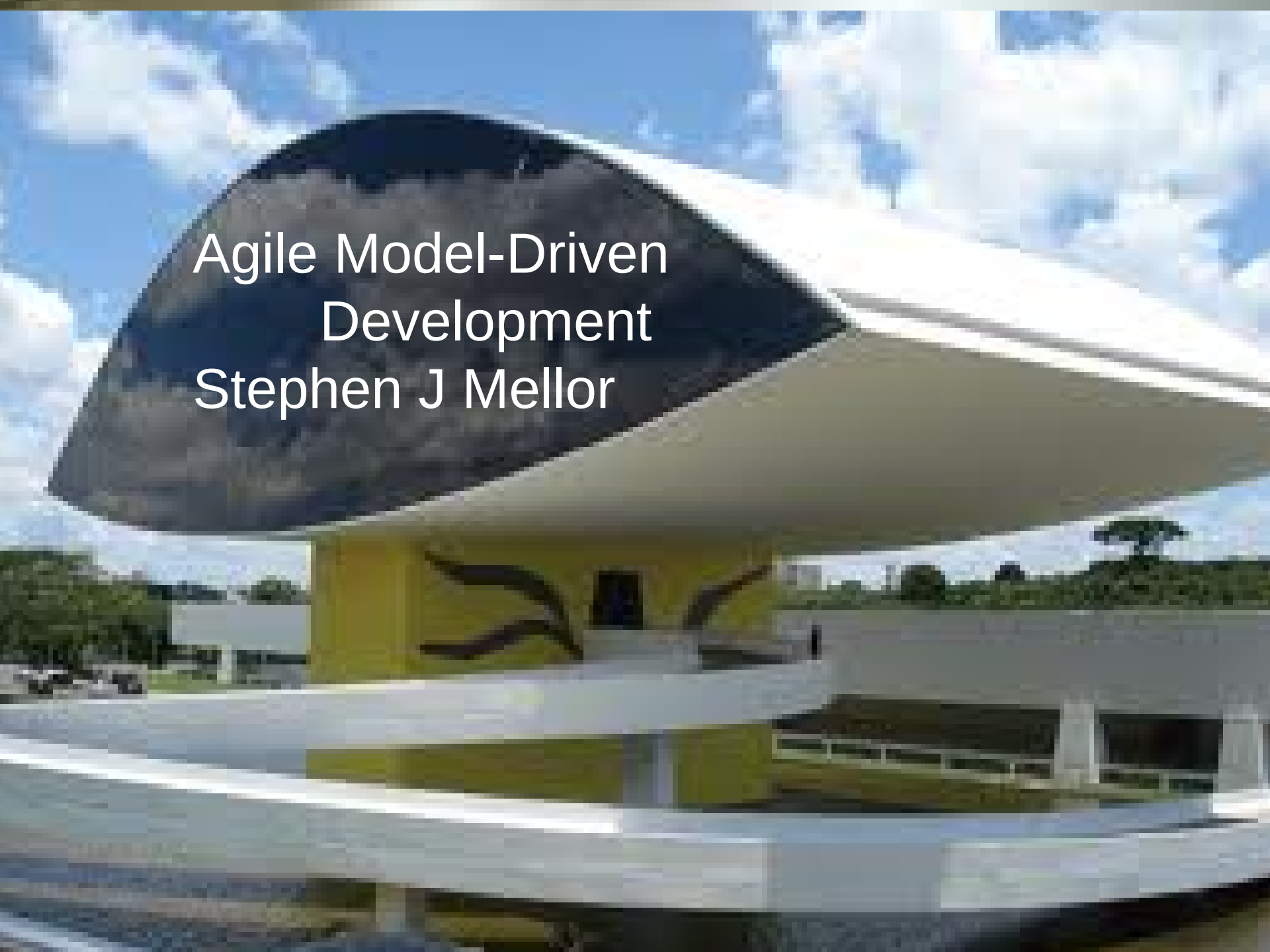
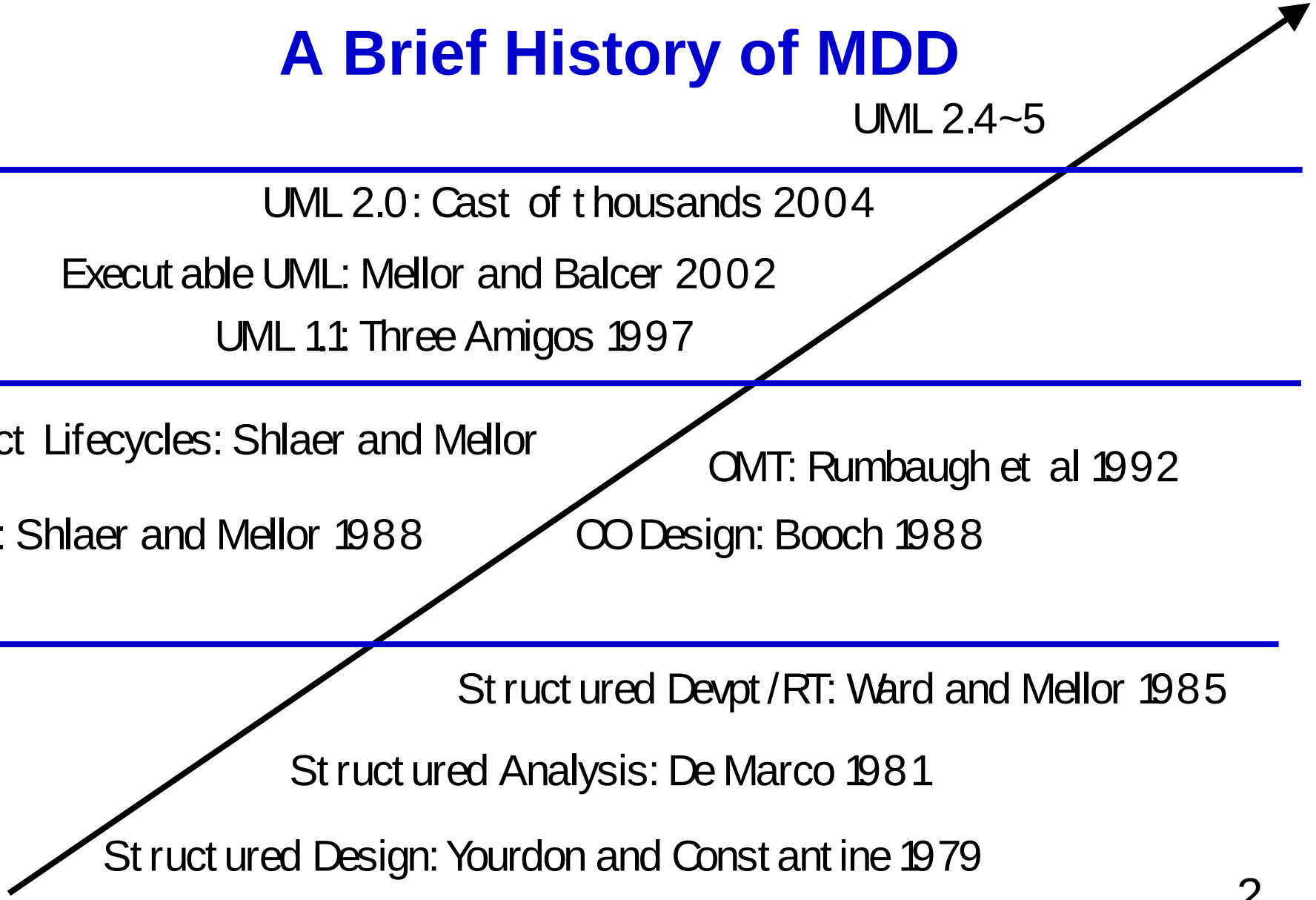
