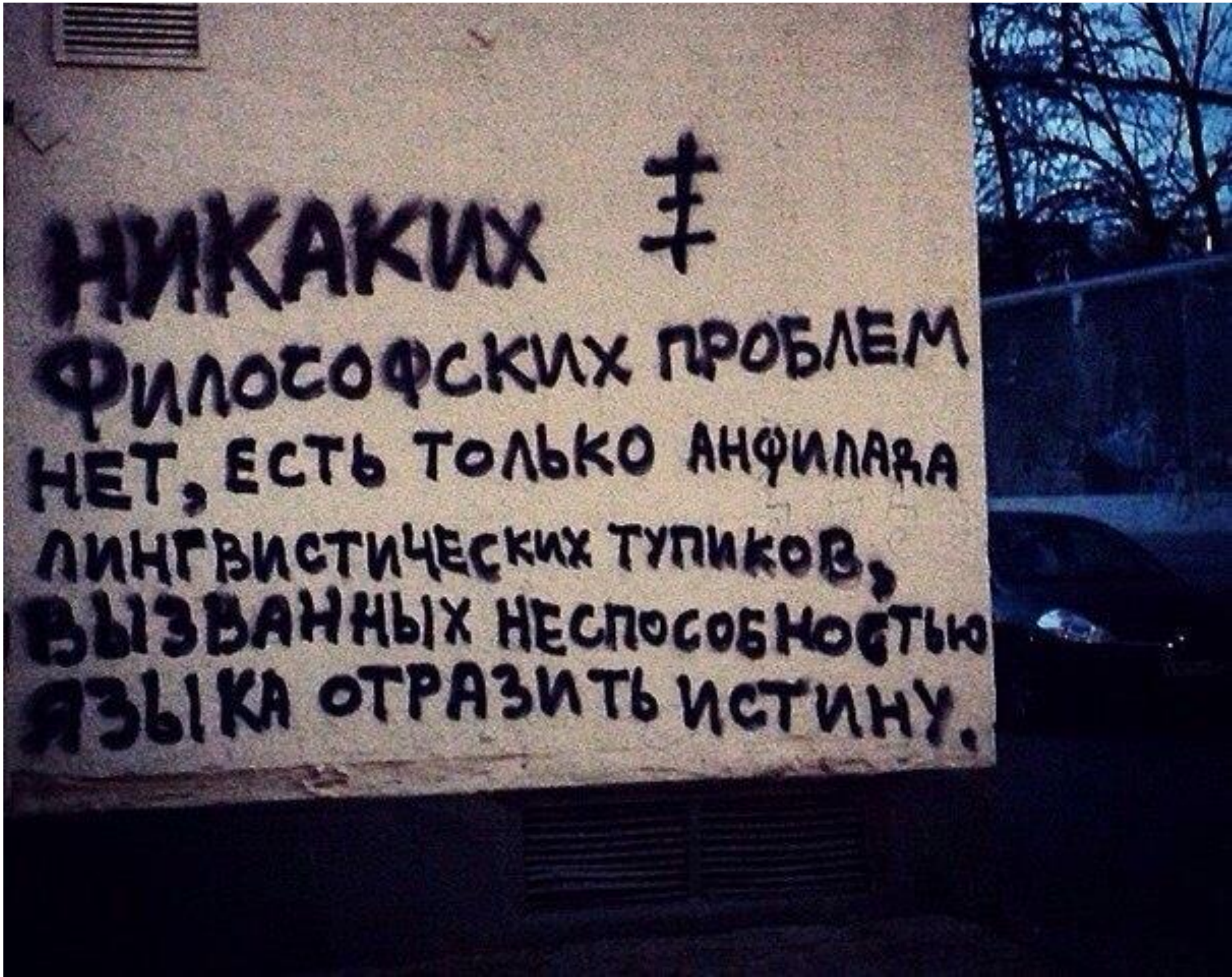
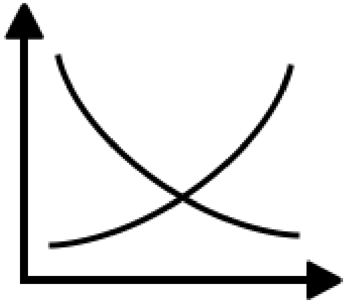
Development Methodologies

Waterfall,
Agile,
ScrumBUT
...

