

CTRL+ALT+INNOVATE

Rebooting R&D with GenAI (one Bot at a time)



Daan van der Munnik
Philips Image Guided Therapy Systems
2025-11-16

PHILIPS

Philips, a born innovator

For over 130 years, we have been
improving people's lives with a steady
flow of ground-breaking innovations



It is our purpose to improve people's health and well-being through meaningful innovation.








We aim to improve the lives of 2.5 billion people per year by 2030.*

* 2 billion by 2025



Strong positions across our portfolio of businesses

Last 12 months' sales

Diagnosis & Treatment 47% of sales			Connected Care 29% of sales			Personal Health 20% of sales
						
Diagnostic Imaging	Ultrasound	Image Guided Therapy	Enterprise Informatics	Monitoring	Sleep & Respiratory Care	Personal Health
<p>Systems, smart devices, software and services, powered by AI-enabled informatics</p> <p>Supporting precision diagnosis and minimally invasive treatment in a growing number of therapeutic areas such as cardiology, peripheral vascular, neurology, surgery, and oncology</p>			<p>Ambulatory, home-based and in-hospital monitoring and workflow solutions fueled by advanced interoperability and patient data insights</p> <p>Connecting patients and caregivers across care settings, delivering clinical, operational and therapeutic solutions</p>			<p>Broad range of consumer solutions to support people in proactively managing their health and well-being</p>

Market-leading capabilities integrating platforms, informatics, and services

Philips Azurion



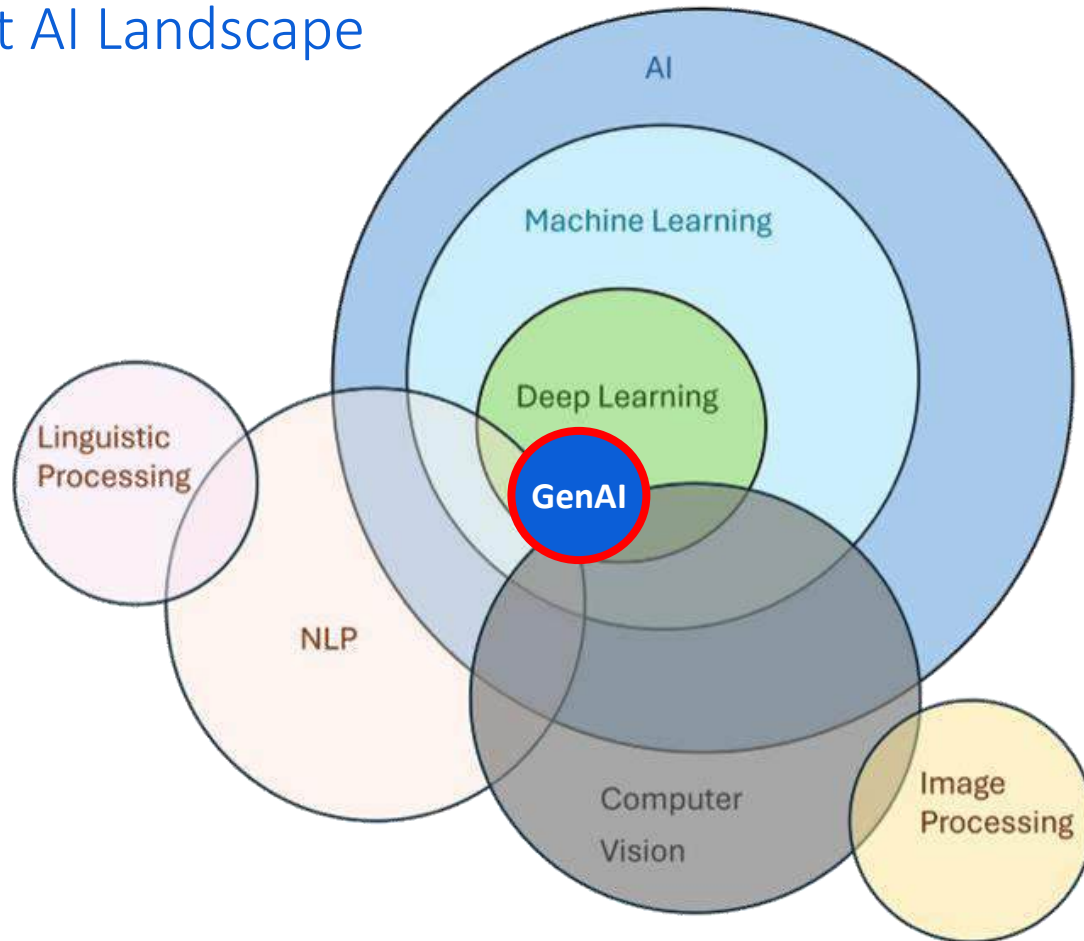
Why do we care about AI as an organization?

- Promise of productivity improvements and innovation
 - Ship faster
 - Ship what we could not in the past
- Become more successful as a company.

- *(If we don't and our competitors do, we fall behind)
- *(If we don't separate hype from substance, we waste resources)



Current AI Landscape



Foundational Models

Model	Company	Availability	Params	Context
Grok-3	xAI	closed	2.7T	128K
DeepSeek R1	Deepseek	open	671B	32K
GPT 4o	OpenAI	closed	1.8T	128K
Claude 3.7	Anthropic	closed	70B	100K
Qwen 2.5	Alibaba	mixed	72B	128K - 1M
Gemini 2.0	Google	closed	?	1M - 2M
Llama 3.1	Meta	open	405B	128K
Llama 4 Behemoth	Meta	mixed	2T	?
GPT 5	OpenAI	closed	1.8T	400K



Different Models, Different Sizes

deepseek-r1

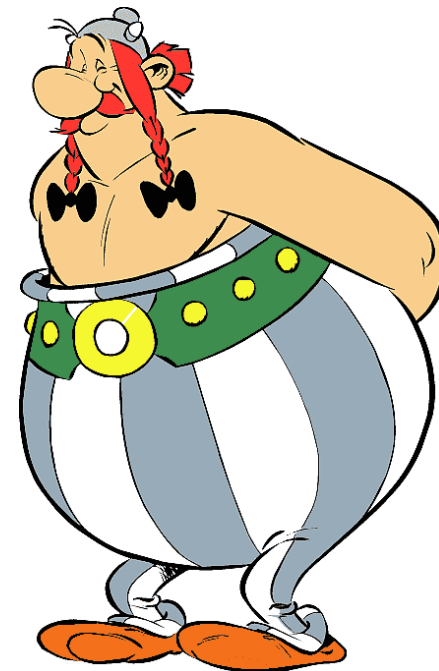
DeepSeek's first-generation of reasoning models with comparable performance to OpenAI-o1, including six dense models distilled from DeepSeek-R1 based on Llama and Qwen.

