Executable Architecture Models

The What, the Why and some How

uwe.wardenbach@gmx.de

Software Architecture (what is it?)

Software Engineering Institute | Carnegie Mellon University

- catalogue of definitions, form for submitting own
- the structure(s) of a system, which comprise software elements, the externally visible properties of those elements, and the relationships among them
- a depiction of the system that aids in the understanding of how the system will behave

Software Architecture (what is it?)

- components
- connectors
- visible properties
    - behaviours
    - constraints

Model
(what is it?)

- a usually miniature representation of something
- a description or analogy used to help visualize something that cannot be directly observed
- a system of postulates, data, and inferences presented as a mathematical description of an entity or state of affairs

Model
(what is it?)

1. stand-in for something
2. analogy/description/abstraction
3. degree of formalization

Model
(why do we need it?)

1. real thing not tangible or to hard to use
2. real thing not yet there
3. inspect and analyze qualities of the real thing
4. reason about properties of the real thing

Software Architecture Model (what is it and why need it?)

A Software Architecture Model is a

- ▷ somewhat formalized
- ▷ abstraction

  of a

- ▷ system’s parts - it’s components

  and their

- ▷ relationships - represented as connectors

  and the

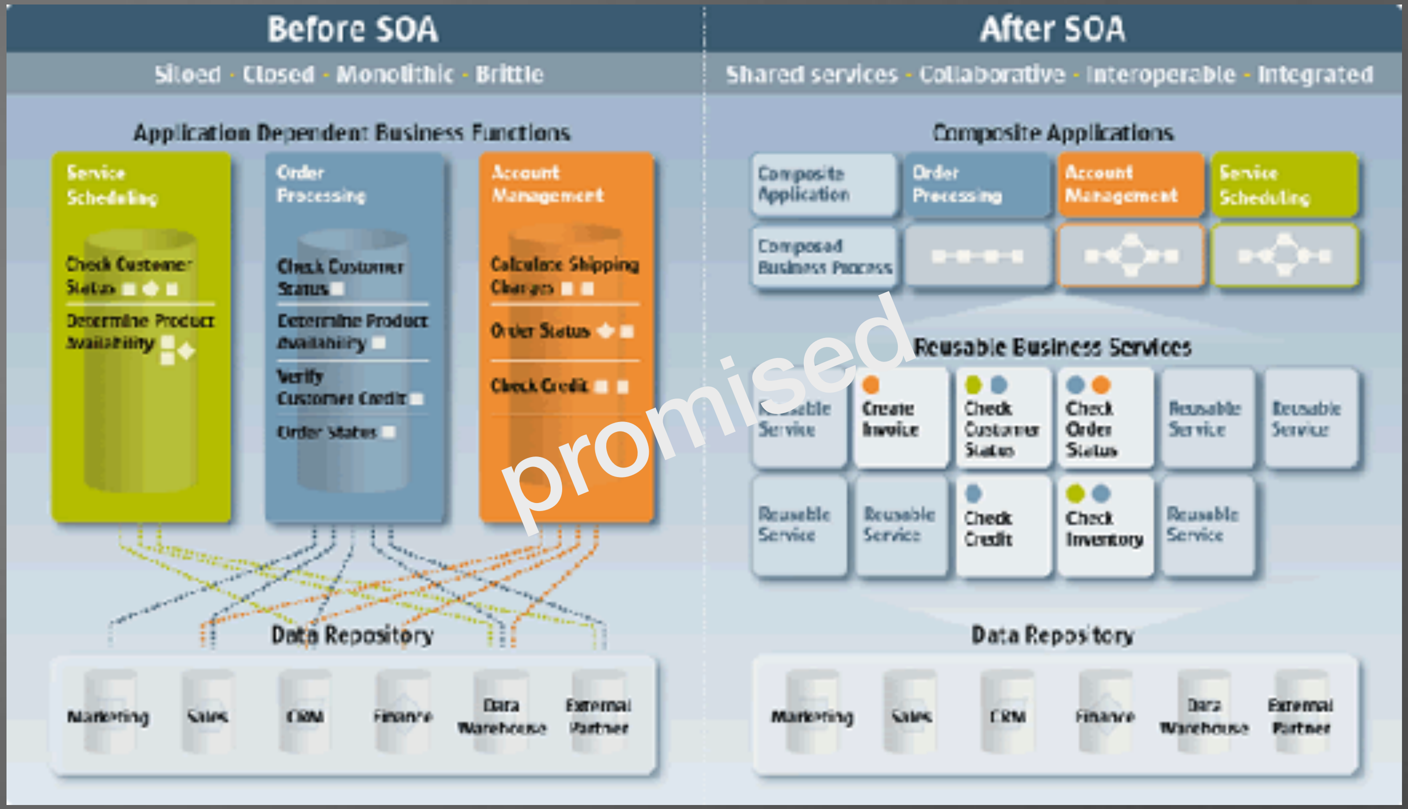
- ▷ externally visible properties - behavior & interaction