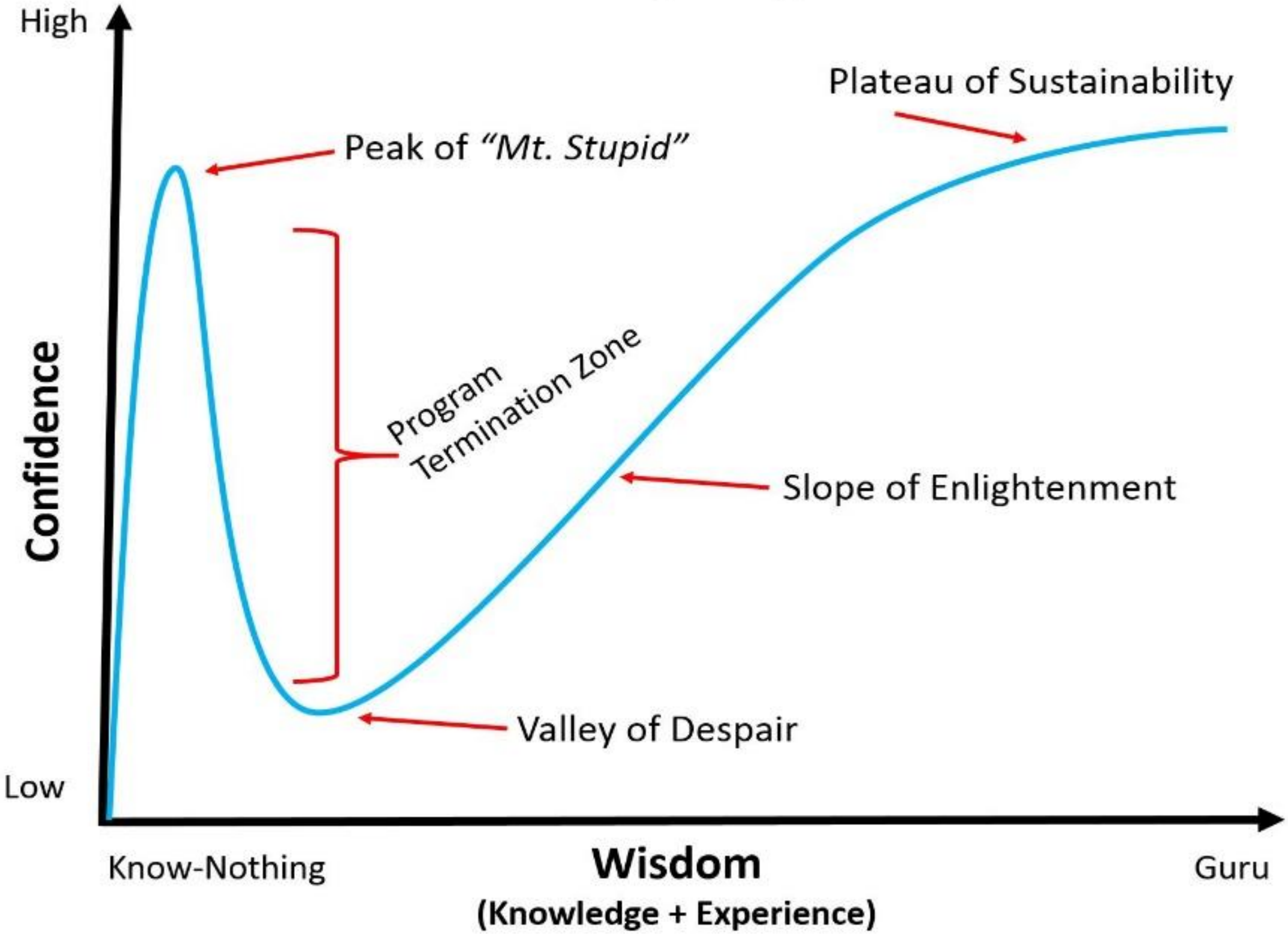


Outsource vs Product vs Startup. What is your next step?