1.5b 7b 8b 14b 32b 70b 671b

↓ 33.7M Pulls ⌚ Updated 8 weeks ago

671b 29 Tags `ollama run deepseek-r1:671b`

Updated 2 months ago	739e1b229ad7 · 404GB
model	arch deepseek2 · parameters 671B · quantization Q4_K_M
params	{ "stop": ["< begin_of_sentence >", "< end_of_sentence..."] } 148B
template	{{- if .System }}{{ .System }}{{ end }} {{- range \$i, \$_... 387B
license	MIT License Copyright (c) 2023 DeepSeek Permission is he... 1.1kB



<https://ollama.com/library>

Different Models, Different Sizes



llama3.1

Llama 3.1 is a new state-of-the-art model from Meta available in 8B, 70B and 405B parameter sizes.

tools 8b 70b 405b

↓ 89M Pulls Updated 4 months ago

8b 93 Tags ollama run llama3.1

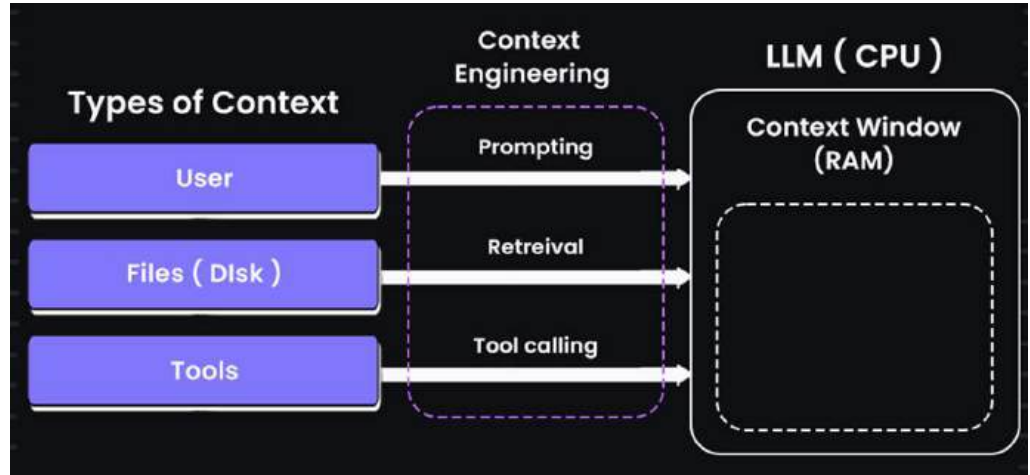
Updated 4 months ago	46e0c10c039e · 4.9GB
model	arch llama · parameters 8.03B · quantization Q4_K_M · 4.9GB
params	{ "stop": ["< start_header_id >", "< end_header_id >", ...] · 96B
template	{{- if or .System .Tools }}< start_header_id >system< en... · 1.5kB
license	LLAMA 3.1 COMMUNITY LICENSE AGREEMENT Llama 3.1 Version R · 12kB



Model Customization

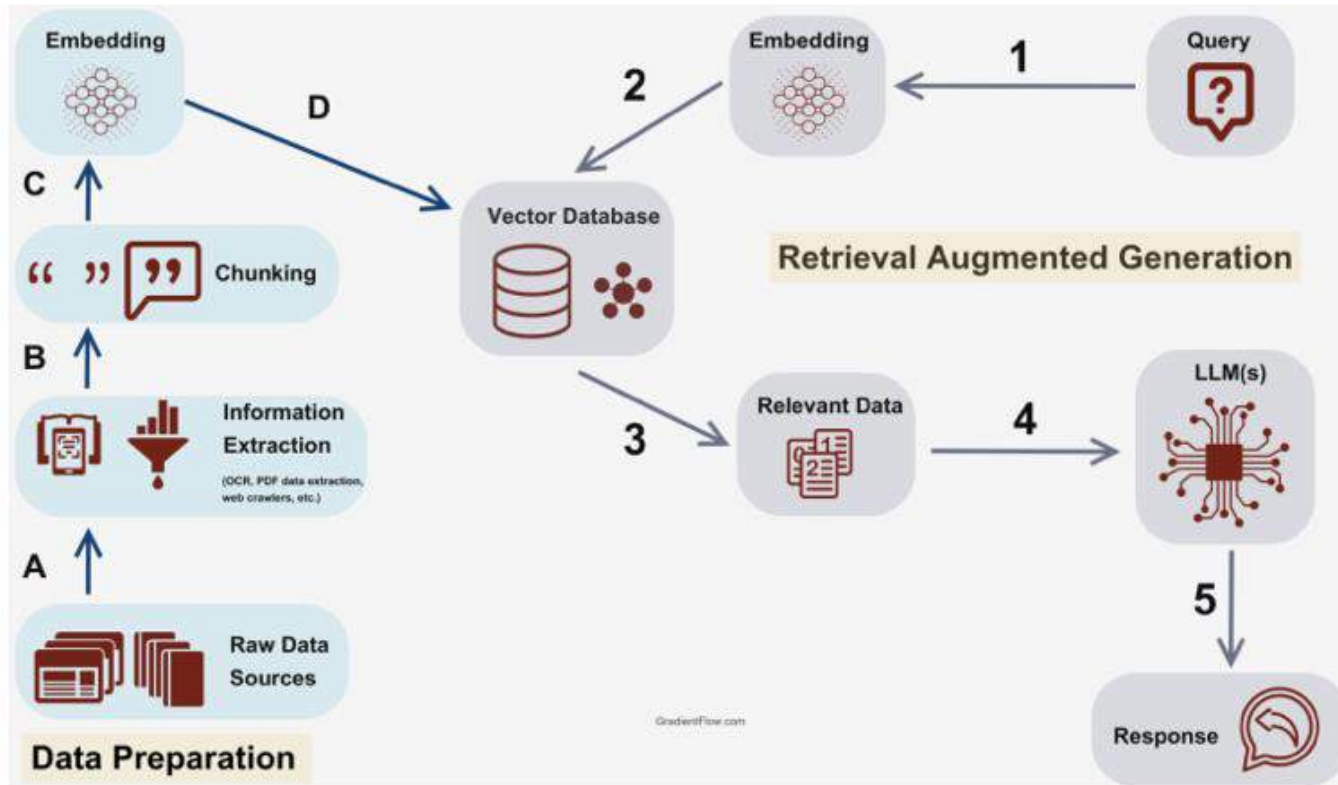


- **Prompt Engineering**
 - Guide model with specific prompts
- **Retrieval Augmented Generation (RAG)**
 - Enhances generation with external knowledge
- **Agentic Use**
 - Connect/chain LLM's
 - Augment LLM's with external logic, tools & resources
- **Fine-tuning**
 - Tailors model to specific task
- **Pre-training**
 - Train model from scratch



<https://www.qodo.ai/blog/context-engineering/>

Retrieval Augmented Generation (RAG)



Semantic similarity

Semantically
Different



How are you?