  with the purpose of

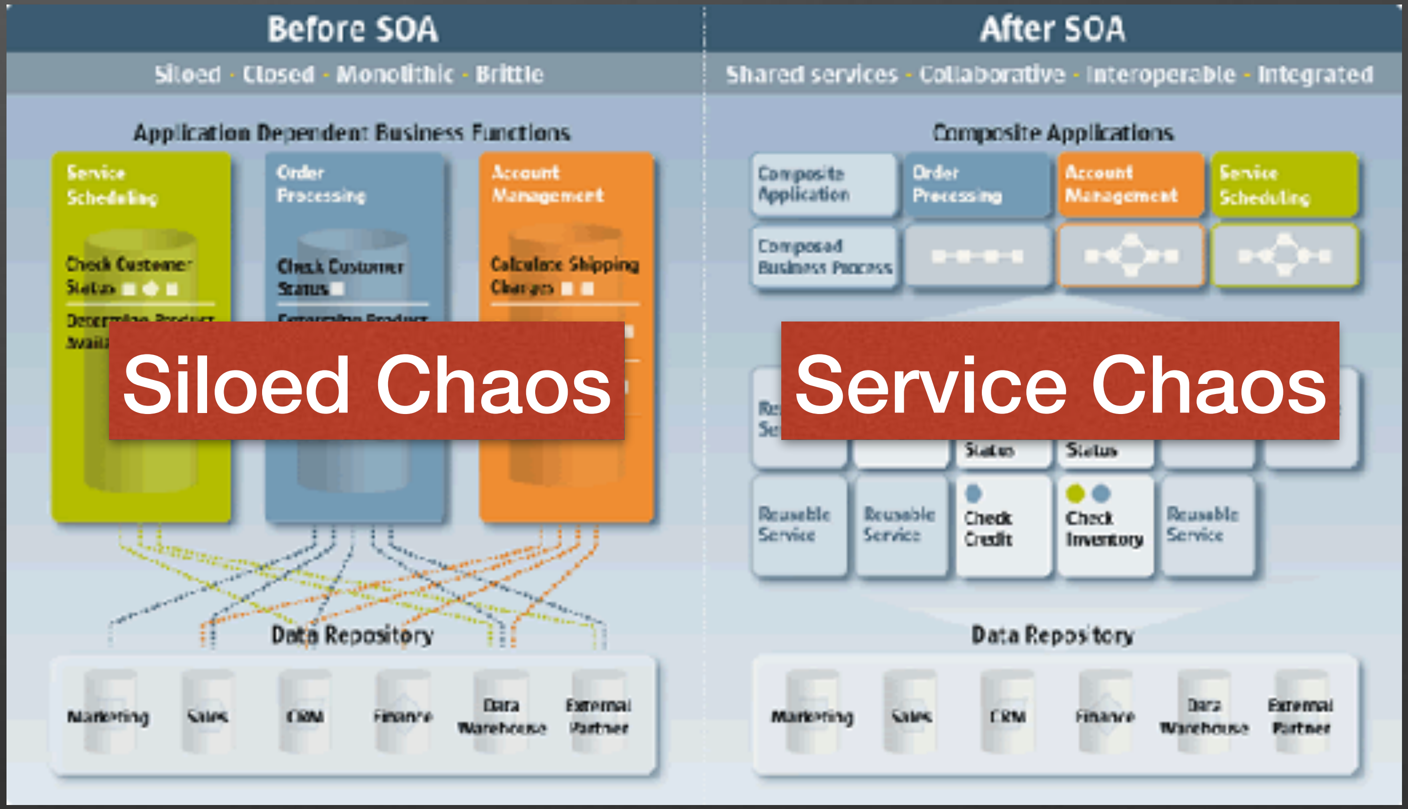
- ▷ understanding the system and reasoning about behaviour and qualities