- Software **outsourcing** company **is not** Tech company. **Key** of **success** is **effective** resource **utilization, ramp-up/ramp-down** set of **software** engineers
- **Non-IT** businesses. **IT** is **only** a **tool**, which **helps** to **solve non-IT issues**
- Tech company/startup – **Eldorado** with lot of **risks**



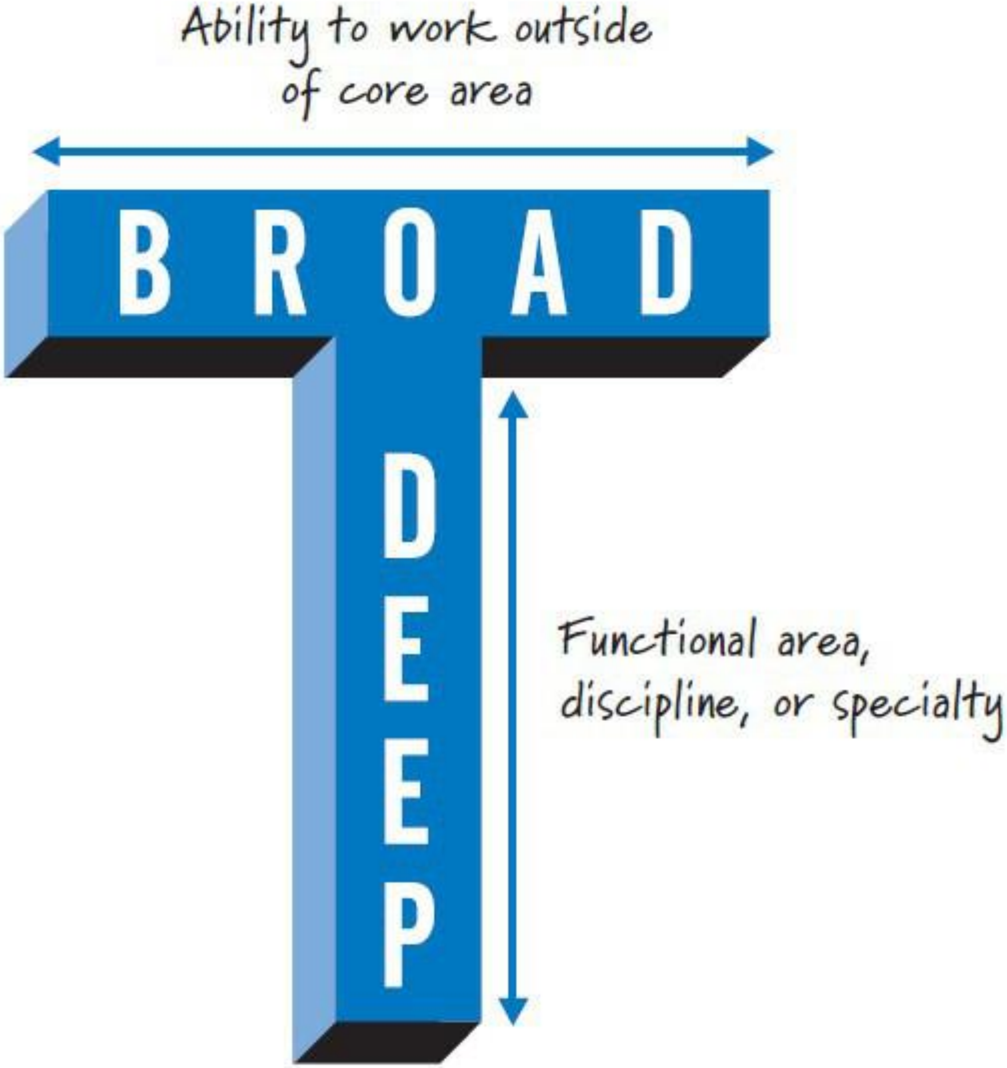
Dunning-Kruger Effect



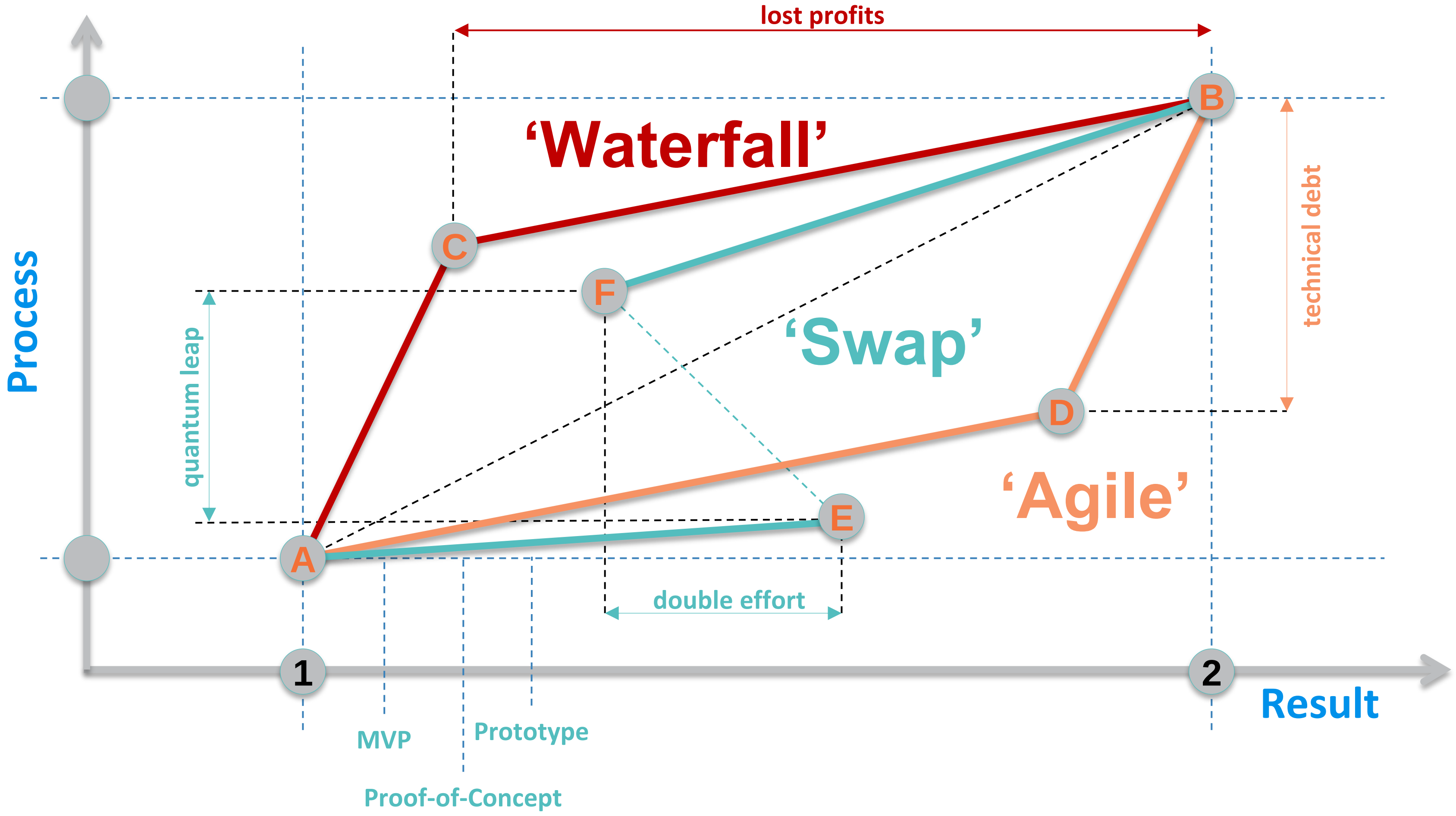
10,000 hours to master your craft.

That's
8 hours a day
5 days a week
44 weeks a year
for 5 and half years

Better get started.