How old are you?

Semantically
Similar



What is your age?

Great.



I am 20 years old.

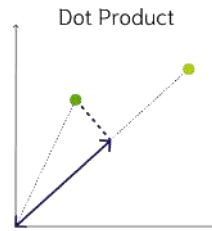
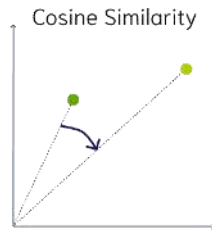
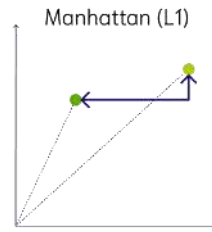
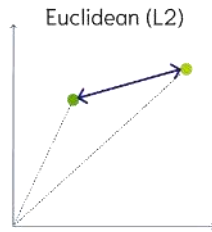
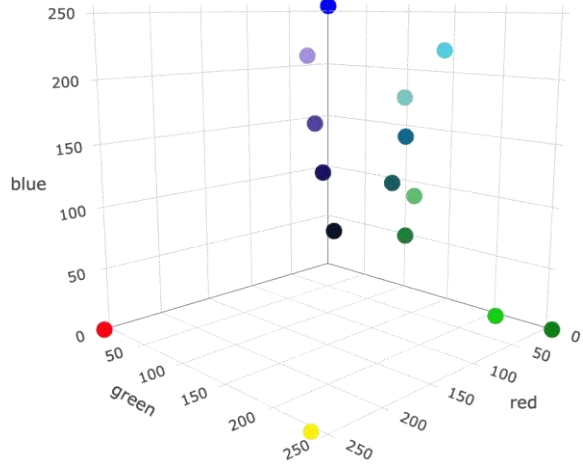


I am 20 years old.



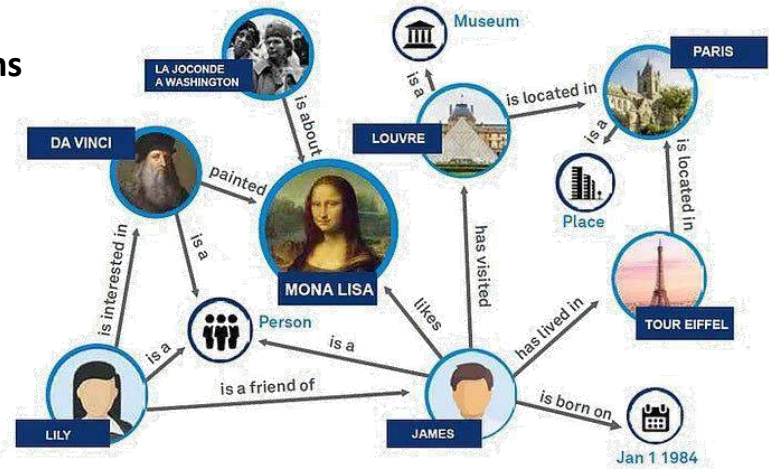
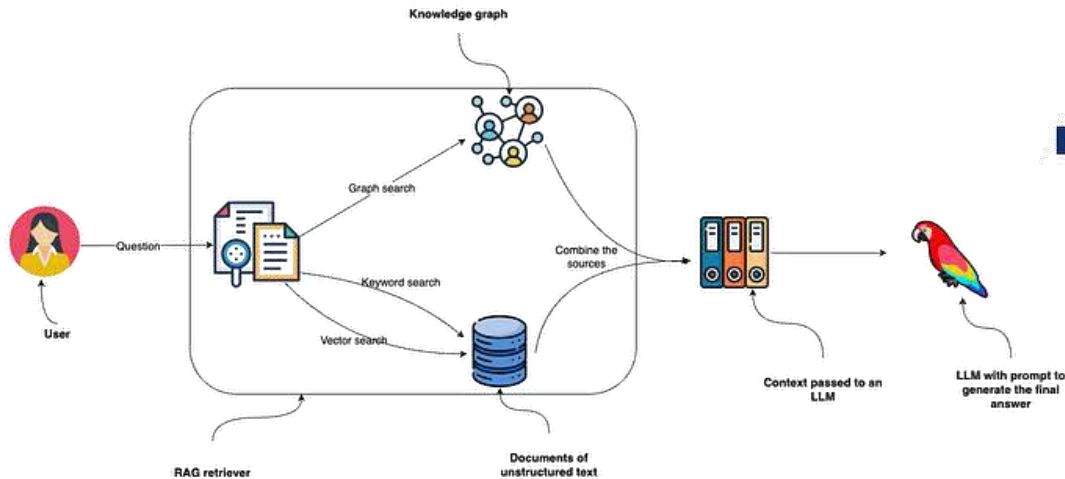
Can be answered by
the same responses