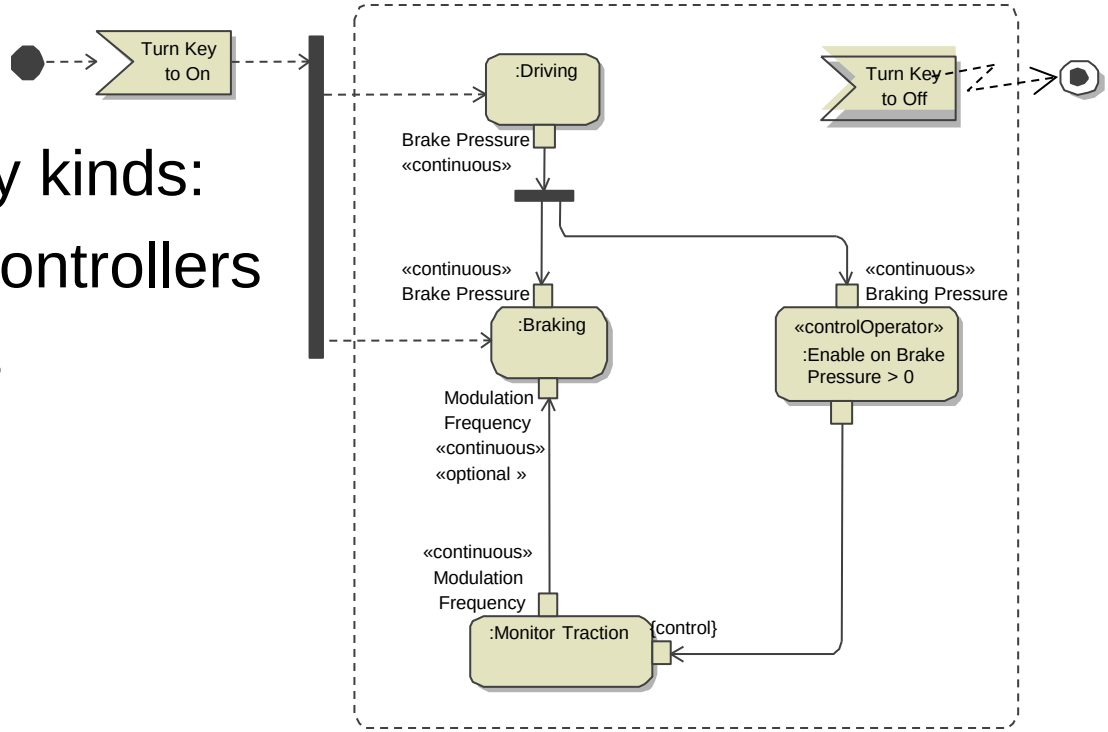