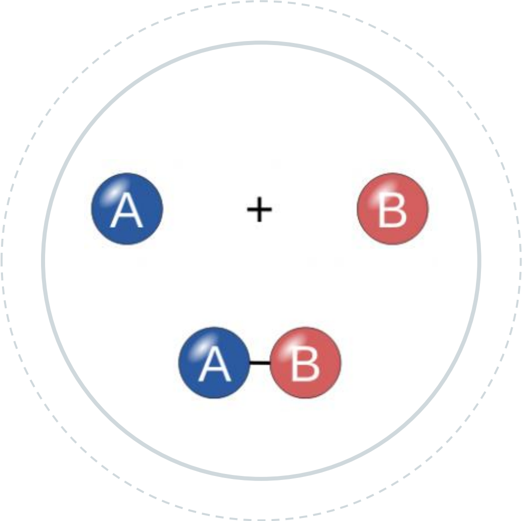

Ability to work outside of core area
BROAD
DEEP
Functional area, discipline, or specialty

Result and Process, Result vs Process



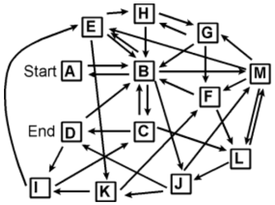
S.M.A.R.T.

- ***Specific*** – target a specific area for improvement.
- ***Measurable*** – quantify or at least suggest an indicator of progress.
- ***Assignable/Achievable*** – specify who will do it.
- ***Realistic*** – state what results can realistically be achieved, given available resources.
- ***Time-related/Time-bound*** – specify when the result(s) can be achieved.



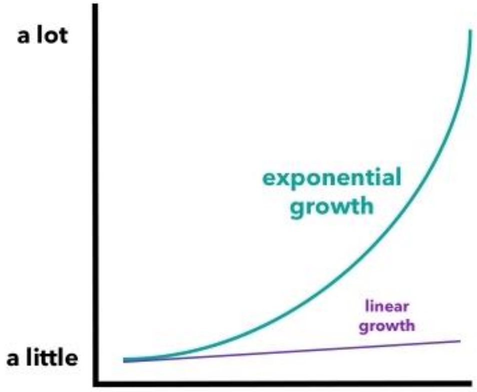
Decomposition
Composition
Verification

Why is Software Development so complicated?

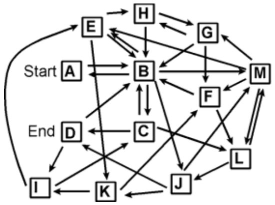


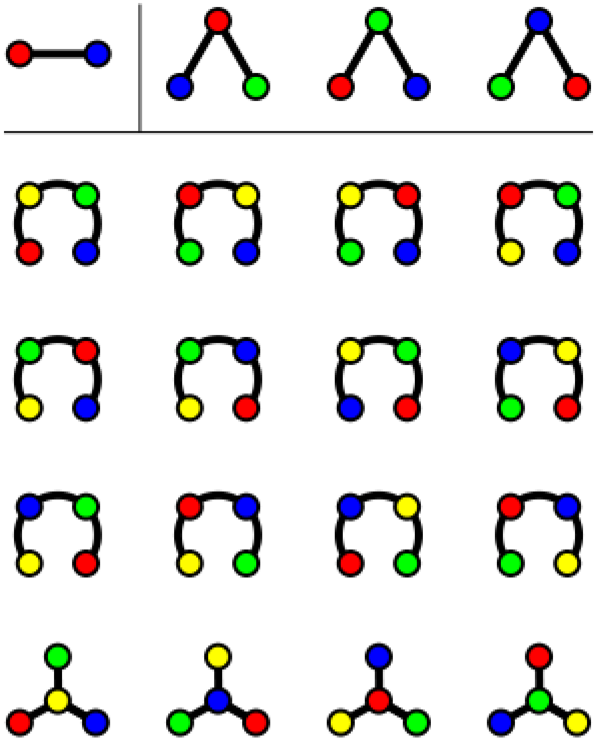
- **Speed** of tech **changes**, **legacy** products
- **Misunderstanding** between **business** owner, **management** and **tech** staff
- **Flexibility** of tech tools
- **Math(program)** model will never be **equal** the **original** system

