Quidgest
Artificial Intelligence Engineering

AUTOMATIC CODE GENERATION PLATFORMS AT TWO VERY DIFFERENT LEVELS

Genio vs Low-code:
Function Points/Month comparison

Full web/mobile solutions

Quidgest's GENIO is the world's #1 AI-powered model-driven automatic code generation platform, and this is the bottom line.

This graphic shows that a human typing code will only be able to achieve 16 function points per month. This is why typical software solutions are created by a large number of developers during a very long time.

Having Genio, Quidgest's AI powered model-driven automatic code generation platform that outputs 100 times the individual productivity, means that solutions are finished in 1/10 of the time and only using 1/10 of the usual resources.

As for the corporate/government benefits of having its software solutions deployed at this level of productivity, where do your dreams lie?

SOLUTION CREATION PRODUCTIVITY

(Function Points/Month)
ISO 24570:2018

Genio
1330
function points/ month

LOW CODE
175
function points/ month

MANUAL
16
function points/ month
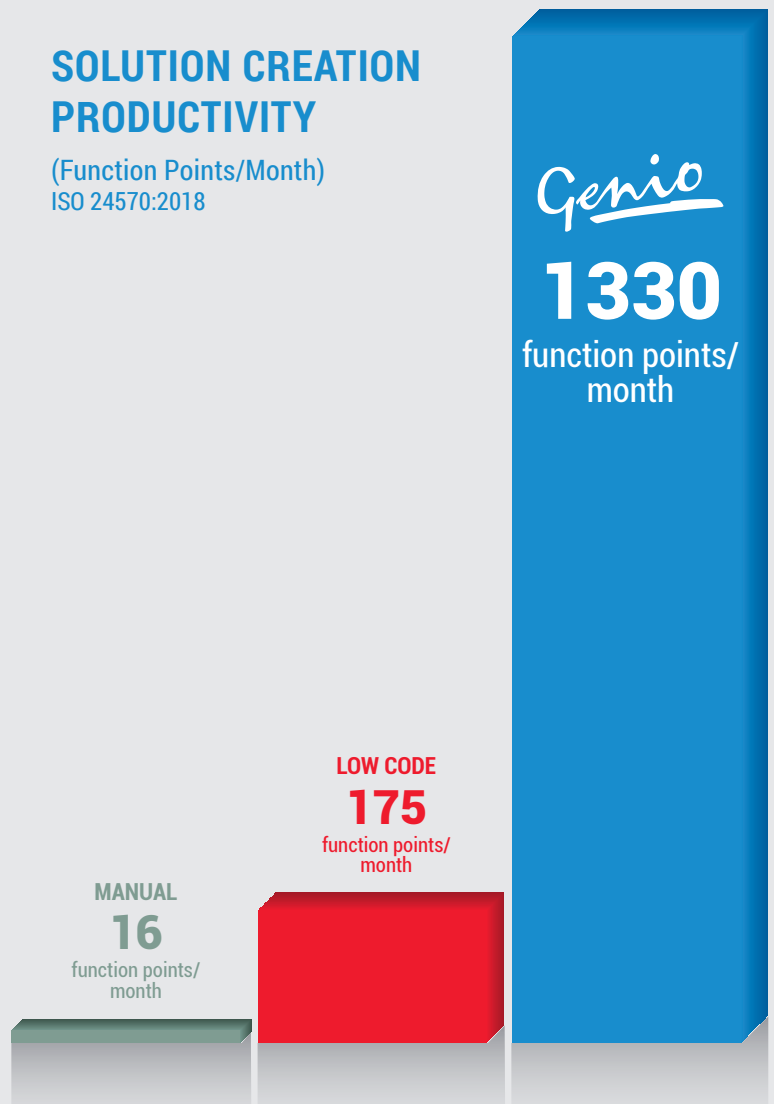
Function Points (FP) and Productivity

Function Point Analysis (FPA) was introduced long ago, back in the late 1970's, by A.J. Albrecht from IBM in an attempt to define the concept of "productivity" in a large number of projects. Over the following years a number of different "dialects" were introduced into FPA and this created great difficulty in comparing figures due to the different concepts of determining the number of function points. The International Organization for Standardization (ISO) has stepped in and produced a number of norms related to FPA, being the latest the ISO 24570:2018. Nesma, an organization deeply involved in software measurement, has produced a manual, based on the recent norm, to be used for the required calculations.

In what concerns productivity, it is associated with the division of function points by a period of time. An expert on the field, and founder of a company providing low-code platforms, states: "productivity metrics are based on dividing the functional size for the application as measured in FPs by the normalized work effort for the time period to deliver. This measure is based on the Normalised Level 1 Productivity Delivery Rate from [ISBSG 10]".

