aG CONNECT
the amazing world of IT

LOW-CODE NO-CODE DAY 2020

Low-code, no-code, where's my code!?

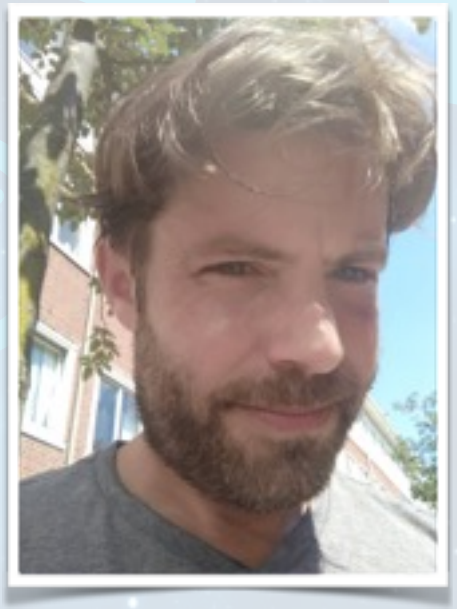
Tijs van der Storm

storm@cwi.nl / @tvdstorm

CWI
Centrum Wiskunde & Informatica

university of groningen

About myself…

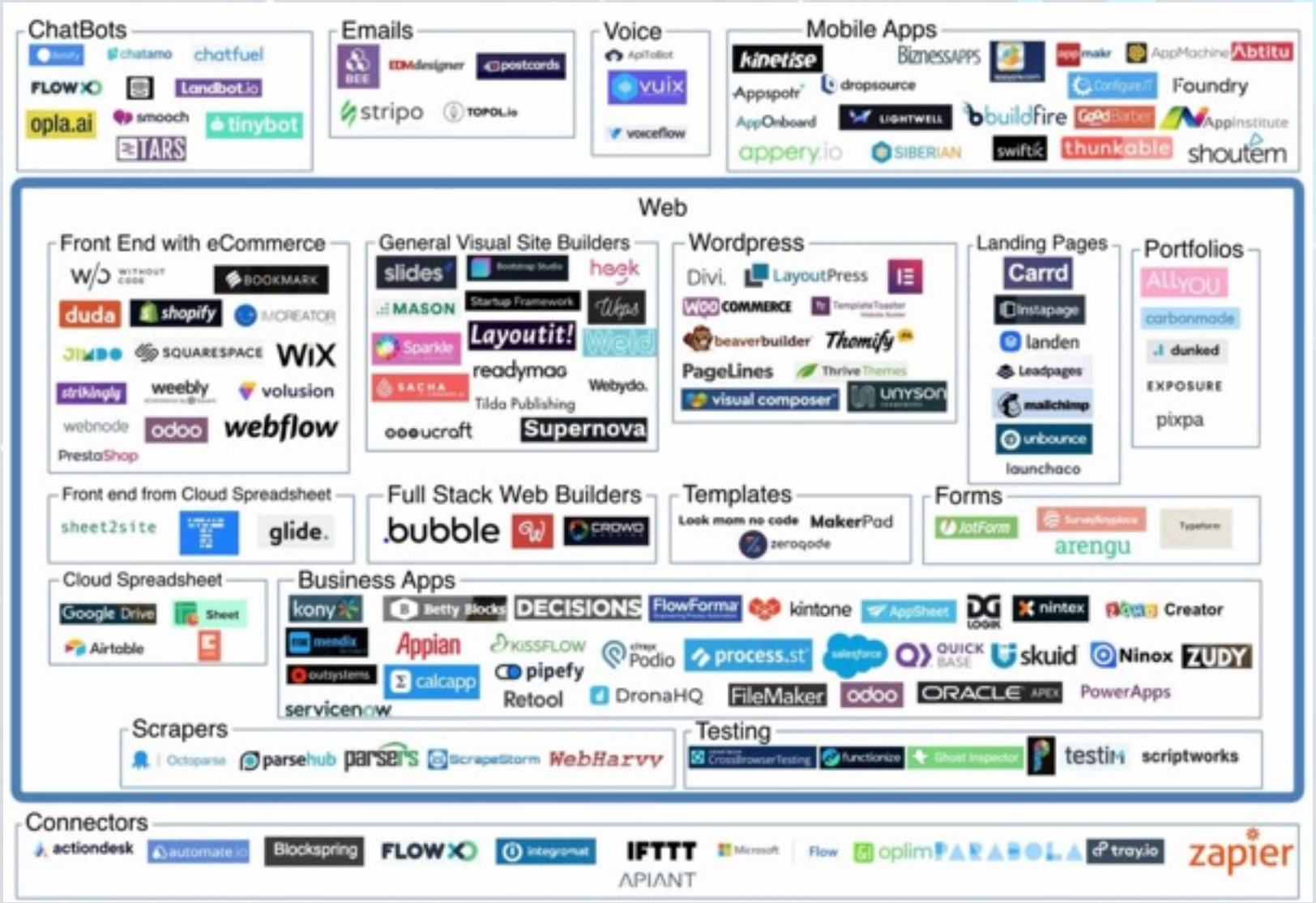


- Group leader Software Analysis & Transformation (SWAT) at Centrum Wiskunde & Informatica (CWI)
- Professor of Software Engineering University of Groningen (RUG)
- Research topics: Domain-specific languages, programming languages, language engineering, model-driven engineering
- Co-designer of Rascal, a metaprogramming language and language workbench (https://rascal-mpl.org)

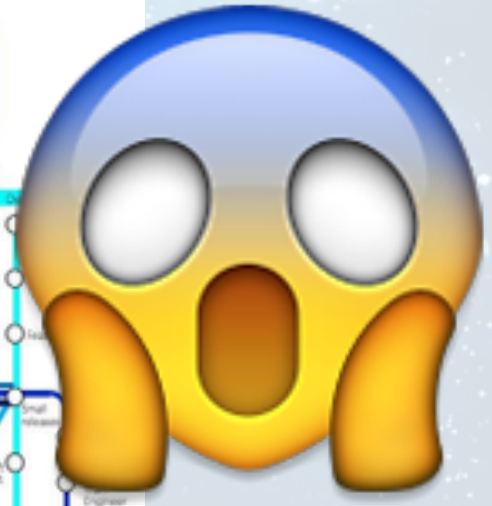
CWI

Centrum Wiskunde & Informatica

rijksuniversiteit groningen



I raise you: the Agile landscape ;)



Grady Booch
@Grady_Booch

No.