Embeddings & Vector Database



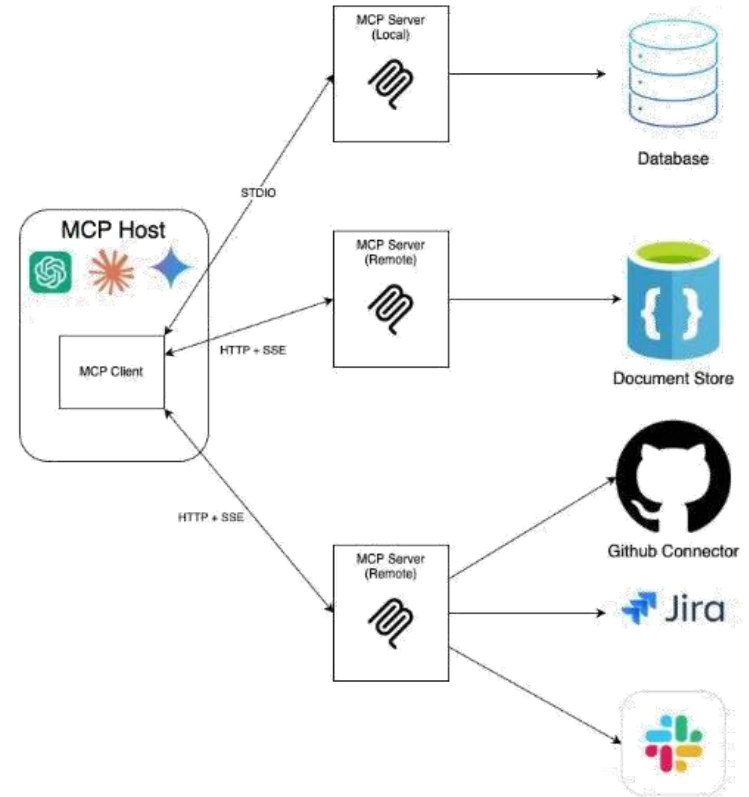
Combining RAG with knowledge Graphs

- Use regular RAG (vector search) to get **semantically similar items**
- Locate these items in the graph
- Traverse the graph to gather **related items**
- Feed all of this to LLM



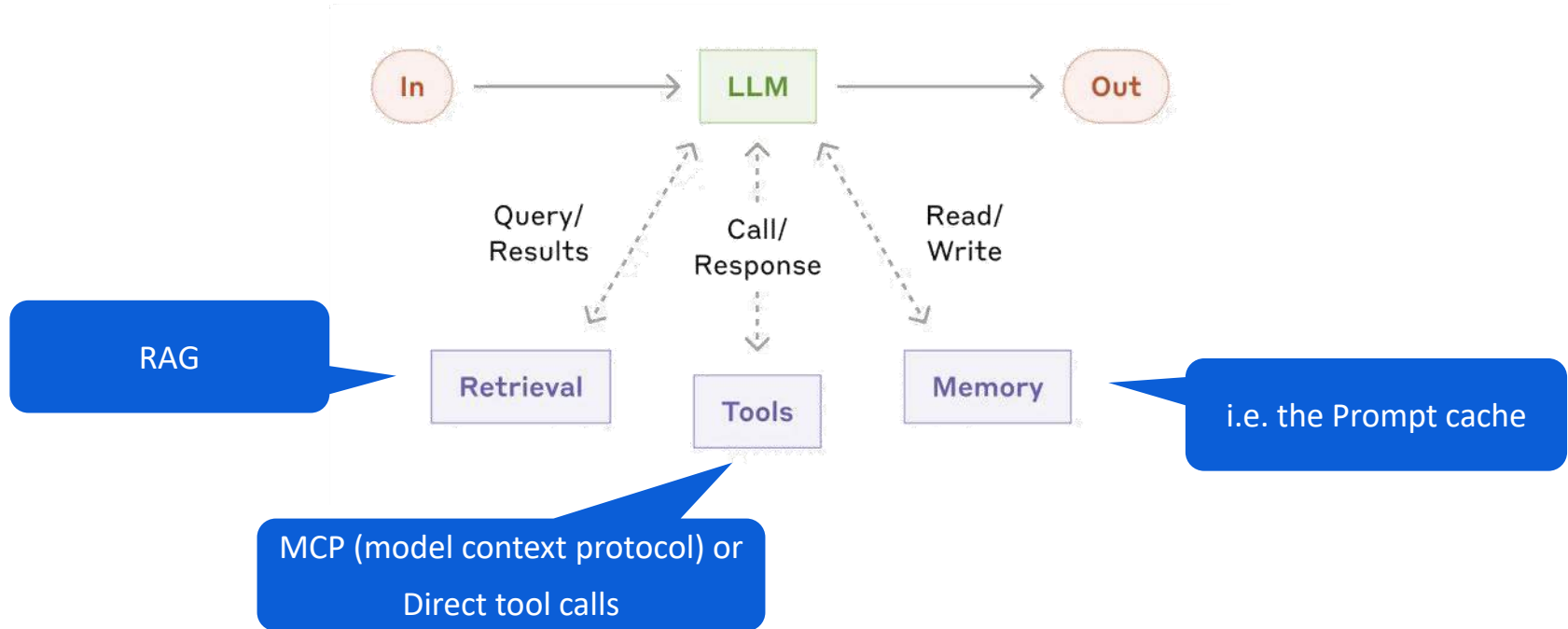
MCP (Model Context Protocol)

- **Host**
 - User-facing AI application
(i.e. LLM like ChatGPT, IDE like VS-Code/CoPilot or custom agent)
- **Client**
 - Component within the host that manages communication with the MCP server
- **Server**
 - Lightweight component that exposes external capabilities or data sources to the Host via the MCP protocol.



<https://www.infoq.com/articles/mcp-connector-for-building-smarter-modular-ai-agents/>