Example Architecture Model

Free-form diagrams plus descriptions



promised



Siloed Chaos

Service Chaos

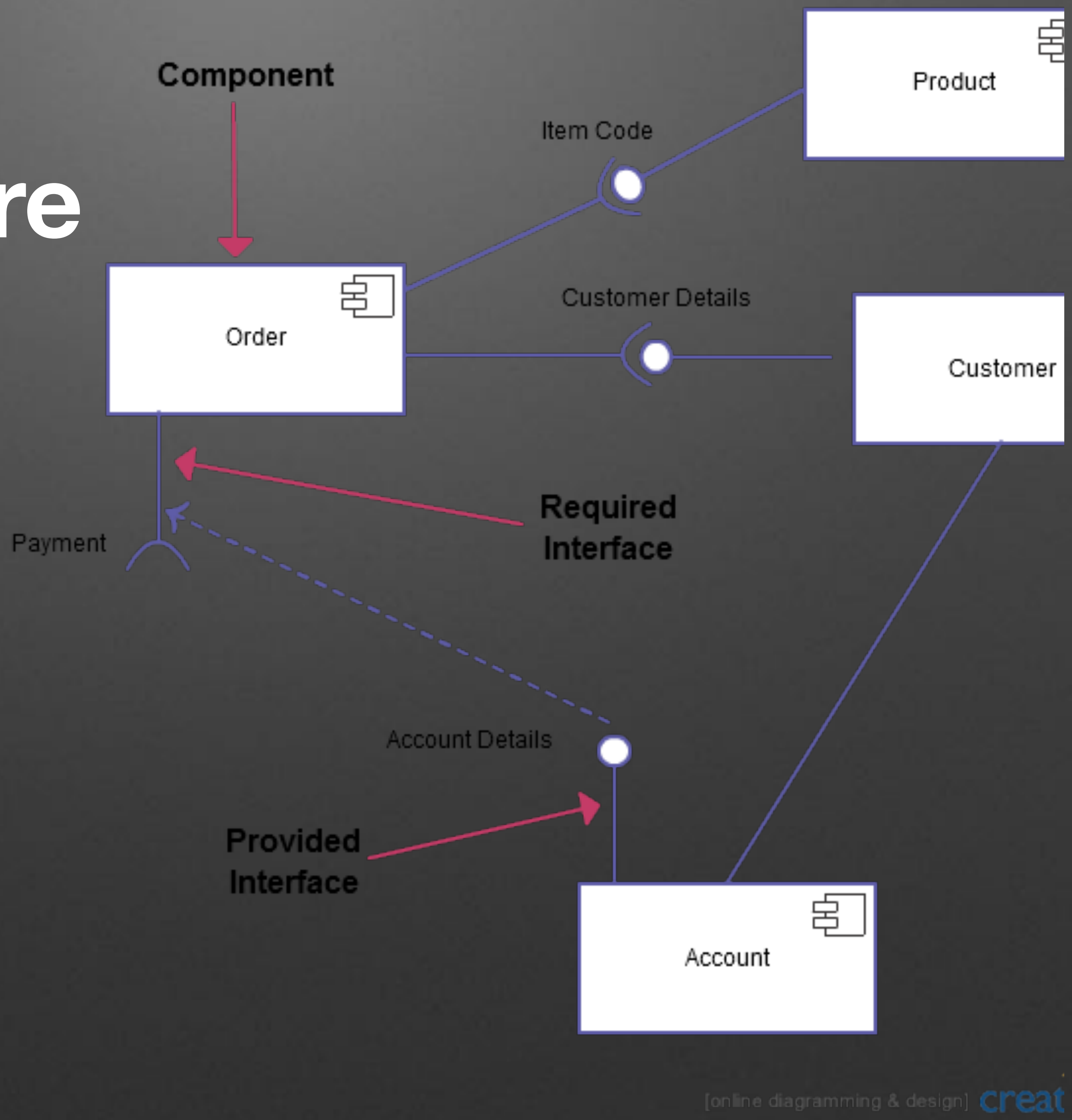
Example Architecture Model

Formal Languages

```
Configuration SimpleSimulation
Component TerrainModel(map : Function)
Port ProvideMap = [Interaction Protocol]
Computation = [provide terrain data] Component = VehicleModel
Port Environment = [Interaction Protocol]
Computation = [compute vehicle movement] Connector UpdateValues(nsims : 1..)
Role Model1   nsims = [Interaction Protocol]
Glue = [Data travels from one Model to another] Instances
Pittsburgh : TerrainModel([map of Pittsburgh]) PAT Bus : VehicleModel
C : UpdateValues(2)
Attachments
Pittsburgh.ProvideMap,PAT Bus.EnvironmentasC.Model
End SimpleSimulation.
```