Will Low-Code and No-Code Platforms Revolutionize Progr...
In a new article in Forbes, a Business Technology professor at the Villanova School of Business argues that the way we ...
developers.slashdot.org

2:13 AM · Feb 24, 2020 · Twitter for iPad

68 Retweets **7** Quote Tweets **296** Likes


François Chollet
@fchollet
"No code" is just the beginning. I'm looking forward to no software, no computer, just living in the woods
10:40 PM · Feb 24, 2020 · Twitter for Android
446 Retweets
55 Quote Tweets
3.1K Likes

Grady Booch
@Grady_Booch

Replying to @manuelhe

One used to have to carry out division but writing a pile of machine language.

As I often say, the entire history of software engineering can be characterized by rising levels of abstraction.

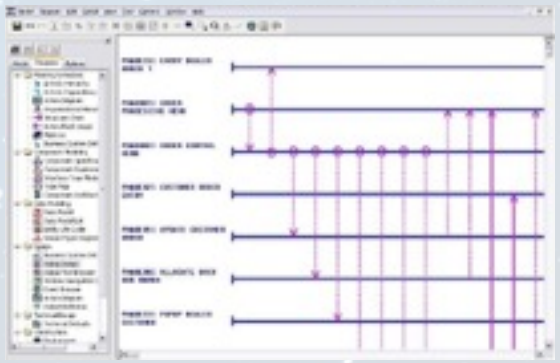
The action of coding never gos away; it just moves up another level of abstraction.