Agile Model-Driven
Development
Stephen J Mellor

A Brief History of MDD



Model-Driven Development

Model-driven development is the idea that we can transform models into systems.

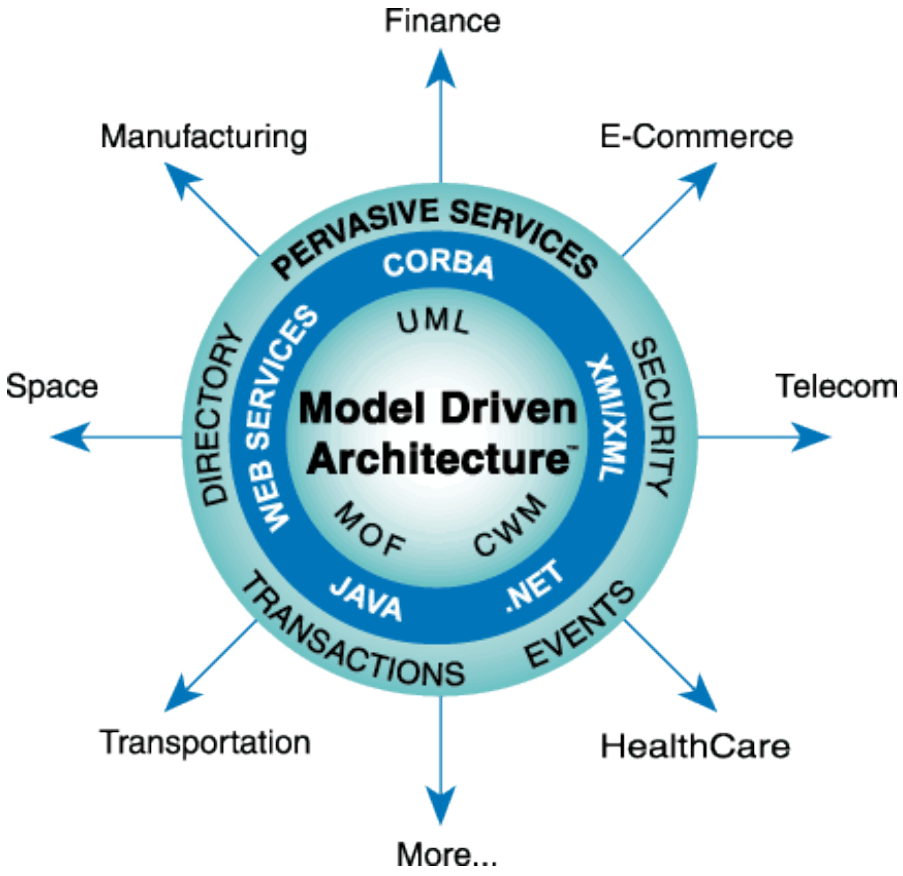


Models can be of many kinds:

- Parametrics for controllers
- Control diagrams
- Programs
- UML
- SysML

We all use model-driven development *today*.

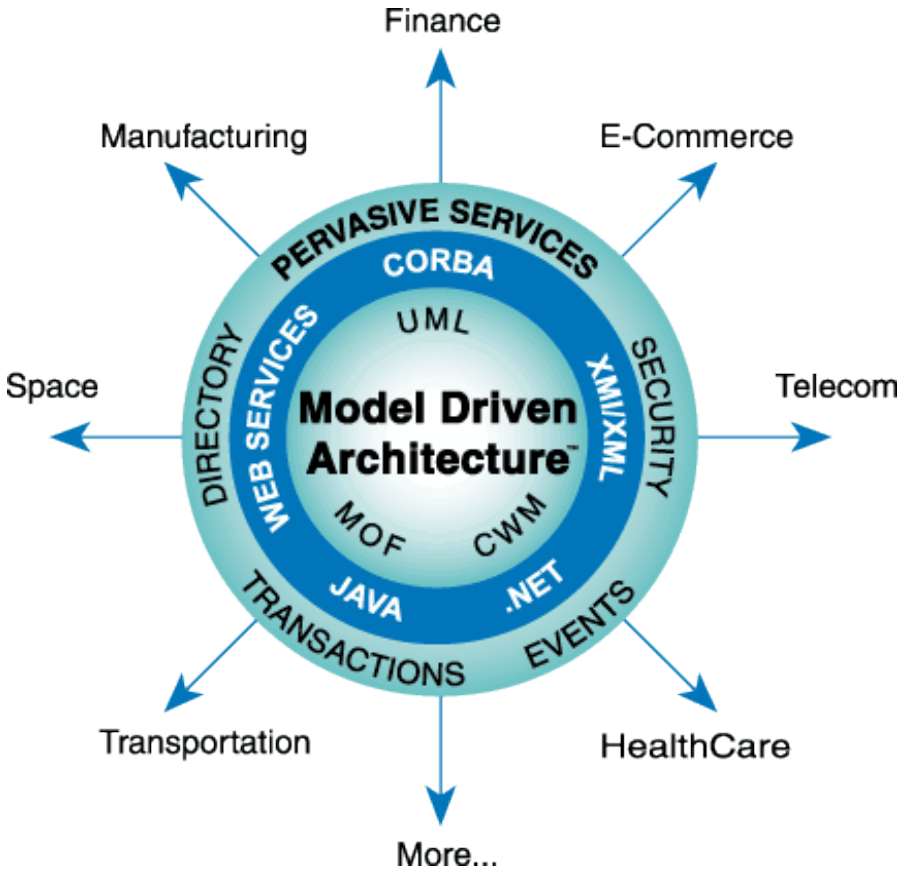
Model-Driven Architecture



® **OMG**

- An OMG initiative to develop standards based on the idea that modeling is a better foundation for developing and maintaining systems
- A brand for standards and products that adhere to those standards
- A set of technologies and techniques associated with those standards

Model-Driven Development/Engineering



® **OMG**

- An OMG initiative to develop standards based on the idea that modeling is a better foundation for developing and maintaining systems
- A brand for standards and products that adhere to those standards
- A set of technologies and techniques for transforming models

