.
3. Data Layer:
    - Database: This layer stores and manages the CRM system's data. It includes a database management system (DBMS) and tables for storing customer information, sales data, marketing campaigns, and support tickets.
    - Data Access: This component provides functionality to interact with the database. It includes services for querying, inserting, updating, and deleting data.
4. Infrastructure Layer:
    - Security: This component handles user authentication, authorization, and data security. It ensures that only authorized users can access the CRM system and that data is protected.
    - Performance and Scalability: This component optimizes system performance, monitors resource usage, and allows the CRM system to scale based on demand.
    - Integration Adapters: These adapters enable integration with external systems, such as CRM data synchronization with an ERP system or customer data import/export.
5. Analytics and Reporting Layer:
    - Reporting: This component generates reports and visualizations based on CRM data. It includes functionalities for creating custom reports, dashboards, and data visualizations.
    - Analytics: This component performs data analysis to gain insights into customer behavior, sales trends, marketing effectiveness, and customer support performance.

It's important to note that this is a simplified model, and the actual software model of a CRM system may vary depending on specific requirements, technologies used, and the complexity of the system.

Generated under the use of ChatGPT (https://chat.openai.com/)

6.2. Requirements by ChatGPT

The entire request history can be found under:

https://cloud.hwr-berlin.de/s/bmjWnHcBr3Fxsxb password: HYConMjzAj

Shorten results:

Question to ChatGPT:

Create a concrete software model of a CRM system for 1000 users and the main requirement of integrity, confidentiality and availability. It should include the typical requirements of a CRM system. Please provide all Functional User Requirements.

Answer from ChatGPT:

Sure! Here are some functional user requirements for a CRM system designed to cater to 1000 users with a focus on integrity, confidentiality, and availability:

1. User Authentication and Authorization:
   - The system should support secure user authentication and enforce strict access controls based on user roles and permissions.
   - Users should have unique credentials to access the system, and passwords should be securely stored using industry-standard encryption.
2. Contact Management:
   - Users should be able to add, view, edit, and delete contacts.
   - Contacts should include essential information such as name, email, phone number, address, and any additional custom fields.
   - The system should provide search and filtering capabilities to easily find and retrieve specific contacts.
3. Lead and Opportunity Management:
   - Users should be able to create and track leads and opportunities.
   - Each lead/opportunity should have relevant information such as source, status, value, expected closing date, and associated contacts.
   - Users should be able to assign leads/opportunities to specific users or teams for follow-up and collaboration.
   - The system should allow progress tracking, including updating the stage of each lead/opportunity.
4. Sales and Pipeline Management:
   - The system should provide functionality for managing the sales pipeline.
   - Users should be able to track the progress of deals, view sales targets, and analyze sales performance.
   - The system should generate reports and visualizations to provide insights into the sales process and identify potential bottlenecks.
5. Customer Interaction Tracking:
   - Users should be able to log and track all interactions with customers, including meetings, calls, emails, and notes.
   - The system should provide a centralized history of customer interactions for easy reference and follow-up.
6. Task and Activity Management:
   - Users should be able to create and manage tasks and activities related to contacts, leads, opportunities, and other CRM entities.
   - The system should support assigning tasks to users, setting due dates, and sending notifications/reminders.

7. Communication Integration:
    - The system should integrate with email clients to facilitate seamless email communication within the CRM.
    - Users should be able to send and receive emails directly from the CRM system and have them automatically linked to relevant contacts or leads.
8. Reporting and Analytics:
    - The system should offer comprehensive reporting capabilities, allowing users to generate predefined and custom reports.
    - Reports should cover various aspects, including sales performance, lead conversion rates, customer satisfaction, and team productivity.
    - The system should provide interactive dashboards and visualizations to analyze data effectively.
9. Integration with Other Systems:
    - The CRM system should support integration with other business applications such as marketing automation, customer support, or ERP systems.
    - Integration should enable seamless data sharing and synchronization between different systems.
10. Mobile Accessibility:
    - The CRM system should have a mobile-friendly interface or a dedicated mobile app to allow users to access and manage CRM data on the go.