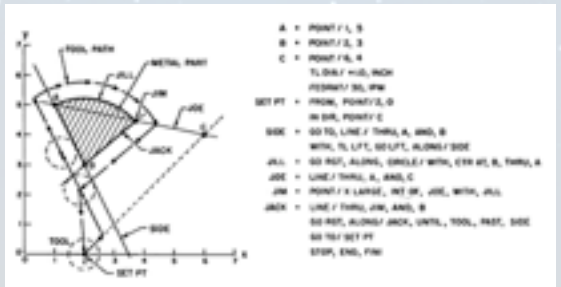
4:51 AM · Feb 24, 2020 · Twitter for iPad

15 Retweets **2** Quote Tweets **103** Likes

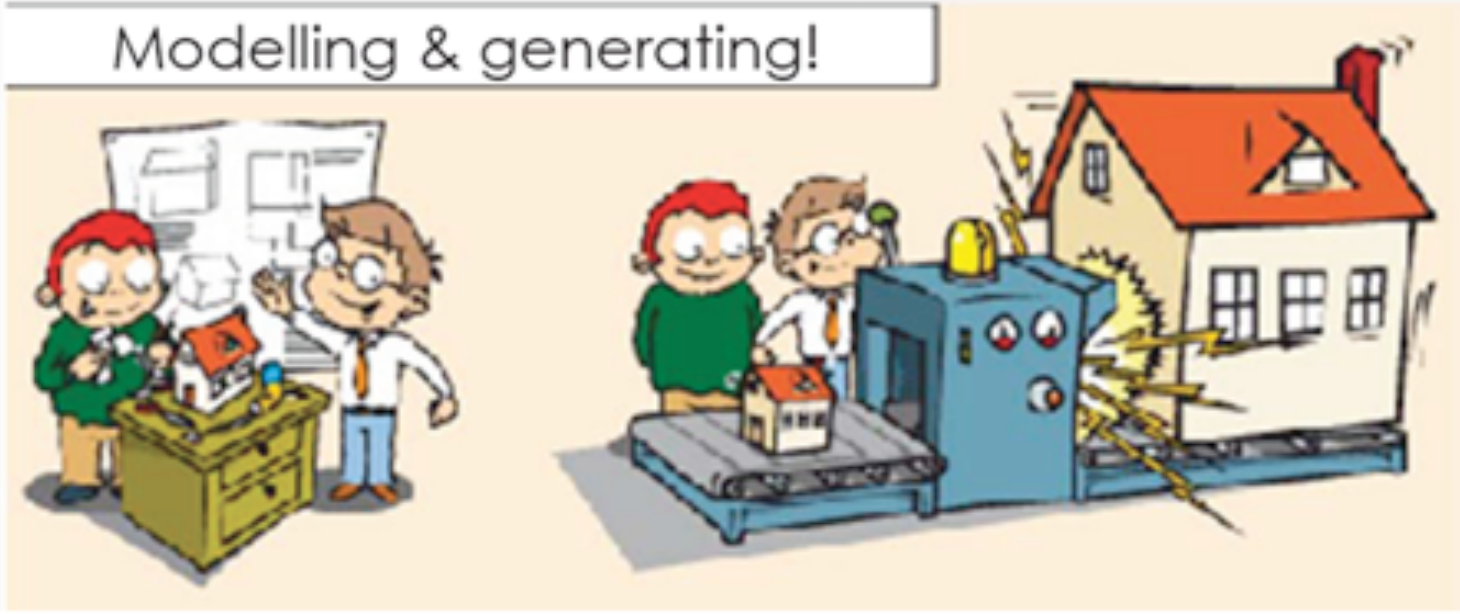
Are low-code/no-code platforms new?

- Computer Aided Software Engineering (CASE) tools
  - e.g., ISDOS, 1968
- Fourth generation languages (4GL)
  - James Martin, 1981: *Application Development without Programmers*
- End-user programming tools
  - Hypercard, spreadsheets, …
- Rapid Application Development (RAD)
  - Delphi, Visual Basic
- Model-driven engineering (MDE)
  - OMG MDA, modeling languages, …
- Domain-Specific Languages (e.g., APT, 1956)





Johan den Haan



Domain-Specific Languages (DSLs)