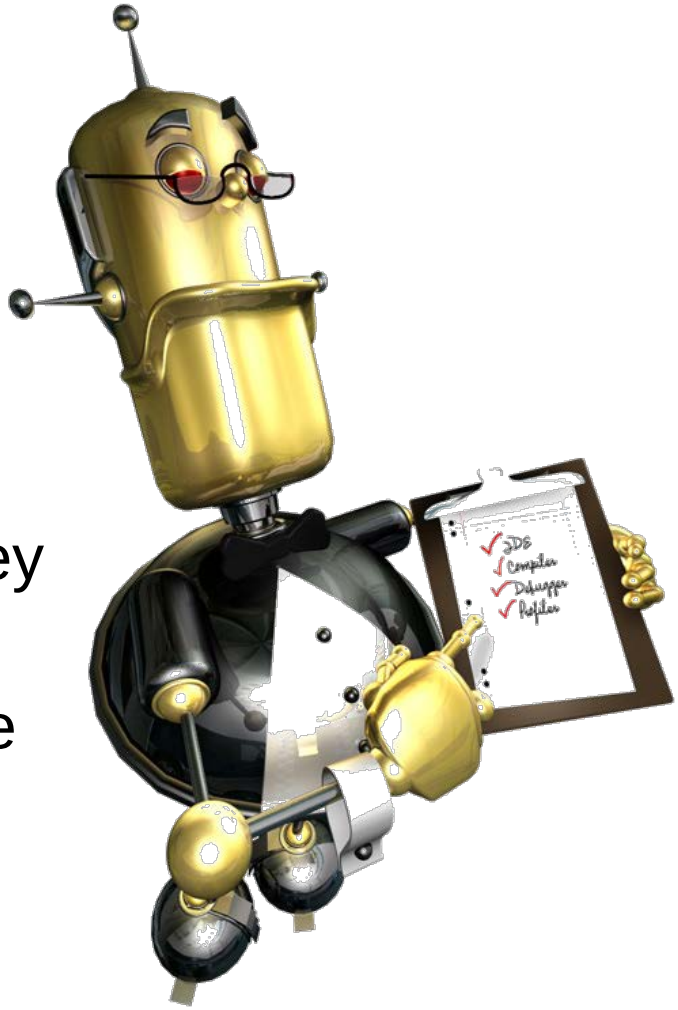
The Agile Critique

Models are bad (they say), because models:

- Don't run.
- Can't be tested automatically.

Models are bad (they say) because they are "documentation" which:

- Has no correspondence to the code
- Is extra work to build…
- …and maintain (or throw away)

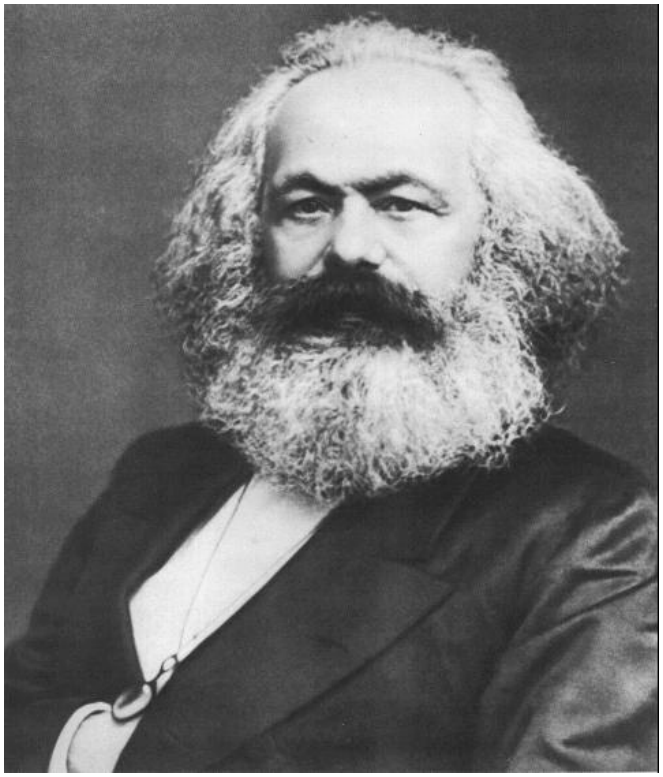


Agile Manifesto

"We are uncovering better ways of developing software by doing it and helping others do it.

We value:

- *Individuals and interactions* over processes and tools
- *Working software* over comprehensive documentation.
- *Customer collaboration* over contract negotiation.
- *Responding to change* over following a plan."



What problem is agility meant to solve?

- Delayed feedback on requirements
- Right solution to the wrong problem
- Unsustainable pace
- Poor customer relations
- Long time-to-market

What solution does agility propose?

- Immediate feedback by executing software
- Frequent releases with consequent feedback
- Timeboxed deliveries
- Customer on Site (aka Whole Team on Site)
- Incremental delivery of working code

Signatories to the Agile Manifesto

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Stephen Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

www.aanpo.org

But models are executable too!

Code is *so very* important because it runs, right?

Yes!

An executable model runs, so it can be verified, right?

Yes…

So, if a model is executed, it is as good as code, right?

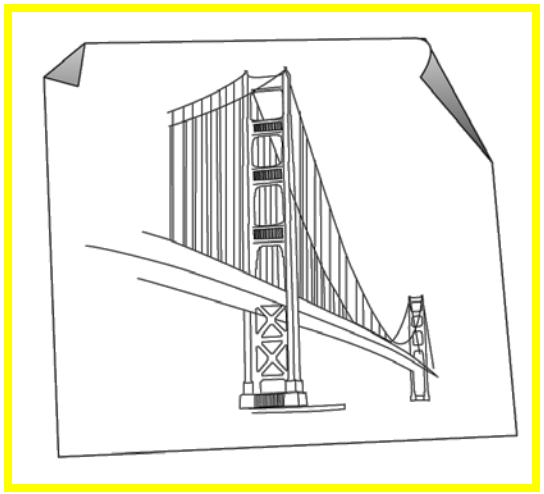
No. *Code* is the most important thing.

Argh!!!!!!!!