Combinatorial Explosion

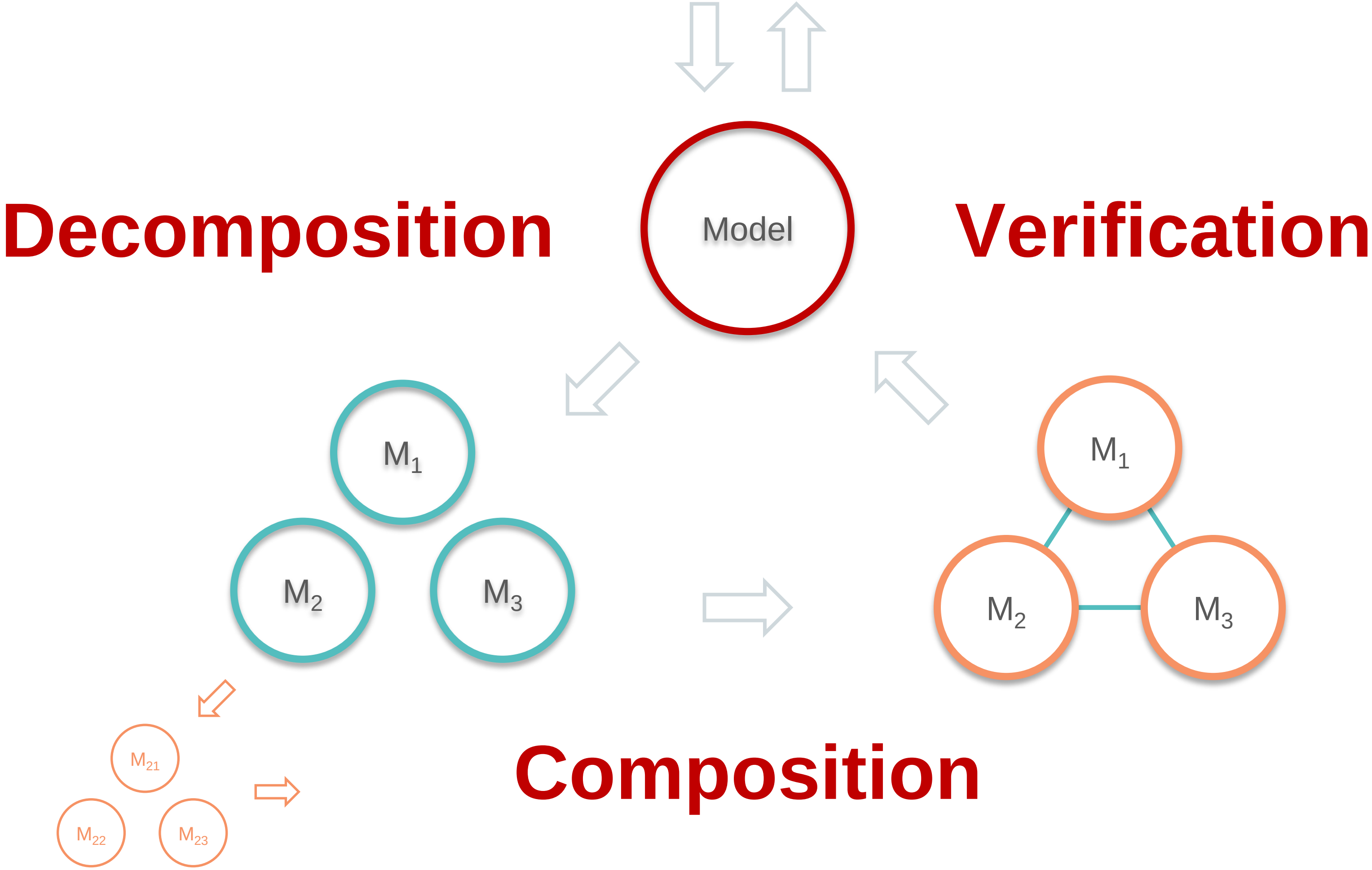


Why is Software Development so complicated?





Cycle of life



Decomposition

Verification

Composition

Variable name
Cache invalidation
Number of abstraction layers