Remember that these requirements are just a starting point, and you may need to further refine and tailor them based on the specific needs and preferences of your organization.

6.3. Legacy Measurement

Table 2 COSMIC FP Measurement for CRM

	Entries E	Exits X	Reads R	Writes W	Sum CFP	
User Authentication and Authorization:						6
Authentication	1	1	1	0	3	
Authorization	1	1	1	0	3	
Contact Management						11
User	1	1	1	1	4	
Contacts	1	1	1	1	4	
Filter	1	1	1	0	3	
Lead and Opportunity Management:						10
Leads	1	1	1	1	4	
Details	0	0	1	1	2	
Assigns	0	0	1	1	2	
Tracking	1	0	0	1	2	
Sales and Pipeline Management:						10
Sales	1	1	1	1	4	
Tracking	1	1	0	1	3	
Reports	1	1	1	0	3	
Customer Interaction Tracking:						6
Tracking	1	1	0	1	3	
History	1	1	1	0	3	
Task and Activity Management:						6
Tasks	1	1	1	1	4	
Assigns	0	0	1	1	2	
Communication Integration:						7
Email	1	1	1	1	4	
Links	1	1	1	0	3	
Reporting and Analytics:						9
Reports	1	1	1	0	3	
Productivity	1	1	1	0	3	
Dashboards	1	1	1	0	3	
Integration with Other Systems:						8
Integration	1	1	1	1	4	
Sharing	1	1	1	1	4	
Mobile Accessibility:						8
Mobile App	1	1	3	3	8	
Sum CFP					**81**	

6.4. FUR Mapping by ChatGPT

The entire request history can be found under:

https://cloud.hwr-berlin.de/s/bmjWnHcBr3Fxsxb password: HYConMjzAj

Shorten results:

Question to ChatGPT:

Please provide all Functional User Requirmentns and data movements (Entries, Exits, Reads, Writes) for a COSMIC FP analysis.

Answer form ChatGPT:

To perform a COSMIC (Counting Software Intensive Projects Using IFPUG Function Points) FP analysis, we need to define the functional user requirements and identify the data movements (entries, exits, reads, writes) associated with each requirement. Here are the functional user requirements and their corresponding data movements:

User Authentication and Authorization:
Data Movements:
Entries: User credentials (username, password)
Reads: User roles and permissions
Writes: N/A
Exits: Access granted or denied

Contact Management:
Data Movements:
Entries: New contact details (name, email, phone number, address, etc.)
Reads: Contact information for viewing or searching
Writes: Updated contact information
Exits: Confirmation of contact creation, retrieval, or update

Lead and Opportunity Management:
Data Movements:
Entries: New lead/opportunity details (source, status, value, etc.)
Reads: Lead/opportunity information for viewing or searching
Writes: Updated lead/opportunity information
Exits: Confirmation of lead/opportunity creation, retrieval, or update

Sales and Pipeline Management:
Data Movements:
Entries: Sales targets, progress updates
Reads: Sales performance data, pipeline status
Writes: N/A
Exits: Sales reports, pipeline analysis

Customer Interaction Tracking:
Data Movements:
Entries: Logged interactions (meetings, calls, emails, notes)
Reads: Interaction history for specific customers
Writes: N/A
Exits: Interaction details, summary reports

Task and Activity Management:
Data Movements:
Entries: New tasks or activities (description, due date, assigned user)
Reads: Task/activity details for viewing or searching
Writes: Updated task/activity information

Exits: Confirmation of task/activity creation, retrieval, or update

Communication Integration:
Data Movements:
Entries: Email content, recipients
Reads: Received emails, linked contacts/leads
Writes: Sent emails, linked contacts/leads
Exits: Email sending/receiving status, linked email history

Reporting and Analytics:
Data Movements:
Entries: Report parameters (filters, time periods, data sources)
Reads: Relevant data for generating reports
Writes: N/A
Exits: Generated reports, visualizations

Integration with Other Systems:
Data Movements:
Entries: Data to be shared or synchronized with other systems
Reads: Data from other integrated systems
Writes: Data updates to other integrated systems
Exits: Synchronization confirmation, shared data status

Mobile Accessibility:
Data Movements:
Entries: User login credentials, mobile device information
Reads: CRM data for viewing or manipulation
Writes: Updated CRM data
Exits: Confirmation of data access or updates via mobile devices

These are general data movements associated with each functional user requirement. It's important to note that the actual data movements may vary depending on the specific implementation and design of the CRM system.