Models, Models, Models

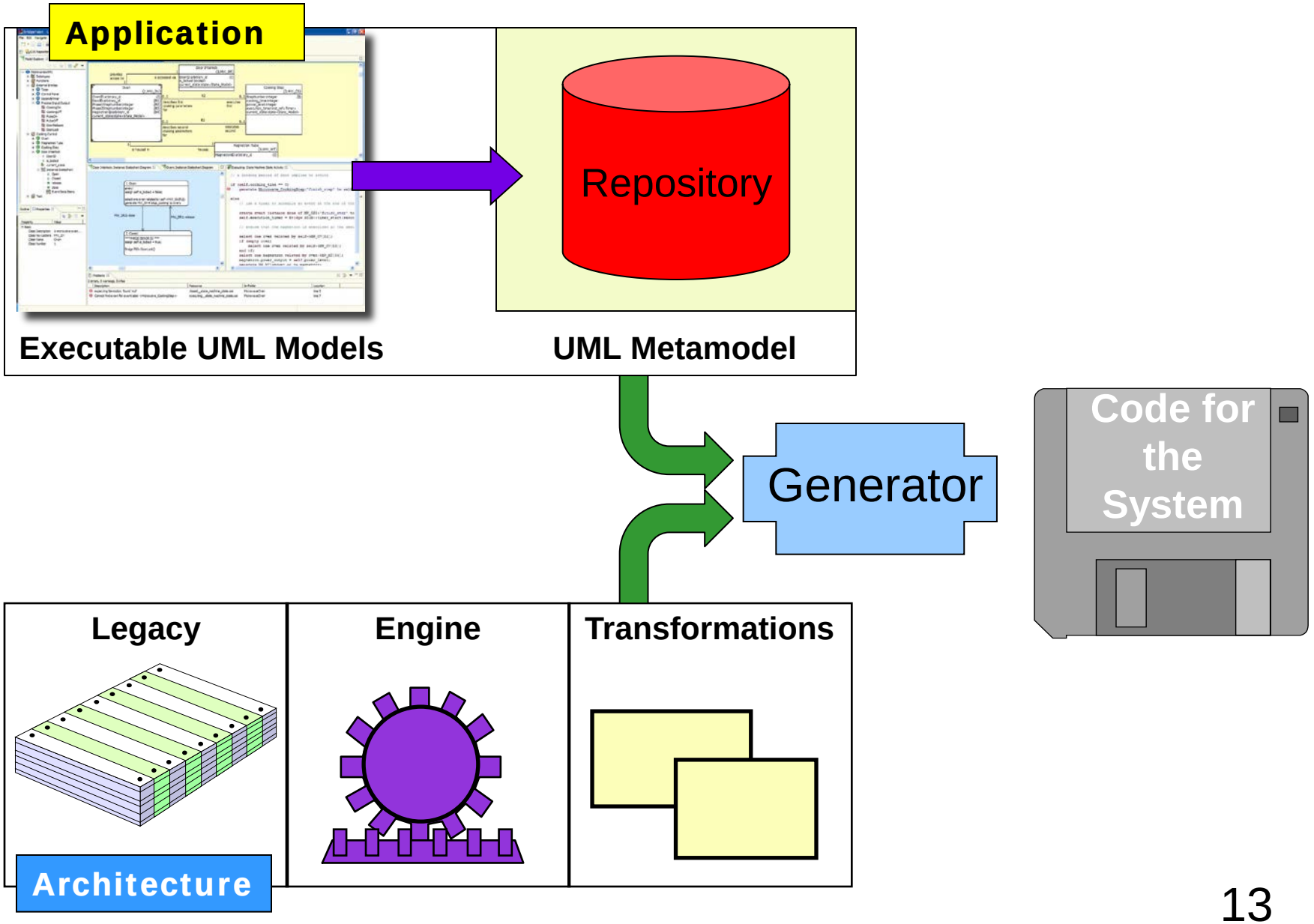
Models

Sketch



Blueprint

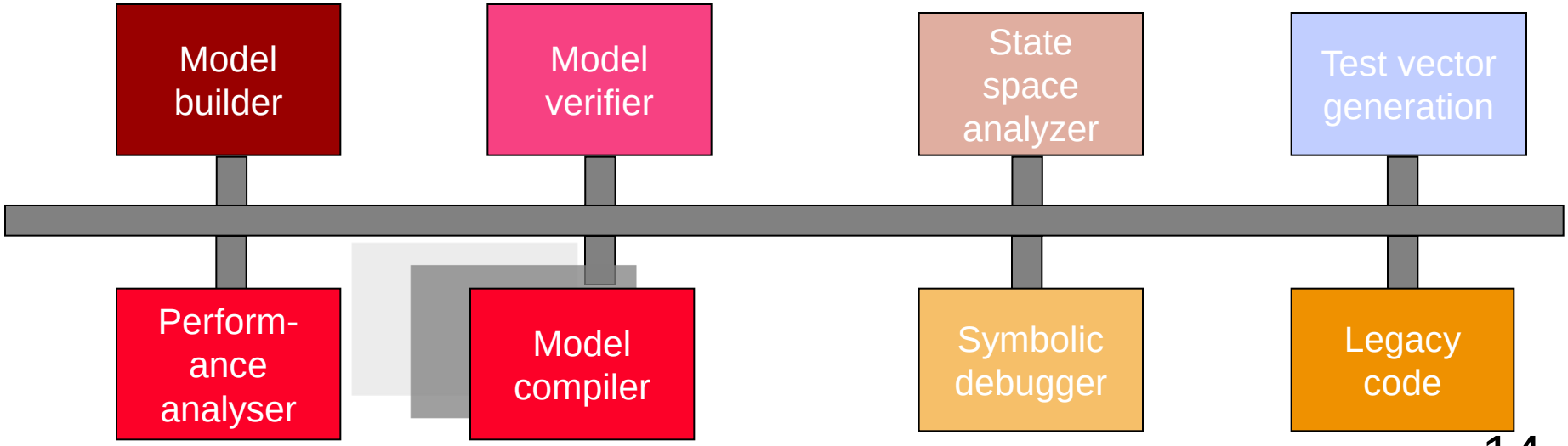
Executable



Executable UML Foundation

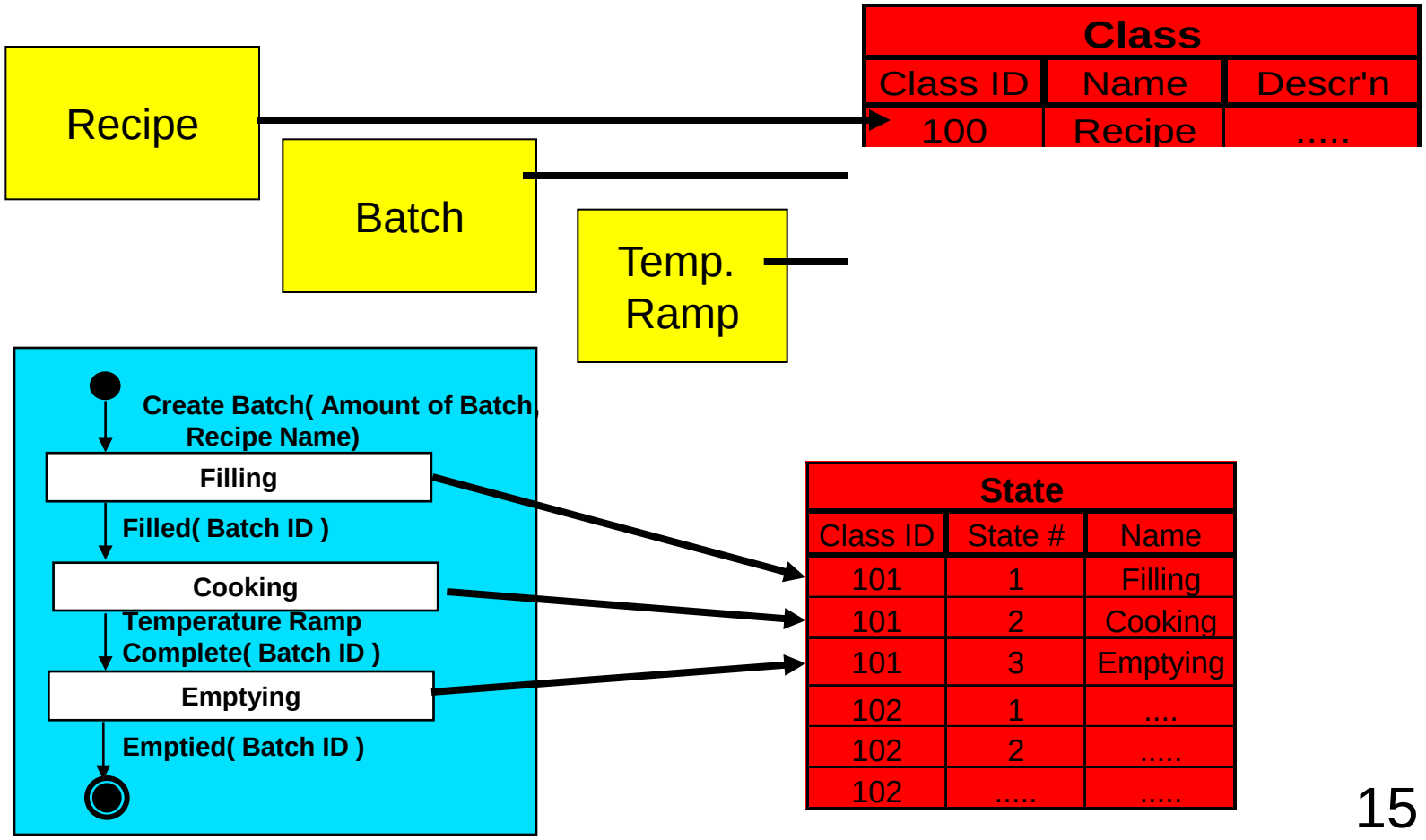
The Executable UML Foundation defines:

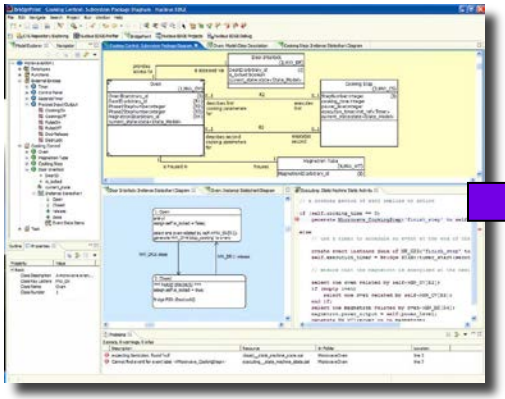
- An executable subset
- A definition of the execution semantics of that subset
- A base semantics



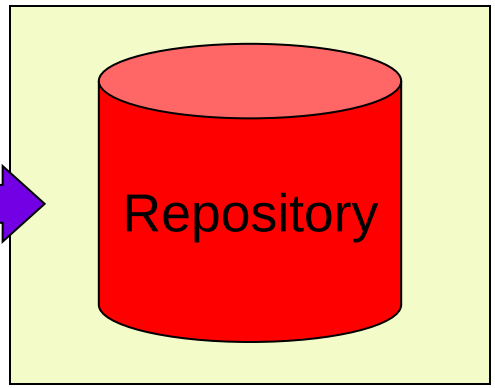
Model Capture

An application model is captured in a *metamodel*.