- DSLs are languages tailored to a particular problem domain
- Empowers domain experts/end users to develop software
- Potential benefits:
  - smaller programs means higher productivity
  - better notations improve validation with stake holders
  - domain-specific error checking and optimization
- Examples: SQL, HTML, “Make”, CSS, TeX, XSLT, Postscript, etc.
- **Low-code/No-Code ≈ “DSLs for business apps “**

www.rascal-mpl.org

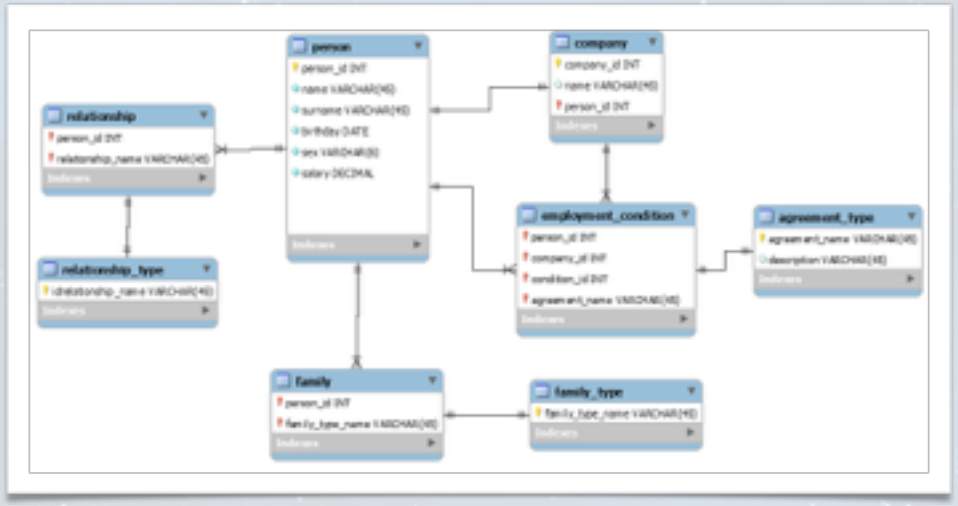
CWI

swat. engineering
control your software

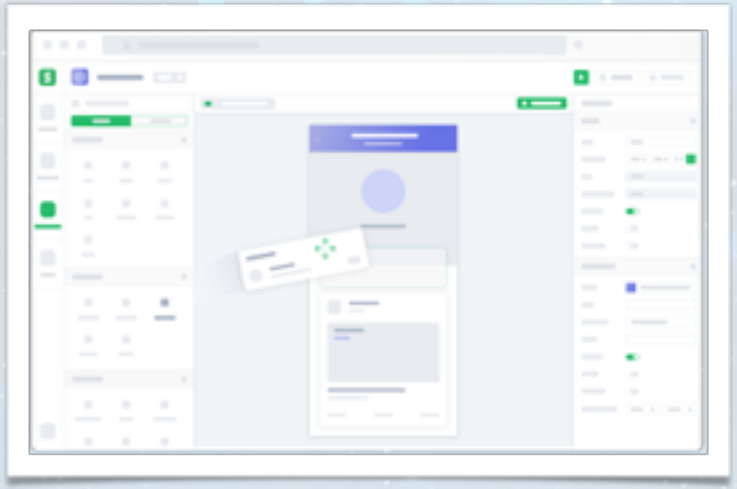
www.swat.engineering

The domain of business apps?

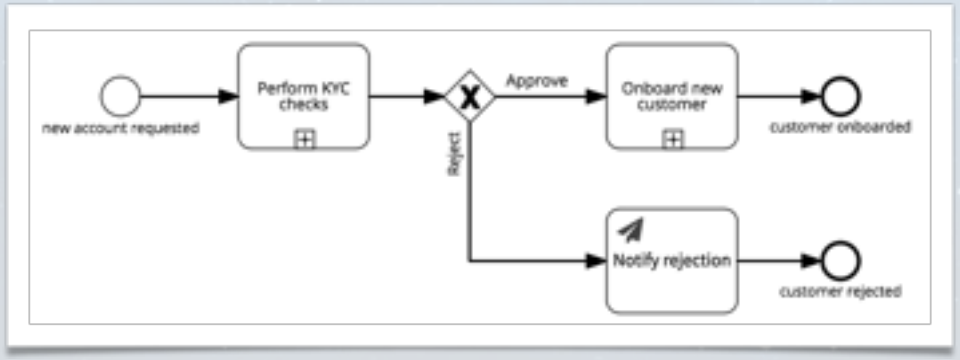
Data Modeling (ER/UML/…)



User Interface (forms/wizards/…)