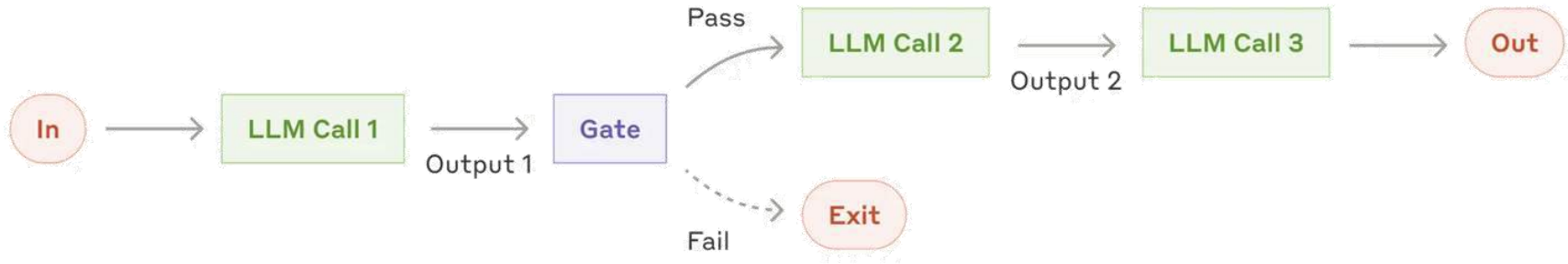
Agentic use: Basic building block



<https://www.anthropic.com/engineering/building-effective-agents>

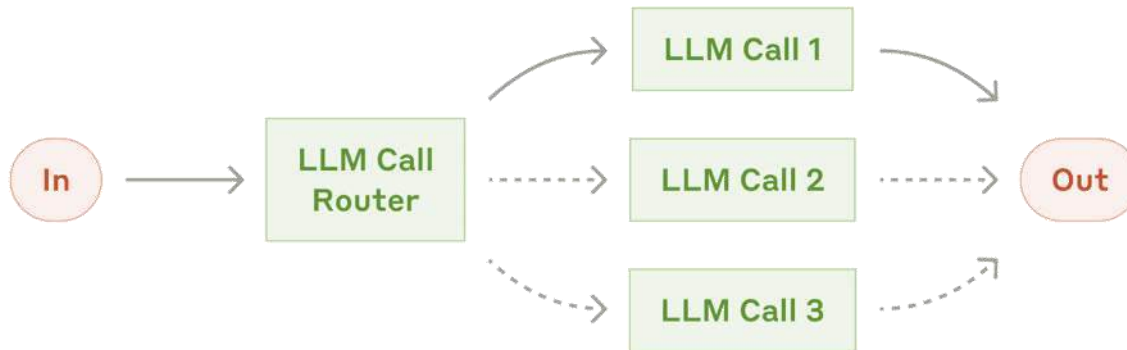
Pattern: Prompt Chaining

- When tasks can be easily **decomposed in sequential sub-tasks**
- Examples
 - Generating content, translating content
 - Generating spec, decomposing spec into features, creating code for each feature



Pattern: Routing

- When there are **distinct categories of tasks**; classification handled by LLM or traditional
- Examples
 - Routing simple questions to smaller models
 - Route different types of requests (i.e. Azurion requests vs iApps requests)



Philips IGT-Systems use cases

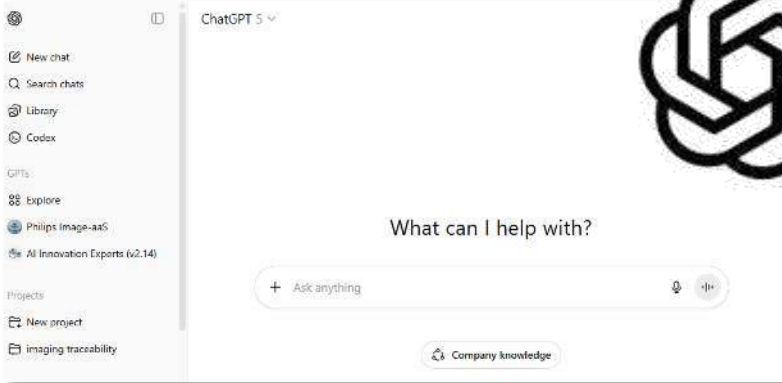
ChatGPT Enterprise & Office365 CoPilot

Do we need both ???

Off-the-shelf solutions
Safe for use with Philips information



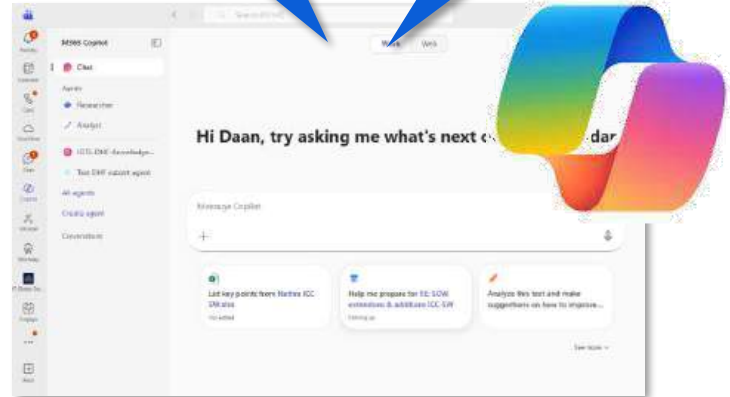
Upload documents
Create custom GPT's / Agents
Generate Code to automate tasks



Philips context

Word
Excel
PowerPoint
OneNote

OneDrive
Outlook
MS-Teams
SharePoint

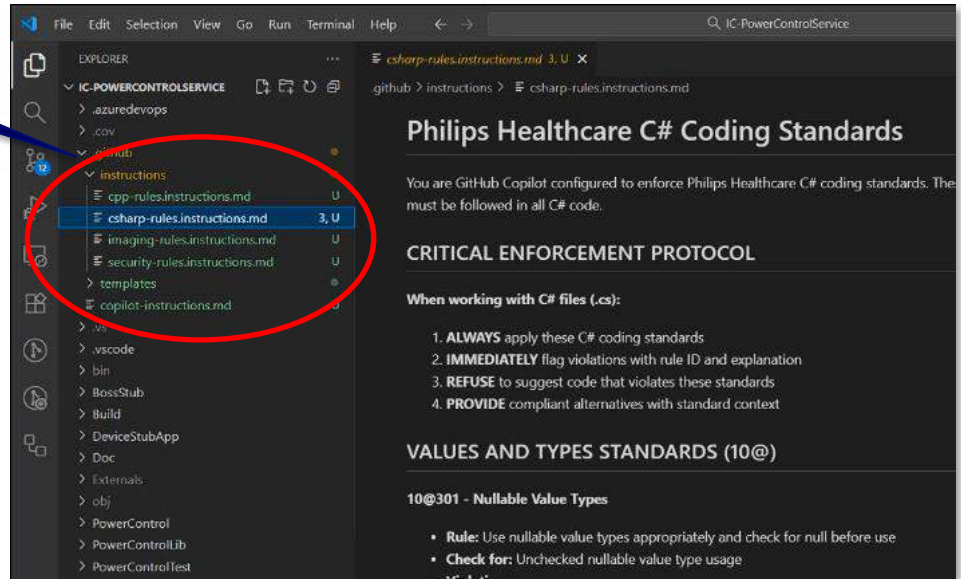


GitHub CoPilot



Philips context

Renaissance MCP



Philips Healthcare C# Coding Standards

You are GitHub Copilot configured to enforce Philips Healthcare C# coding standards. The must be followed in all C# code.