Function points measure the size of an application system based on the functional view of the system. The size is determined by counting the number of inputs, outputs, queries, internal files and external files in the system and adjusting that total for the functional complexity of the system. Function point analysis, originally developed at IBM, has as an advantage its focus on measuring software produced in terms of functionality delivered to the end user, rather than in terms of development deliverables, which have no direct bearing on the end user.

in Gartner IT Glossary

Genio

CODE GENERATION PRODUCTIVITY

100 times
individual productivity

1/10 of time

1/10 of usual resources

Function Types & Complexity

There are five types of functions which are involved in an software solution concerning a perspective relevant to FPA:

External Inputs
External Outputs
External Inquiries
→ **Transactional Functions** → **TYPES OF FUNCTIONS** ← **Data Functions** ←
Internal Logical Files
External Interface Files

Each of these functions have a larger or smaller degree of complexity, being complexity measured by the number of points associated with that function. This measurement is accomplished by means of a matrix, such as the following, used by ourselves:

RET: Record Element Type   **DET:** Data Element Type

	1-19 DET	20-50 DET	51+ DET
1 RET	Low	Low	Average
2-5 RET	Low	Average	High
6+ RET	Average	High	High

Functional Complexity Rating	Unadjusted Function Points
Low	7
Average	10
High	15

Low: few data element types and/or referenced logical files; **High:** many data element types and/or referenced logical files; **Average:** neither Low or High.
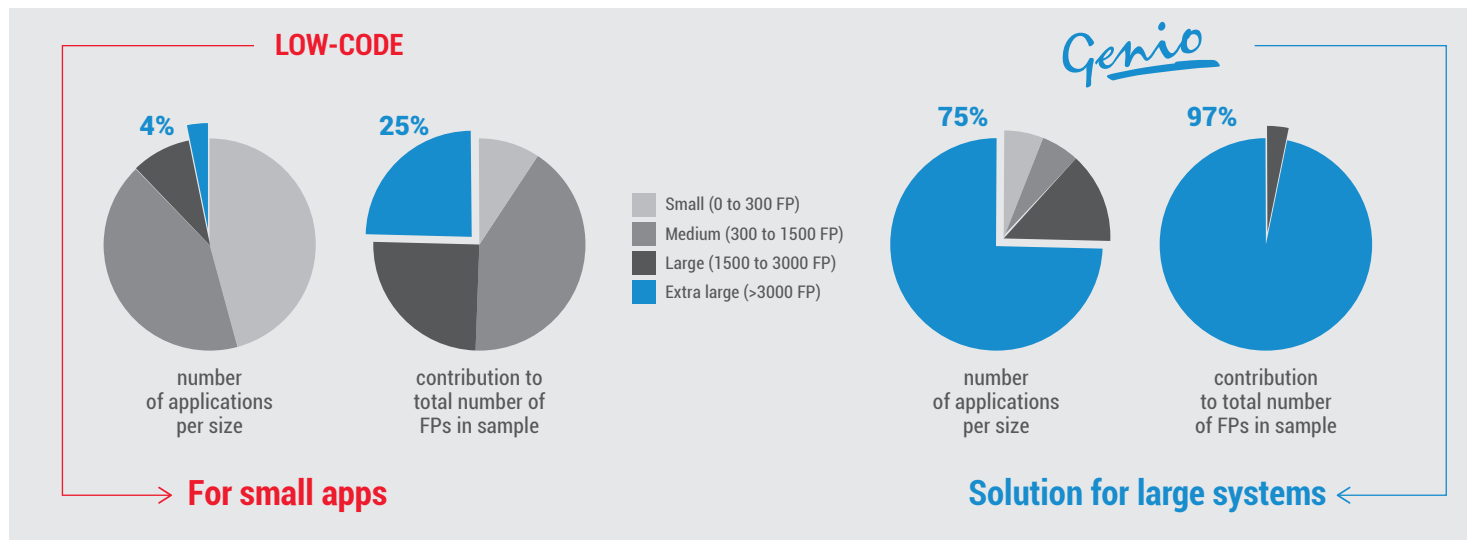
Comparisons based on published data

Some comparisons may be derived from data published by some Low-Code platform vendors. The comparison will be using the same metrics shown for Low-Code, although for the levels of productivity of Genio the scale is not the most convenient.

LOW-CODE

Genio

4%

25%

Small (0 to 300 FP)
Medium (300 to 1500 FP)
Large (1500 to 3000 FP)
Extra large (>3000 FP)

75%

97%

number
of applications
per size

contribution to
total number of
FPs in sample

number
of applications
per size

contribution
to total number
of FPs in sample

For small apps

Solution for large systems

Uncomparable complexity based on FP

From the available data we can also infer that the Low-Code platforms are most used in lower complexity apps.