6.5. CFP Measurement by ChatGPT

Entire request history: https://cloud.hwr-berlin.de/s/YraAoBnECefeayR password: pH65oAgyrW

Shorten results:

To estimate the Cosmic Function Points (FP) for the given CRM system functionalities, we will consider the data movements involved in each functionality. Cosmic FP is a measure of the functional size of a software system, considering the inputs, outputs, inquiries, files, and external interfaces. Each data movement category is assigned a weight, and we calculate the total unadjusted function points (UFP) by summing up the weights of the data movements.

Here is the estimation for each functionality:

1. User Authentication and Authorization:
   Data Movements:
   - Entries: 2 (user credentials)
   - Reads: 2 (user roles and permissions)
   - Writes: 0
   - Exits: 1 (access granted or denied)

   Total UFP for User Authentication and Authorization: 5

2. Contact Management:
   Data Movements:
   - Entries: 1 (new contact details)
   - Reads: 2 (contact information)
   - Writes: 1 (updated contact information)
   - Exits: 1 (confirmation of contact creation, retrieval, or update)

   Total UFP for Contact Management: 5

3. Lead and Opportunity Management:
   Data Movements:
   - Entries: 1 (new lead/opportunity details)
   - Reads: 2 (lead/opportunity information)
   - Writes: 1 (updated lead/opportunity information)
   - Exits: 1 (confirmation of lead/opportunity creation, retrieval, or update)

   Total UFP for Lead and Opportunity Management: 5

4. Sales and Pipeline Management:
   Data Movements:
   - Entries: 2 (sales targets, progress updates)
   - Reads: 2 (sales performance data, pipeline status)
   - Writes: 0
   - Exits: 2 (sales reports, pipeline analysis)

   Total UFP for Sales and Pipeline Management: 6

5. Customer Interaction Tracking:
   Data Movements:
   - Entries: 1 (logged interactions)
   - Reads: 2 (interaction history)
   - Writes: 0
   - Exits: 2 (interaction details, summary reports)

   Total UFP for Customer Interaction Tracking: 5

6. Task and Activity Management:
   Data Movements:
   - Entries: 1 (new tasks or activities)
   - Reads: 2 (task/activity details)
   - Writes: 1 (updated task/activity information)
   - Exits: 1 (confirmation of task/activity creation, retrieval, or update)

   Total UFP for Task and Activity Management: 5

7. Communication Integration:
   Data Movements:
   - Entries: 2 (email content, recipients)
   - Reads: 2 (received emails, linked contacts/leads)
   - Writes: 1 (sent emails, linked contacts/leads)
   - Exits: 2 (email sending/receiving status, linked email history)

   Total UFP for Communication Integration: 7

8. Reporting and Analytics:
   Data Movements:
   - Entries: 1 (report parameters)
   - Reads: 2 (relevant data for generating reports)
   - Writes: 0
   - Exits: 2 (generated reports, visualizations)

   Total UFP for Reporting and Analytics: 5

9. Integration with Other Systems:
   Data Movements:
   - Entries: 1 (data to be shared or synchronized with other systems)
   - Reads: 2 (data from other integrated systems)
   - Writes: 1 (data updates to other integrated systems)
   - Exits: 2 (synchronization confirmation, shared data status)

   Total UFP for Integration with Other Systems: 6

10. Mobile Accessibility:
    Data Movements:
    - Entries: 2 (user login credentials, mobile device information)
    - Reads: 2 (CRM data)
    - Writes: 1 (updated CRM data)
    - Exits: 1 (confirmation of data access or updates via mobile devices)

    Total UFP for Mobile Accessibility: 6

Now, to calculate the total Cosmic FP, we sum up the UFP for each functionality:

Total Cosmic FP = 5 + 5 + 5 + 6 + 5 + 5 + 7 + 5 + 6 + 6 = 55

Therefore, the estimated Cosmic Function Points (FP) for the given CRM system is 55.

