

SOFTWARE DEVELOPMENT WITHOUT FINAL TESTING



Quidgest

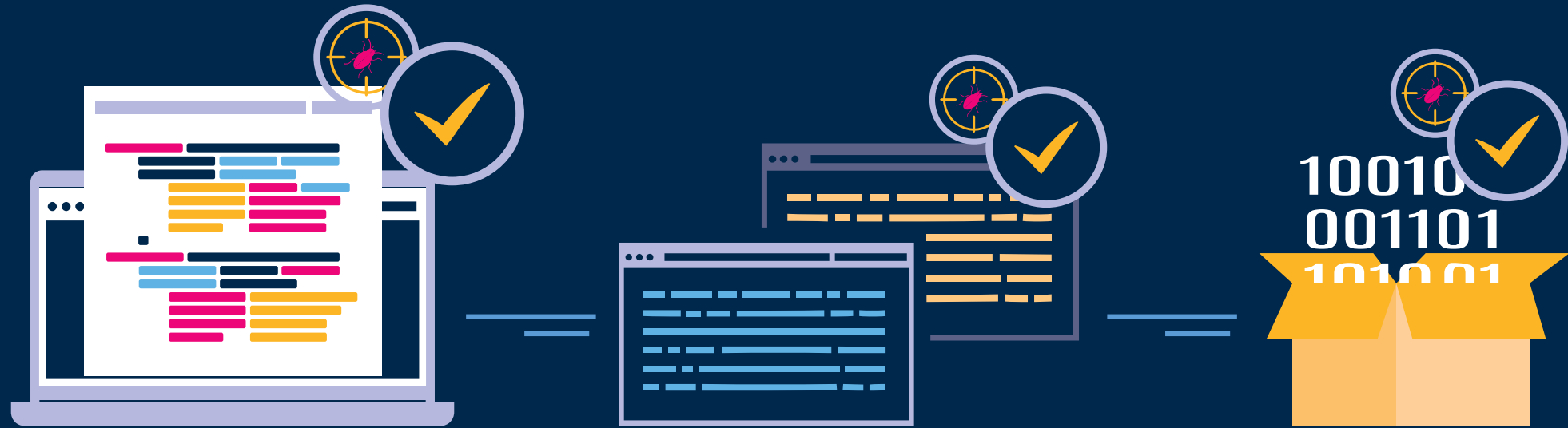
How much time and money do you spend doing tests after developing a piece of software?

Can you imagine creating complex software solutions without testing them?

NO ERRORS AT THE FIRST TRY?



SOFTWARE DEVELOPMENT WITH **BUILT-IN** QUALITY ASSURANCE



**The main problem
of software
industry is that
Software has
inferior Quality
standards than
most industries**

- Despite having adopted several proven methodologies and performing more testing than other industries, it still trails behind.
- Software producers assume that they are always going to find bugs in a software.
- Slow rate of improvement.
- Product stops working arbitrarily causing frustration to users.
- Technical debt is often not accounted for.
- More testing than other industries, worse efficacy.

It is a bit nonsensical that the software industry, seems to ignore Deming's principles and still allocates immense resources for quality control tests.



EFFORT (TIME/MONEY) ON SOFTWARE DEVELOPMENT

50%

50%

Software creation

Testing after creation

Creating software without bugs shouldn't be considered an impossible goal

“Cease dependence on inspection to achieve quality. Eliminate the need for massive inspection by building quality into the product in the first place.”

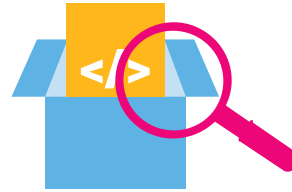
3rd Deming principle



- ✓ Follow Total Quality principles from the pioneer W. Edwards Deming
- ✓ Use Built-in quality at development phase
- ✓ Apply concepts like Shigeo Shingo poka-yoke
- ✓ Reduce tests based on inspection to a minimum, by promoting high quality product design and processes.
- ✓ Quality results from built-in, not from final, controls.

Quality Assurance (QA) is not Quality Control (QC)

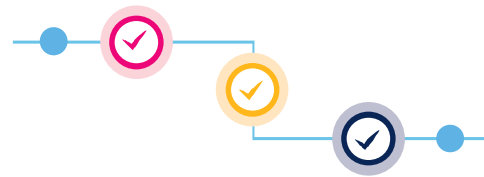
QA as usually executed in the software industry is **WRONG**.



Tests shouldn't be done at the end of the development cycle upon a "beta" product



Having dedicated testers working separately from developers is ineffective



There is a lack of focus on continuous process improvements



The only software generation platform with **built-in quality tests** just right on software creation