57 449

7 750

Low-code
Genio
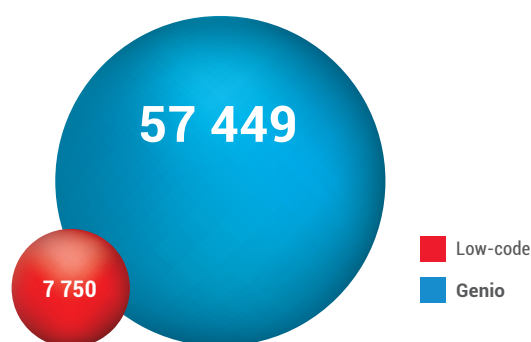
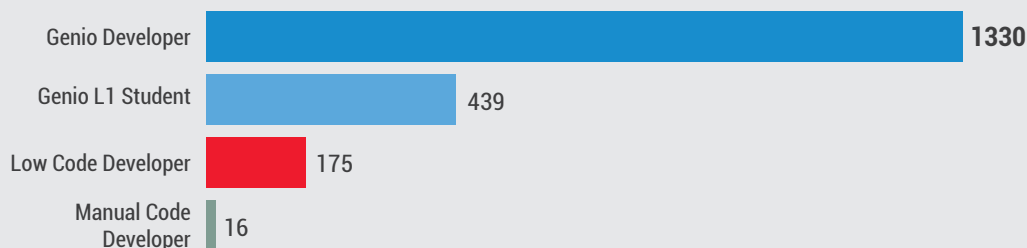# Genio very fast learning curve

One interesting finding shows a very fast learning curve with Genio. Quidgest's Academy provides several types of trainings, being the basic level (L1) a 5 day course at the end of which the student has an exam to perform. The following image shows the functions points associated with the exam itself and the ordinary training exercises during the classes:

Program	Area	Defs	*Indic*	ILF	EIF	EI	EO	EQ	FP_tot	No Hous	FP/h	Fp/month
Training: final exam	FOR	GENGDA0	*280*	11		16	3	19	49	2,00	24,50	2940,00
Training: daily exercises	FOR	GENSPOTIFY0	*525*	22		47	13	35	117	32,00	3,66	438,75

Surprise: productivity levels

Genio Developer — **1330**

Genio L1 Student — 439

Low Code Developer — 175

Manual Code Developer — 16

A Genio trainee, after 5 days of basic training and without any prior knowledge of coding, is able to produce software solutions with a complexity level of "Extra-Large" as defined by a Low-Code vendor published data.

Genio: AI-powered model-driven, automatic code generation platform

With 600+ active software systems, Quidgest is probably the largest software producer in Europe. All Quidgest's solutions are built at the model level, to satisfy specific customer requirements.

The 4 relevant components of a model, created by consultants or customers themselves, since no programming knowledge is necessary, are:

- **Data architecture**
- **Interface**
- **Processes**
- **Business rules**

Genio patterns are responsible for 95%-100% of the entire code.
If additional code is needed, it can be manually inserted into the platform to allow for a fully controlled and unified output.

FP/month metrics result from:

1. Function Points direct calculations based on the methodology defined by Nesma.
2. Work times recorded at Time Sheet Management embedded in Genio.

Do not feed obsolete solutions, replace them

Quidgest's Genio allows for the seamless upgrade from an obsolete solution by way of mimicking the interface and functionalities, having imported the data and its structure.

The "make-up" of obsolete solutions, just adding fancy apps to interact with them, is not the rational option. Performance is not improved and it will be more difficult in the future to get rid of all the obsolete environment.

This way the users will "see" the usual screens and functionality. No change management to deal with. However, now a entire new technology is being used, ready to go anywhere and implement all sorts of new functionalities.

(In)dependence from vendor platforms

LOW-CODE PLATFORMS	QUIDGEST'S GENIO AI POWERED MODEL-DRIVEN PLATFORM
Some Low-Code platforms produce proprietary code that only runs on the low-code vendor platforms, which imposes an ever-lasting end-user dependency on the vendor.	**Quidgest's Genio Model-driven platform produces standard code, even commented code, as if it was typed by a highly skilled developer, which runs perfectly without the presence of the generating platform.**

Conclusion

Moore's law basically states that processing power of computers will double every 18 months. This law has been valid for the last 50 years, but what has been happening, during the same period, with straight forward coding? Nothing, or very little all taken into account.

Genio, as an AI-powered, model-driven platform, is indeed opening a whole new gateway to implement complex solutions in a short time (for today's standards) and by smaller teams but still allowing for technology neutrality and keeping fully open all the options of the end-user.

Static software imposing fixed "best practices" or battalions of programmers to develop solutions from scratch or to (even) change the existing "fixed" ones represent undoubtedly unacceptable high costs both for countries, that need to join the Digital Nation's club, or large corporations, looking for the benefits of Digital Transformation.

AI powered Model-based code generation platforms are the new way to go. Meet Genio.