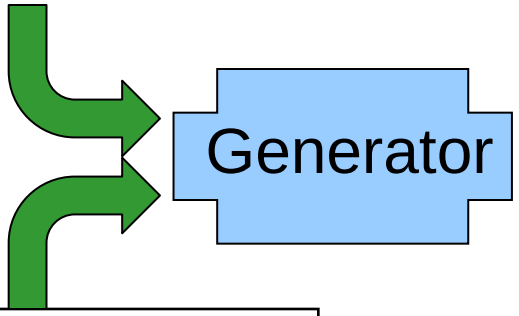


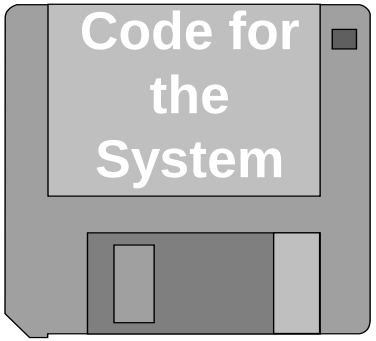


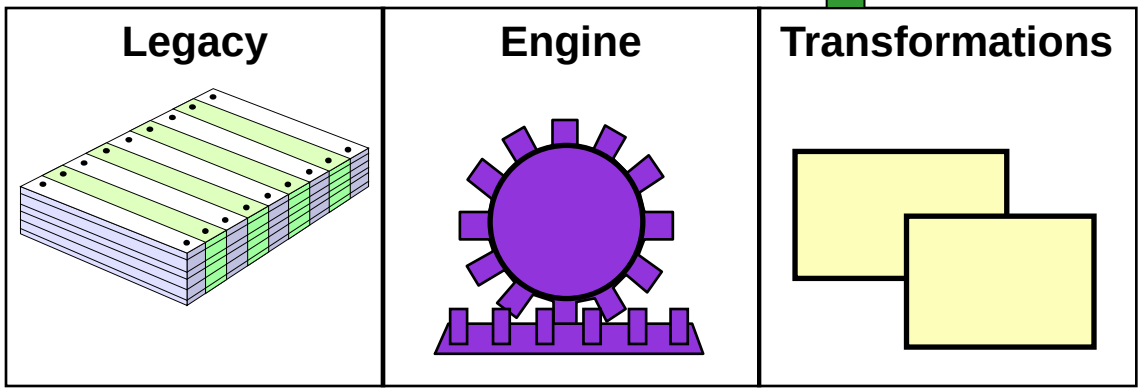
Executable UML Models



UML Metamodel



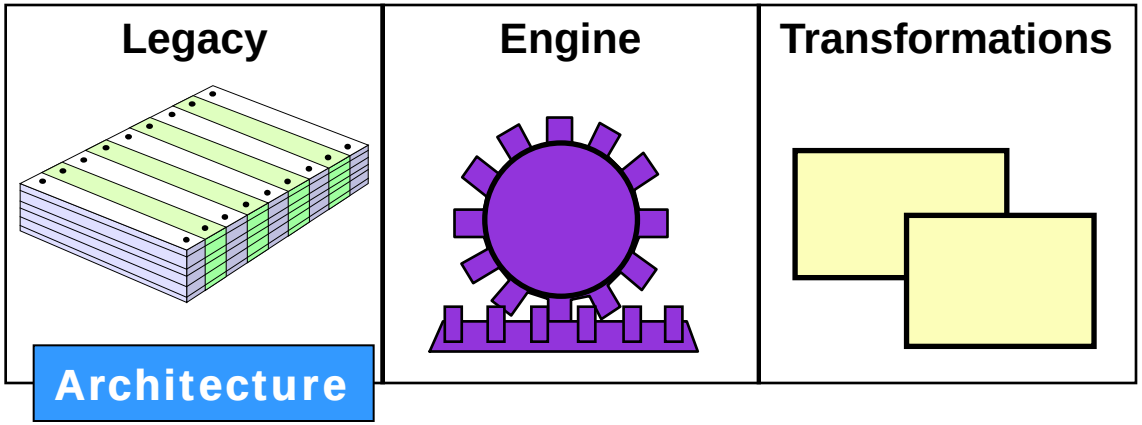




Software Architecture

An *application-independent* software architecture proclaims and enforces the organization of:

- data
- control
- structures
- time

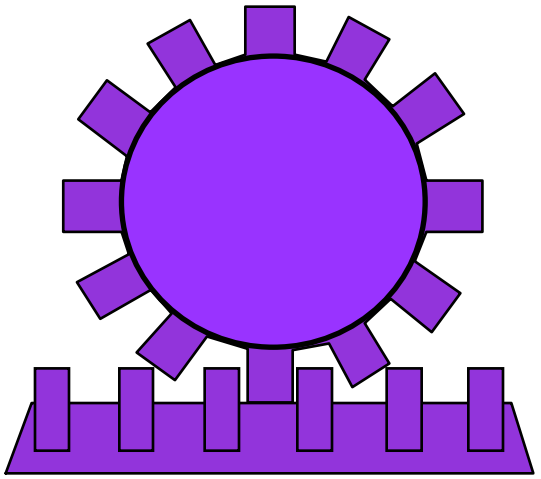


An architecture is realized as a *model compiler.*

Execution Engine

A limited set of reusable components sufficient to execute Executable UML.

- Data access
- State machines
- Activation of threads
- Order of execution



Execution engine

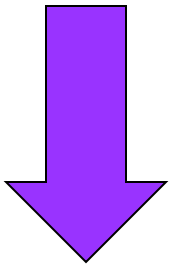
Transformations

A generator executes transformation rules against the populated metamodel.

Class		
Class ID	Name	Descr'n
100	Recipe	.....
101	Batch	.....
102	Temp Ramp	.....