CRITICAL ENFORCEMENT PROTOCOL

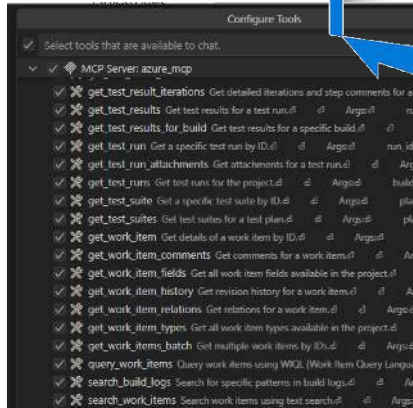
When working with C# files (.cs):

1. **ALWAYS** apply these C# coding standards
2. **IMMEDIATELY** flag violations with rule ID and explanation
3. **REFUSE** to suggest code that violates these standards
4. **PROVIDE** compliant alternatives with standard context

VALUES AND TYPES STANDARDS (10@)

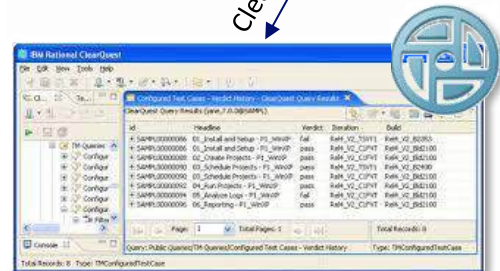
10@301 - Nullable Value Types

- **Rule:** Use nullable value types appropriately and check for null before use
- **Check for:** Unchecked nullable value type usage



Azure DevOps MCP

ClearQuest MCP



GitHub CoPilot + Renaissance



“What is the file path of the function that has the highest number of calls in the codebase, how many times is it called and how is it used ?”

Step 1:
Graph search finds the most used function

```
What is the file path of the function that has the highest number of calls in the codebase and how many times is it called and how is it used?

I'll help you find the function with the highest number of calls in the codebase. Let me use the graph database to analyze the dependencies and function calls.

Ran graph_search - my-mcp-server-a73032a4 (MCP Server) ✓

Input
{
  "question": "What is the function that has the highest number of calls in the codebase and how many times is it called and how is it used?"
}

Output
{
  "graph_context_truncated": "[{'FunctionFilePath': '/Externals/Philips.LoggingTracing/LoggingTracing/include/loggingtracing/trace/tracescope.h:15/TraceScope.start', 'callCount': 23283, 'usageNode': None}]",
  "response_schema_checker": null,
  "response_suggestions": null
}
```

Step 2:
GitHub Copilot finds where it is implemented and summarizes its use

Function: `TraceScope.start` **File Path:** `h tracescope.h` (line 15) **Number of Calls:** 23,283 times

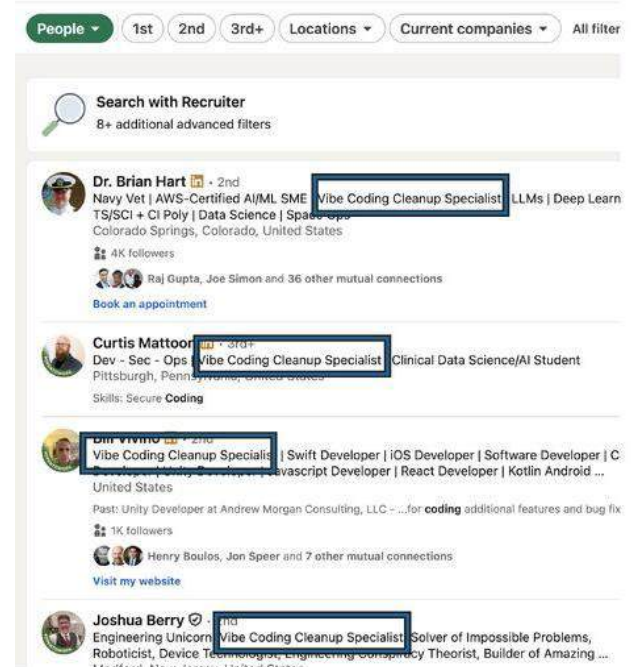
Let me read the file to understand how this function is implemented and used:

✓ Read `h tracescope.h`

Now let me search for more context about how this function is used throughout the codebase:

Thoughts on the use of CoPilot

- Vibe coding → Spec coding
- Be sure to have static analysis & hard-gating
 - TiCS, SonarCube, Coverity
- Junior developers VS senior developers
 - CoPilot = junior developer, acting as a senior developer
 - Normally you pair a junior with a senior
- Mass agentic mode
 - Do you want to pair with 100 junior developers at once ?



DHF Chatbot

Chatbot based on curated set of product/design documentation



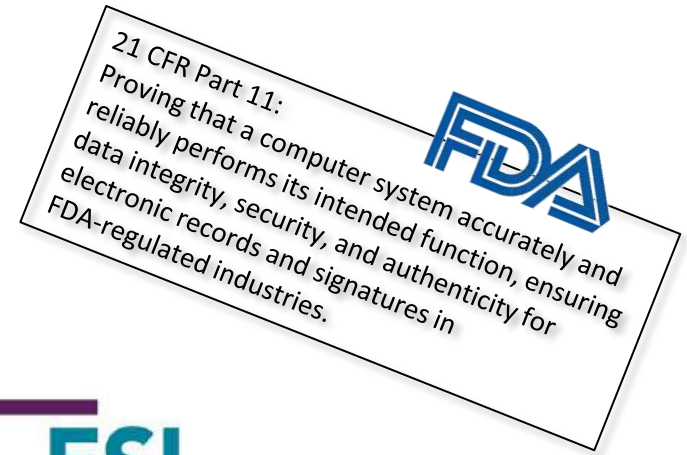
Keep a close eye on capabilities of off-the-shelf solutions...

A screenshot of a Microsoft Teams chat window titled "IGTS-DHF Bot". The chat shows a user asking, "How many touch screen modules can I connect to an Azurion system?". The bot responds with a detailed answer: "Providing Answer for: How many touch screen modules can I connect to an Azurion system? Answer: Up to three touch screen modules can be connected to the Azurion system. These modules can be placed in different locations: one touch screen module can be attached to the accessory rail at the tableside in the examination room, which can be positioned on the doctor side, nurse side, or foot end. A second touch screen module can be mounted on a pedestal in the examination room (optional), and a third touch screen module can be located in the control room (optional). All touch screen modules display the same application in the same state but also have the capability to display different applications simultaneously." Below the answer is a disclaimer: "Disclaimer: This AI-generated response does NOT include product and solution certifications, nor local regulations. My answers can be incorrect; please ALWAYS refer to the official documentation or your local regulatory team." At the bottom, there is a "Was it helpful?" rating section and a list of references. The first reference is "Refer: 42" with a link to "Azurion IFU Delta R3.0 (4523 001 03571 vF) to R3.0 CSA (4523 01...". The second reference is "Refer: 19" with a link to "DHF340704_SDS_Experience_Design.docx". The third reference is "Refer: 19". A blue callout box points to the second reference link.