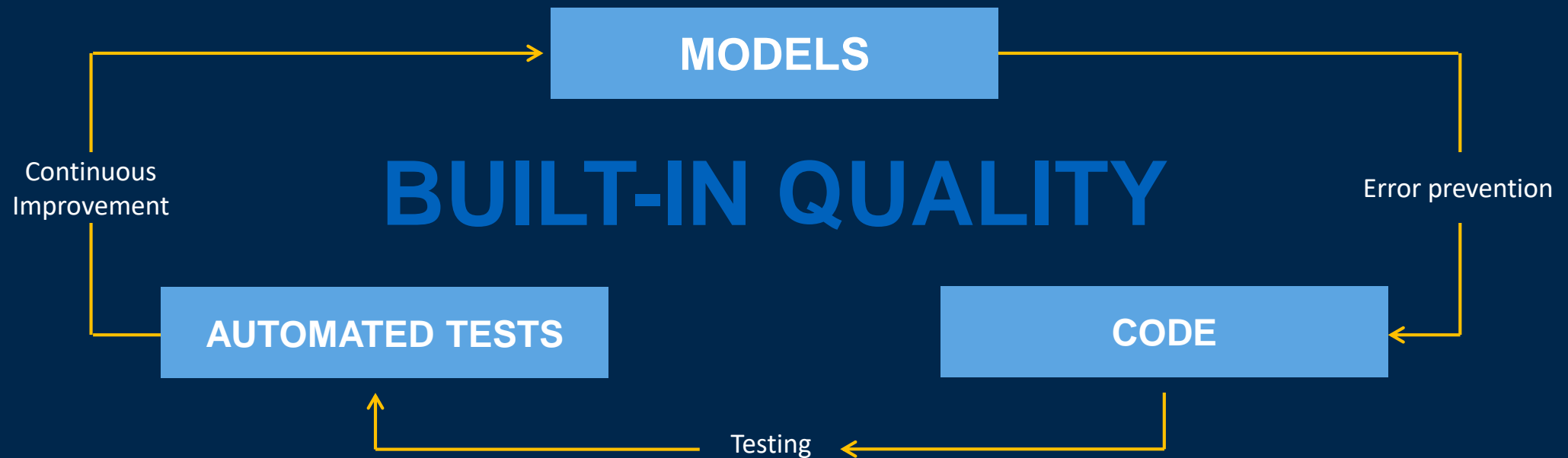
Based on model-driven engineering and artificial intelligence, this advanced platform prevents you from making most of the usual mistakes during the programming process.

Genio has been developed to put software quality at an unbeatable level. It is designed with software quality as a core principle and achieves it with built-in automated testing and validation. This enables developers to create software that works right out-the-box easily.





Built-in quality on the software development process is a unique approach developed and implemented exclusively by Genio!



Steer away from traditional quality control tests with these core principles

Kaizen 改善

Continuous improvement

Kanban 看板

Visual signs

Poka-yoke ポカヨケ

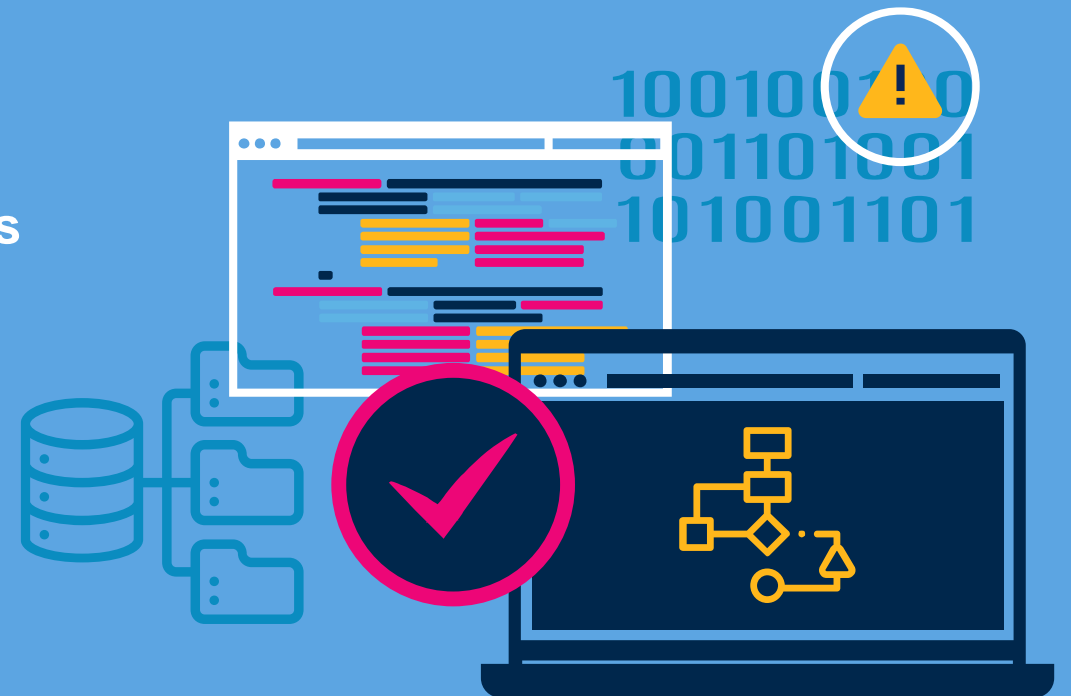
Mistake-proofing

Without a match, and more relevant than ever, this philosophy results from **+30 years of innovation** at Quidgest, due to disruptive business changes requiring better and faster software development processes.

Benefit from **model-driven software development** and from an industrial process highly customized

Genio models have built-in automated tests that can check for inconsistencies and errors

- » Untranslated strings
- » Incompatible data-types
- » Invalid database relationships
- » Invalid formulas
- » Missing resources
- » Missing configuration



In Deming's own words, **build quality in the process**, rather than inspect quality out of it

We build Quality into our models towards a point of **“zero quality control”**.

Quality issues are either prevented altogether, or caught at the source.

Quidgest

BUILT-IN QUALITY



100% Quality Assurance through
ZERO QUALITY CONTROL