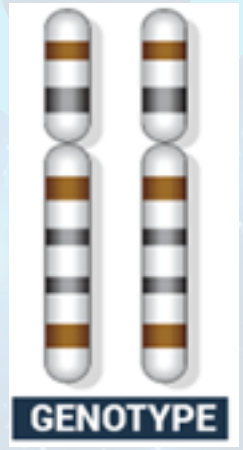
Workflow (flow/BPMN/…)



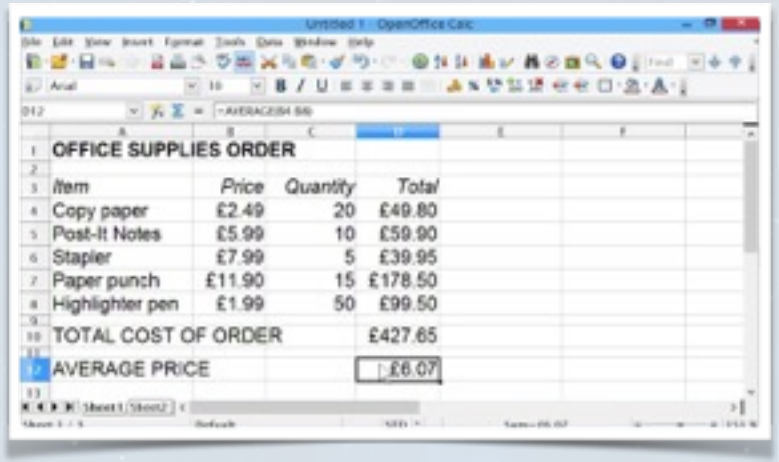
+some computation

So what does “no code” mean?

- Examples of UIs to **define and edit** business apps
  - diagrams, drag-drop, “builders”, …
- Structured information used to make computers do something
  - Also known as **CODE** ;)
- No-code = “no curly braces, parentheses, or semicolons…”
- Different ways (UIs) of specifying the same information
- A bit like
  - genotype (structure/information content)
  - phenotype (how it looks)

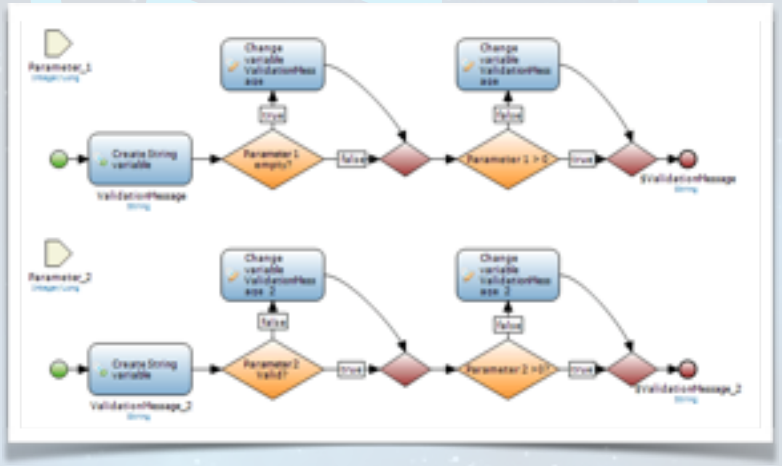


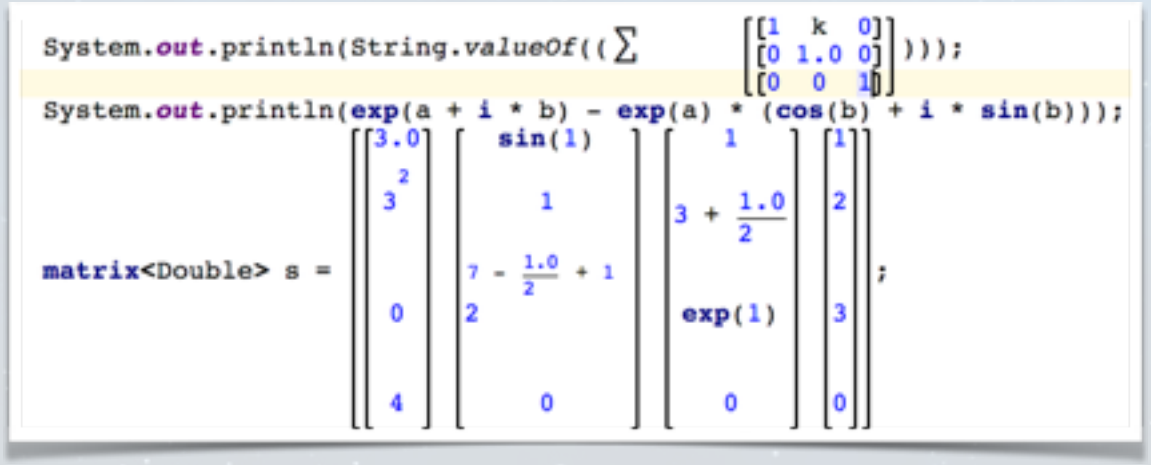
The many faces of code...