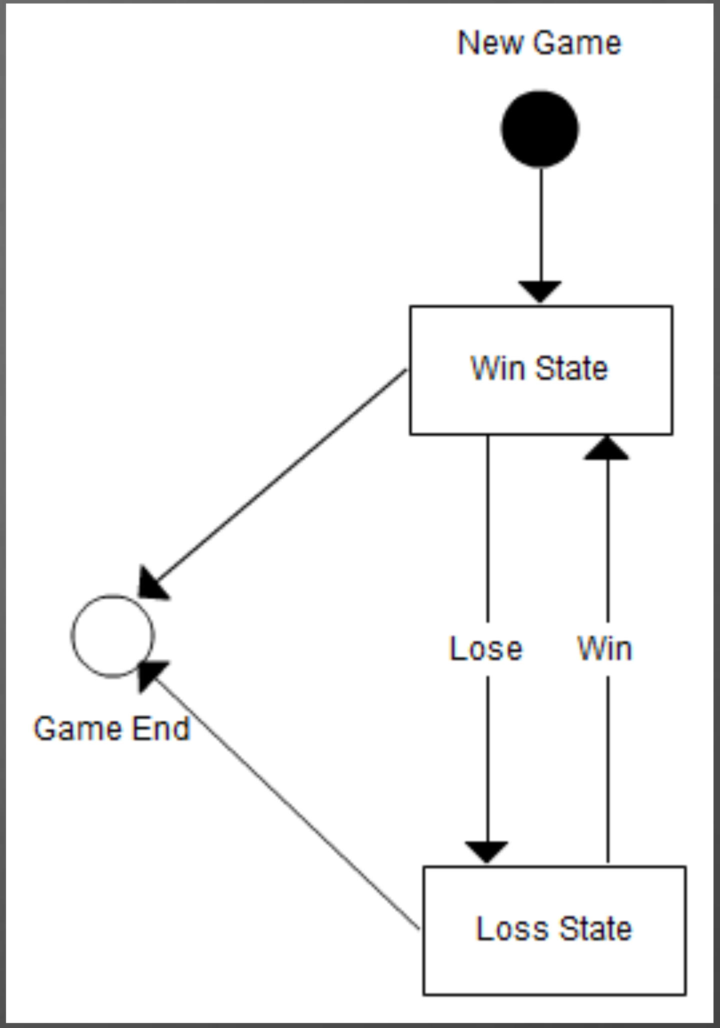
Example Architecture Model

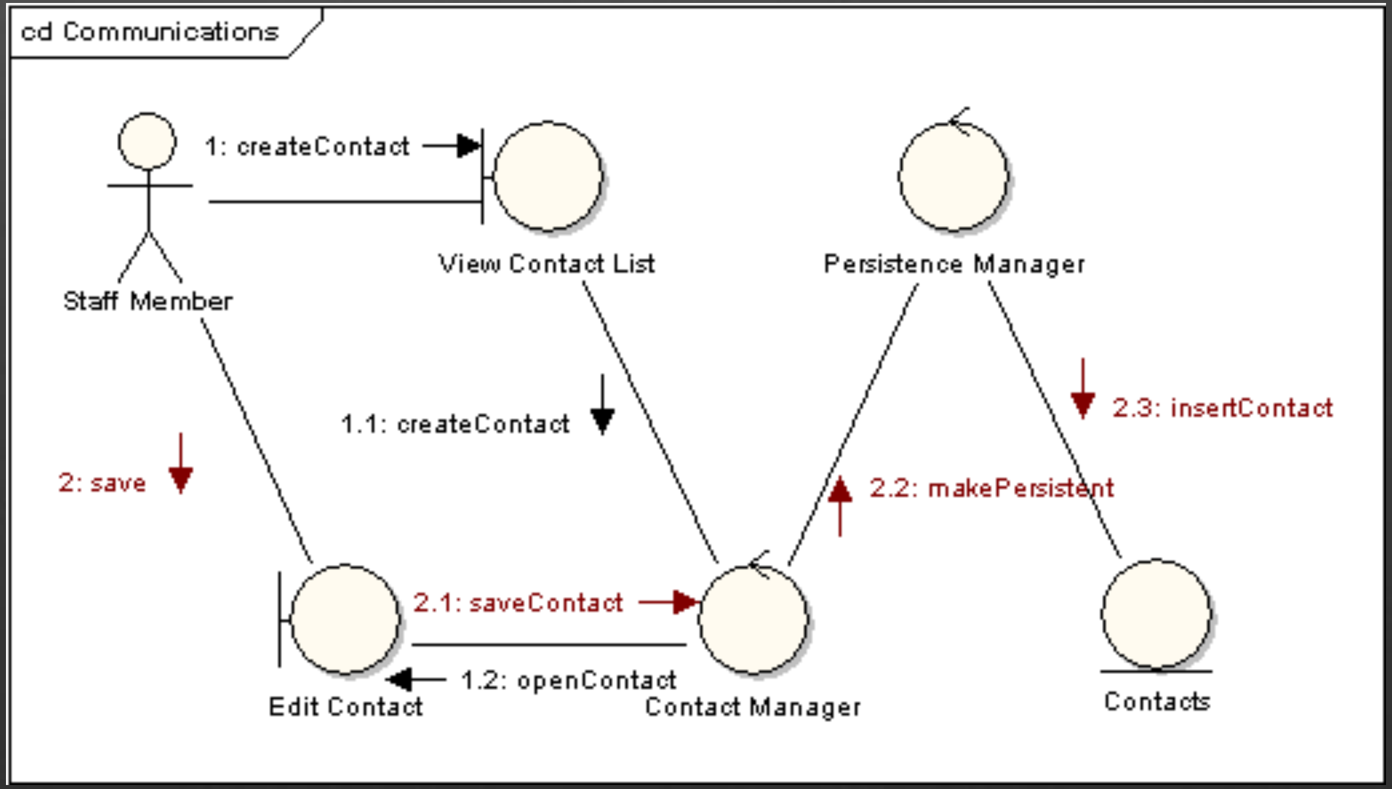
UML



Example Architecture Model

UML

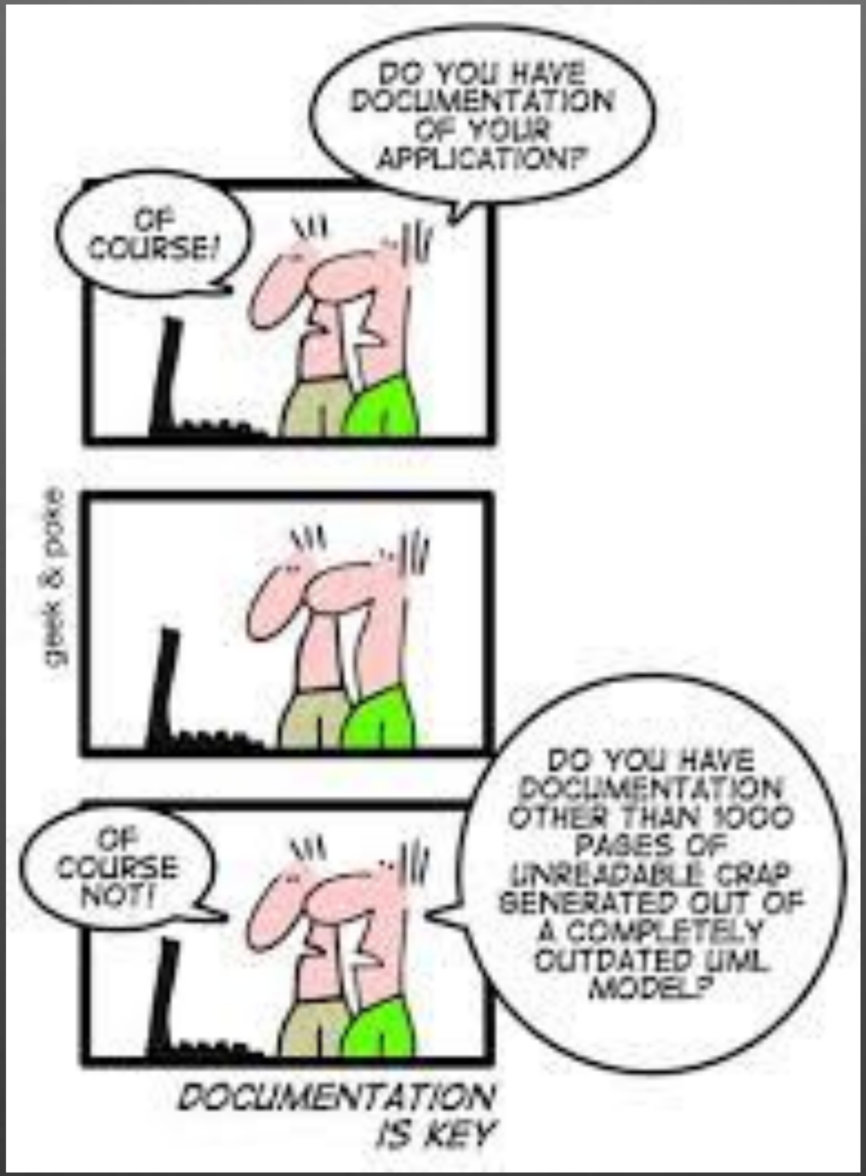




The Problem With Diagrams

The Problem With Formal Languages

The Problem With UML


DO YOU HAVE DOCUMENTATION OF YOUR APPLICATION?
OF COURSE!
geek & poke
DO YOU HAVE DOCUMENTATION OTHER THAN 1000 PAGES OF UNREADABLE CRAP GENERATED OUT OF A COMPLETELY OUTDATED UML MODEL?
OF COURSE NOT!
DOCUMENTATION IS KEY

Approach:
let’s code our architecture

1. abstraction: don’t do it, just specify it
2. formalization: code - invent your own ADL
3. reasoning: re-design & re-factor
4. inspection: automated testing