- Read the repository
- Make substitutions

```
[template classToStruct( s: Class)]
Struct [s.name/] { … } ;
[/template]
```

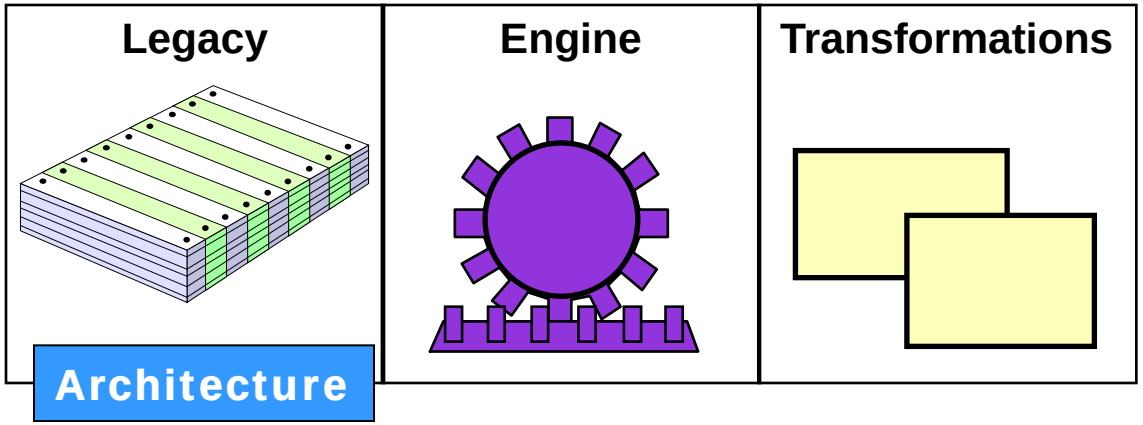


```
Struct Recipe { …};
Struct Batch { … };
Struct TempRamp { … };
```

- Generate text

Model Compilers

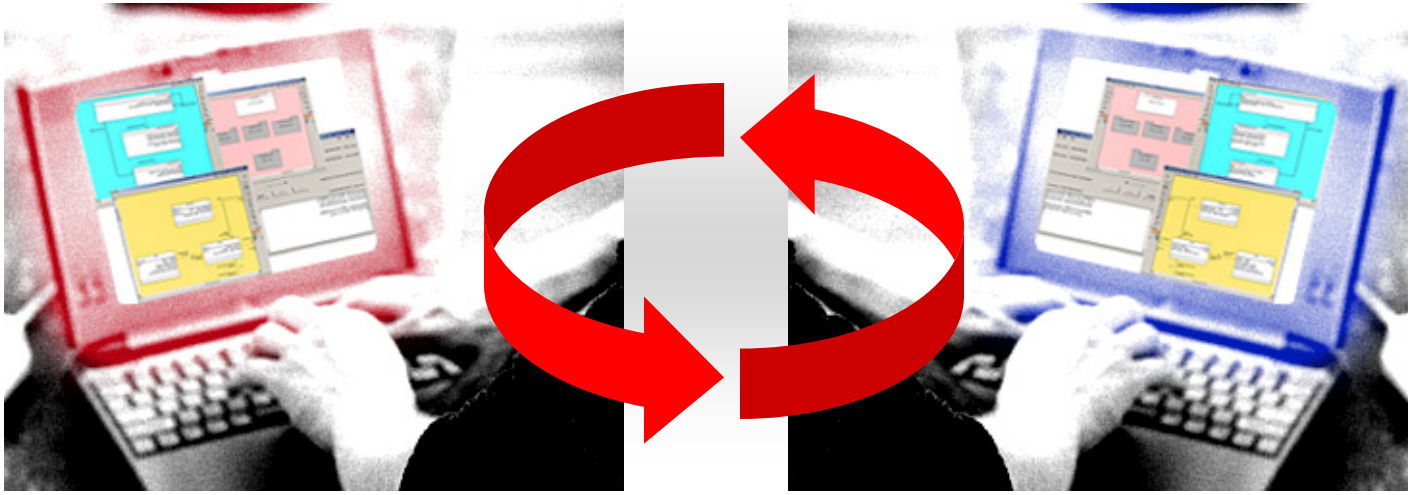
- Build a complete system from models *consistently*
- Translate into the selected architecture for the system



- Leverages expertise of best architects,
- Captures that expertise.
- Reuses expertise across many applications
- Maintains application and architecture separately

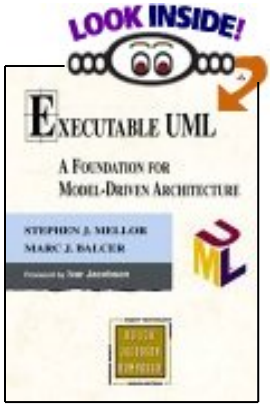
How can modeling be agile?

- Immediate feedback by executing *models**
- Frequent releases with consequent feedback
- Timeboxed deliveries
- Customer on Site (aka Whole Team on Site)
- Incremental delivery of working *models*



* The Agile Manifesto uses "software," not "code."

Want to learn more?



Executable UML: A Foundation for Model-Driven Architecture, Stephen J. Mellor, Marc Balcer

Comprehensive language introduction and reference

OBJECT MANAGEMENT GROUP

Semantics Of A Foundational Subset For Executable UML Models (fUML)

www.omg.org/spec/FUML

An Open-Source Reference Implementation for a Foundational Subset fro Executable Models

Model Driven Solutions under contract to Lockheed Martin Corporation

fuml.modeldriven.org

Concrete Syntax for a UML Action Language: Action Language For Foundational UML (ALF)

omg.org/spec/ALF/1.0.1/Beta3/PDF/

Thank you

StephenMellor@StephenMellor.com