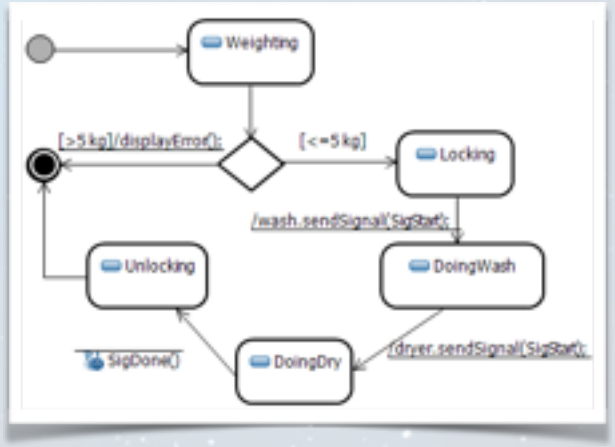


OFFICE SUPPLIES ORDER

Item	Price	Quantity	Total
Copy paper	£2.49	20	£49.80
Post-It Notes	£5.99	10	£59.90
Stapler	£7.99	5	£39.95
Paper punch	£11.90	15	£178.50
Highlighter pen	£1.99	50	£99.50
TOTAL COST OF ORDER			£427.65
AVERAGE PRICE			£6.07