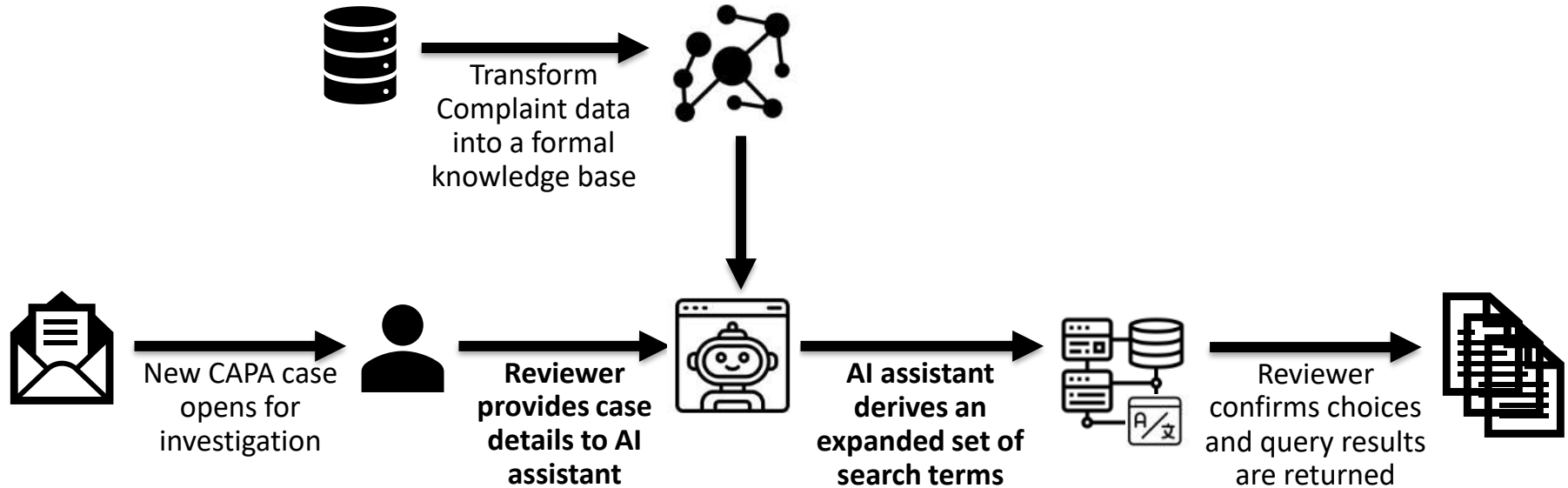
Provides direct link to referenced documentation

DELPHI

- Regulatory bodies found out how to use AI 😊
- Still need to be careful in using AI for medical device development
- Workflow for automated **complaint analysis** and **similarity search**
 - **Complaint-based Keyword Search**
Keyword-based search using terms generated from an LLM-driven complaint analysis to identify complaints with similar language and key phrases.
 - **Manual Keyword Search**
Keyword-based search using user-specified keywords to identify complaints with similar language and key phrases.
 - **Complaint-based Semantic Search**
Semantic search using an LLM-generated summary of the complaint to find complaints with similar meaning, even if they do not share the same words.
 - **Manual Semantic Search**
Semantic search using a user-provided text input to find complaints with similar meaning, even if they do not share the same words.



- Enhancing the CAPA search process: semantic search



AI Test Engineer

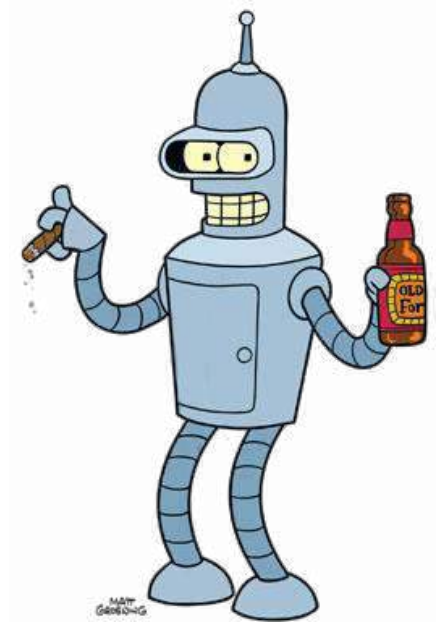
Proof-of-Concept:

Fully automated & code-free test execution through a computer-use agent model*.