6.6. FUR Mapping and CFP Measurement by trained ChatGPT

Entire request history: https://cloud.hwr-berlin.de/s/5N7jQzf2oD8MZYw password: rNYwtYWQoc

Grouped Results:

1. User Authentication and Authorization:
   Requirement 1: Entries: 1 Exits: 1 Reads: 1 Writes: 0 Sum: 3 CFP
   Requirement 2: Entries: 1 Exits: 1 Reads: 1 Writes: 0 Sum: 3 CFP
   Sum of Group 1: 6 CFP
2. Contact Management:
   Requirement 1: Entries: 1 Exits: 1 Reads: 1 Writes: 1 Sum: 4 CFP
   Requirement 2: Entries: 1 Exits: 1 Reads: 1 Writes: 0 Sum: 3 CFP
   Requirement 3: Entries: 1 Exits: 1 Reads: 1 Writes: 0 Sum: 3 CFP
   Sum of Group 2: 10 CFP
3. Lead and Opportunity Management:
   Requirement 1: Entries: 1 Exits: 1 Reads: 1 Writes: 0 Sum: 3 CFP
   Requirement 2: Entries: 1 Exits: 1 Reads: 1 Writes: 1 Sum: 4 CFP
   Requirement 3: Entries: 1 Exits: 1 Reads: 1 Writes: 0 Sum: 3 CFP
   Requirement 4: Entries: 0 Exits: 0 Reads: 1 Writes: 1 Sum: 2 CFP
   Sum of Group 3: 12 CFP
4. Sales and Pipeline Management:
   Requirement 1: Entries: 1 Exits: 1 Reads: 1 Writes: 0 Sum: 3 CFP
   Requirement 2: Entries: 1 Exits: 1 Reads: 1 Writes: 0 Sum: 3 CFP
   Requirement 3: Entries: 1 Exits: 1 Reads: 1 Writes: 0 Sum: 3 CFP
   Sum of Group 4: 9 CFP
5. Customer Interaction Tracking:
   Requirement 1: Entries: 1 Exits: 1 Reads: 1 Writes: 1 Sum: 4 CFP
   Requirement 2: Entries: 1 Exits: 1 Reads: 1 Writes: 0 Sum: 3 CFP
   Sum of Group 5: 7 CFP
6. Task and Activity Management:
   Requirement 1: Entries: 1 Exits: 1 Reads: 1 Writes: 1 Sum: 4 CFP
   Requirement 2: Entries: 1 Exits: 1 Reads: 1 Writes: 0 Sum: 3 CFP
   Sum of Group 6: 7 CFP
7. Communication Integration:
   Requirement 1: Entries: 1 Exits: 1 Reads: 1 Writes: 0 Sum: 3 CFP
   Requirement 2: Entries: 1 Exits: 1 Reads: 1 Writes: 1 Sum: 4 CFP
   Sum of Group 7: 7 CFP
8. Reporting and Analytics:
   Requirement 1: Entries: 1 Exits: 1 Reads: 1 Writes: 0 Sum: 3 CFP
   Requirement 2: Entries: 1 Exits: 1 Reads: 1 Writes: 0 Sum: 3 CFP
   Requirement 3: Entries: 1 Exits: 1 Reads: 1 Writes: 0 Sum: 3 CFP
   Sum of Group 8: 9 CFP
9. Integration with Other Systems:
   Requirement 1: Entries: 1 Exits: 1 Reads: 1 Writes: 0 Sum: 3 CFP
   Requirement 2: Entries: 1 Exits: 1 Reads: 1 Writes: 0 Sum: 3 CFP
   Sum of Group 9: 6 CFP
10. Mobile Accessibility:
    Requirement 1: Entries: 1 Exits: 1 Reads: 1 Writes: 0 Sum: 3 CFP
    Sum of Group 10: 3 CFP

Sum of all CFP: 78 CFP