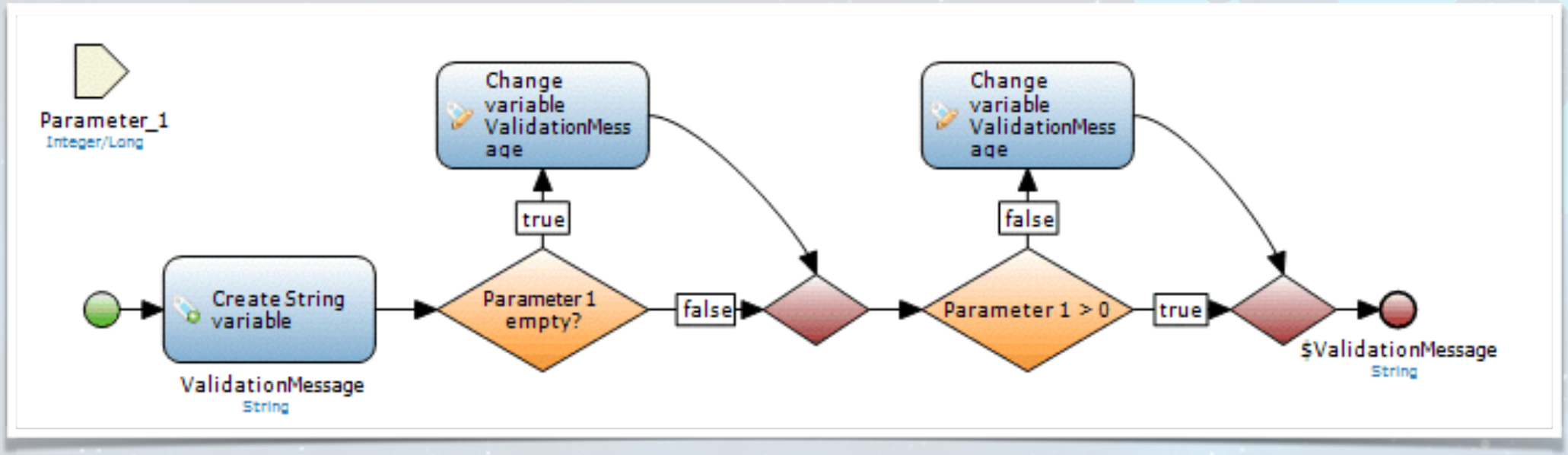






aGCONNECT the amazing world of IT

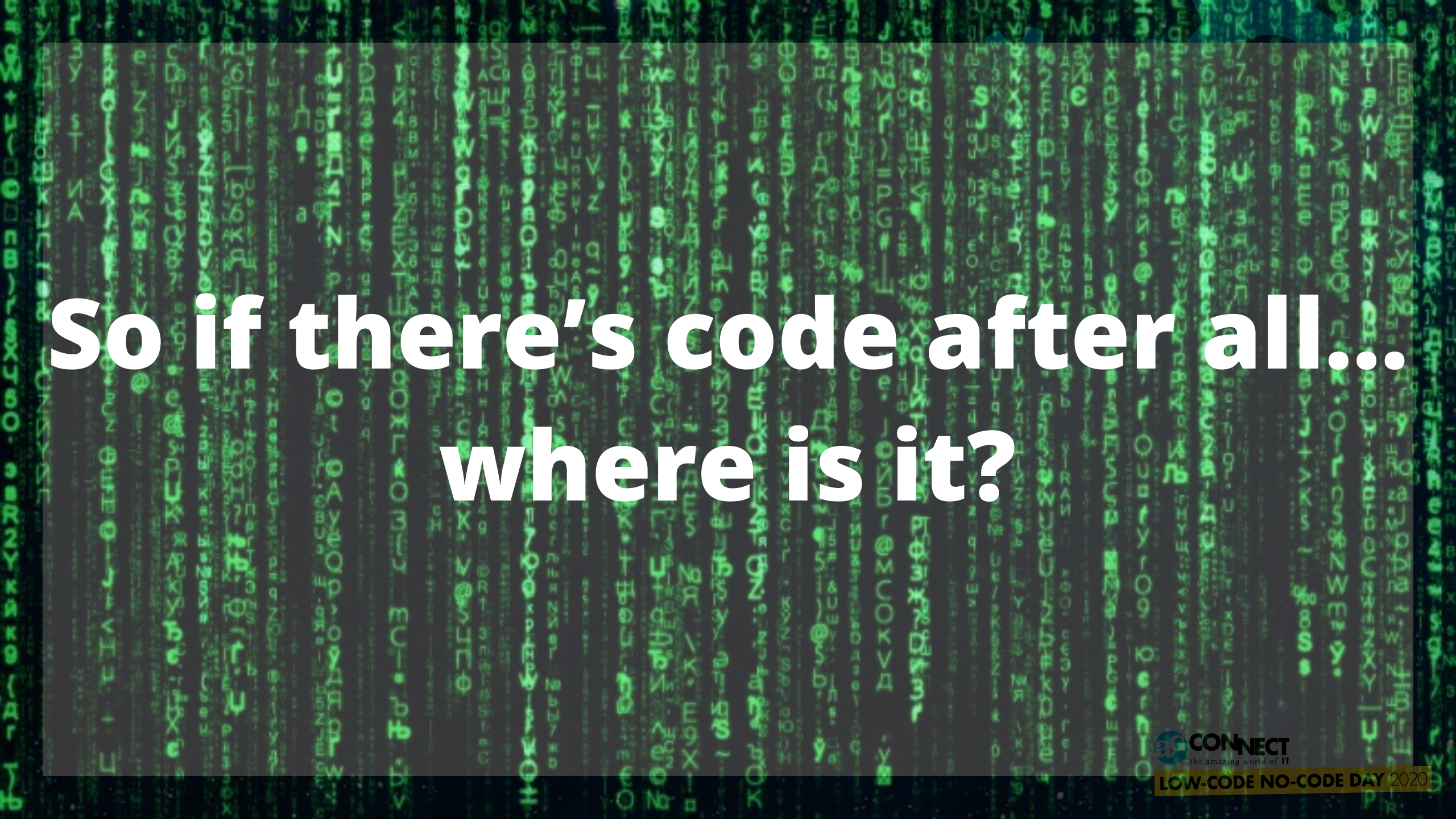
LOW-CODE NO-CODE DAY 2020



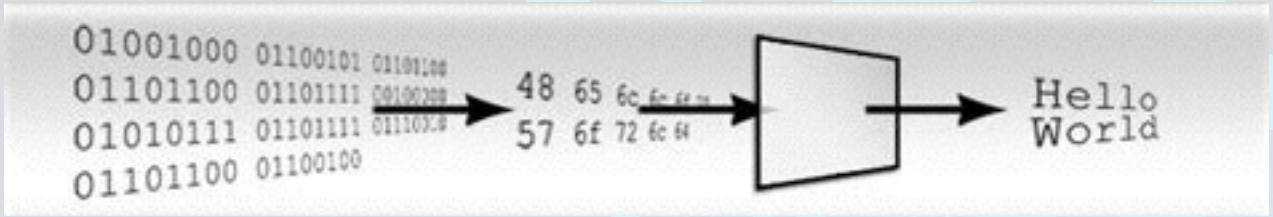
VS

```
if (Parameter1 == null) {
  ValidationMessage = "…";
}
if (!(Parameter1 > 0)) {
  ValidationMessage = "…";
}
return ValidationMessage;
```

Different phenotypes, same genotype


So if there's code after all...
where is it?

Where's my code?

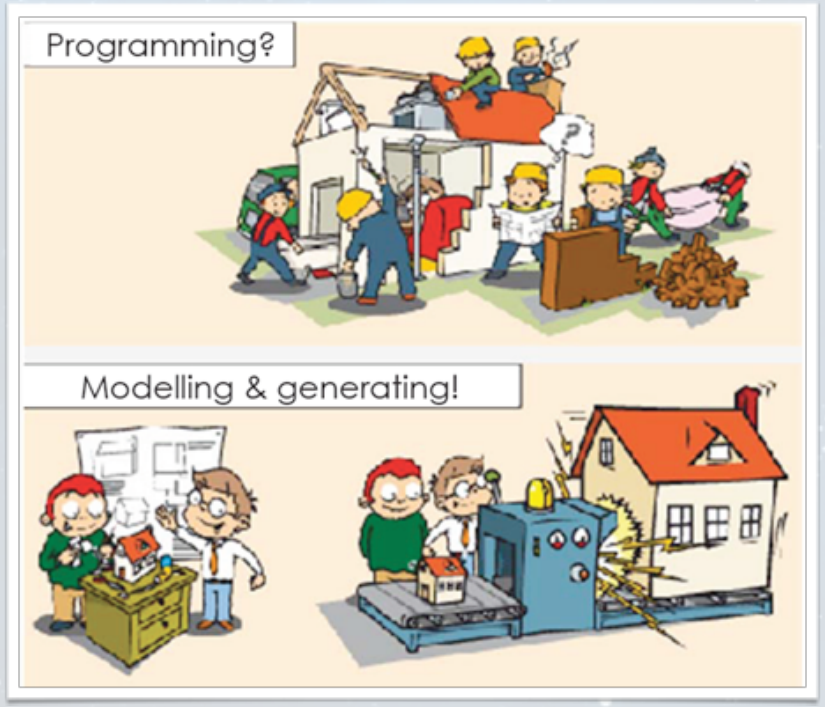


- **Format**: what's the storage format?
  - XML, JSON, YAML, plain text, proprietary binary, …?
- **Type**: how is it structured?
  - Is there a meta-model, a grammar, an XML Schema, …?
- **Storage**: how is it stored?
  - File system, database, repository, …?
- **Evolution**: how is the code versioned, released, deployed?
  - What is the traceability of code to running app?
  - How to compare versions (i.e. `diff`)

Where's my code?

- **Isolation**: how intertwined is the encoding with the platform?
  - Can we "look" at the code **without** the platform?
- **Ownership**: who owns my code?
  - What if the vendor goes broke, ends up in merger?
- **Sustainability**: what will happen in the long run?
  - E.g., 20 year from now?
  - Don't believe the lie "we generate maintainable code, here you go"

A nice vision…



But what if these get lost…

And we're stuck with a lot of this…?

Low-code, no-code, where's my code!?

- Low-code platforms **ARE** ***programming*** systems
- There **IS** code, but it just looks different
- DSLs for business apps with nice UIs
- Always ask the question: **where's my code?**
  - (because even it's hidden, it's there)