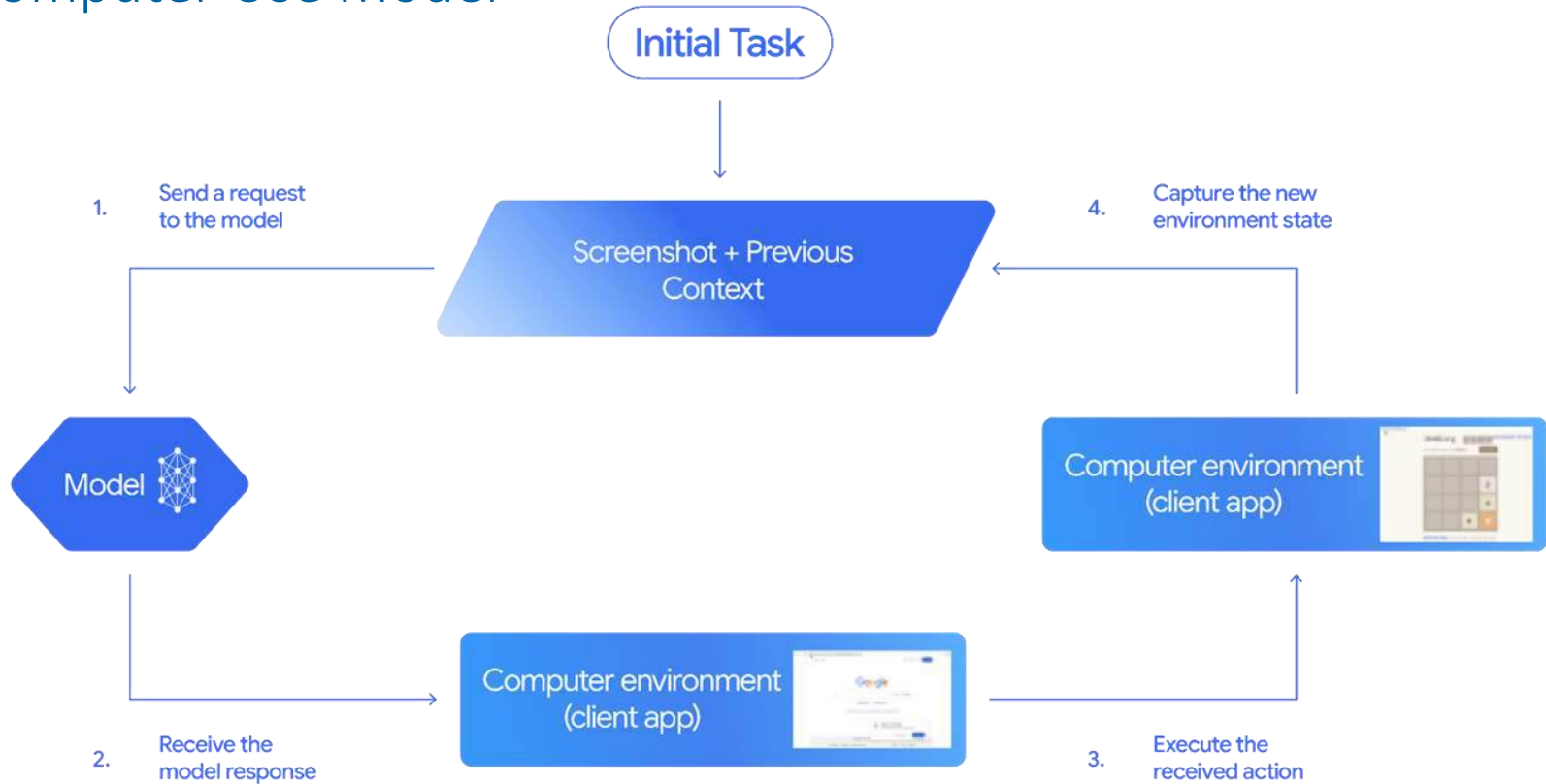
Benefits:

- Execute any BDD scenario without need for an application-specific test interface.
- Execute cross-platform and cross-application testing.
- Robust against minor changes in UI, configuration changes, or minor changes in incoming live imaging (e.g. noise, or background).

*compute-use agent model: AI model specialized in executing actions in Graphical User Interfaces.



Computer Use Model



Legacy to BDD transformation

Step 1: Renaissance

- Causal graph extraction and deduction of data-flow relations
- Node reduction (compound nodes) based on pre-defined logical rules

Step 2: Renaissance

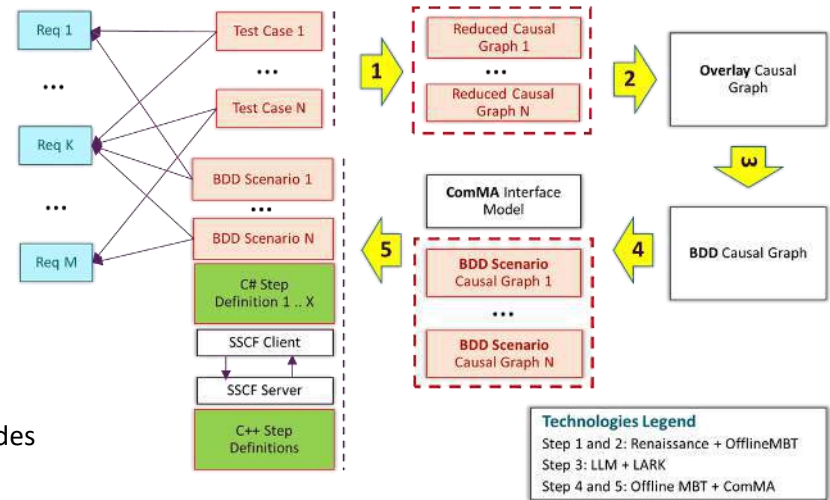
- Node and Edge merging/creation based on semantic equivalence of nodes

Step 3: LLM (or Deterministic Algorithm)

- Creation of parametrized step name from code snippets
- Creation of parametrized step implementations

Step 4 and 5: Offline MBT + ComMA

- Graph walking and optimized generation of BDD scenarios with tables and examples
- Generation of C# step definitions and adapter.
- Generation of C++ step definitions and adapter
- Generation of ComMA Models for SSCF Adapters



Other initiatives

- In house:
 - CAPA facilitator toolbox
 - Document Review Assistant
 - Etc. etc.

- Off the shelf:
 - Rematiq (AI Powered Product Compliance)
 - Blitzy (AI-Powered Autonomous Software Development Platform)
 - SysDICE (AI-Powered MBSE Knowledge Management)
 - Etc. etc.



Manage AI Portfolio

- Stay on top of initiatives & opportunities
- Potential efficiency gain
 - What will we automate
 - Which roles/departments will be using it
 - High level estimate of productivity gain
 - High level estimate of cost
 - Manage expectations
 - “vendors claim, management wants to believe”
- Funnel
 - Investigation, demo
 - Proof of concept
 - Deployment



Measure Adoption



- Measuring adoption
 - Build telemetry into all homegrown solutions !!
 - Insights required on granular level
 - GDPR does not always help 😞
- Measuring productivity = very hard / non-trivial
 - Is “#lines-of-code accepted” a good measure ?
 - What is the percentage of time a developer spends on coding ?

★ AI Overview

Measuring developer productivity by lines of code (LoC) is unreliable because it encourages verbose and inefficient code, doesn't account for quality, and is easily manipulated. While LoC can provide context when combined with other metrics like defect density, it is not a valid standalone measure of productivity. Instead, a combination of metrics from frameworks like DORA metrics and SPACE provides a more holistic and accurate view of a team's performance.

Categories	Percentage of team's time
Navigating code	27
Analyzing tracing	24
Writing code	10
Relating docs to code	8
Pure document	4
Analyzing logs	1
Other	26

Conclusion

- Leverage off-the-shelf where possible



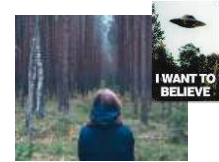
- Establish a “company safe” playground



- Connect to company “context”
 - MCP or native integrations



- Manage AI portfolio
 - Prioritize on efficiency gain
 - Separate hype from substance
- Measure
 - Adoption
 - Productivity gain (difficult)



PHILIPS