

Retaking Rules for Developers

Ryan Brush
@ryanbrush

Strangeloop 2014

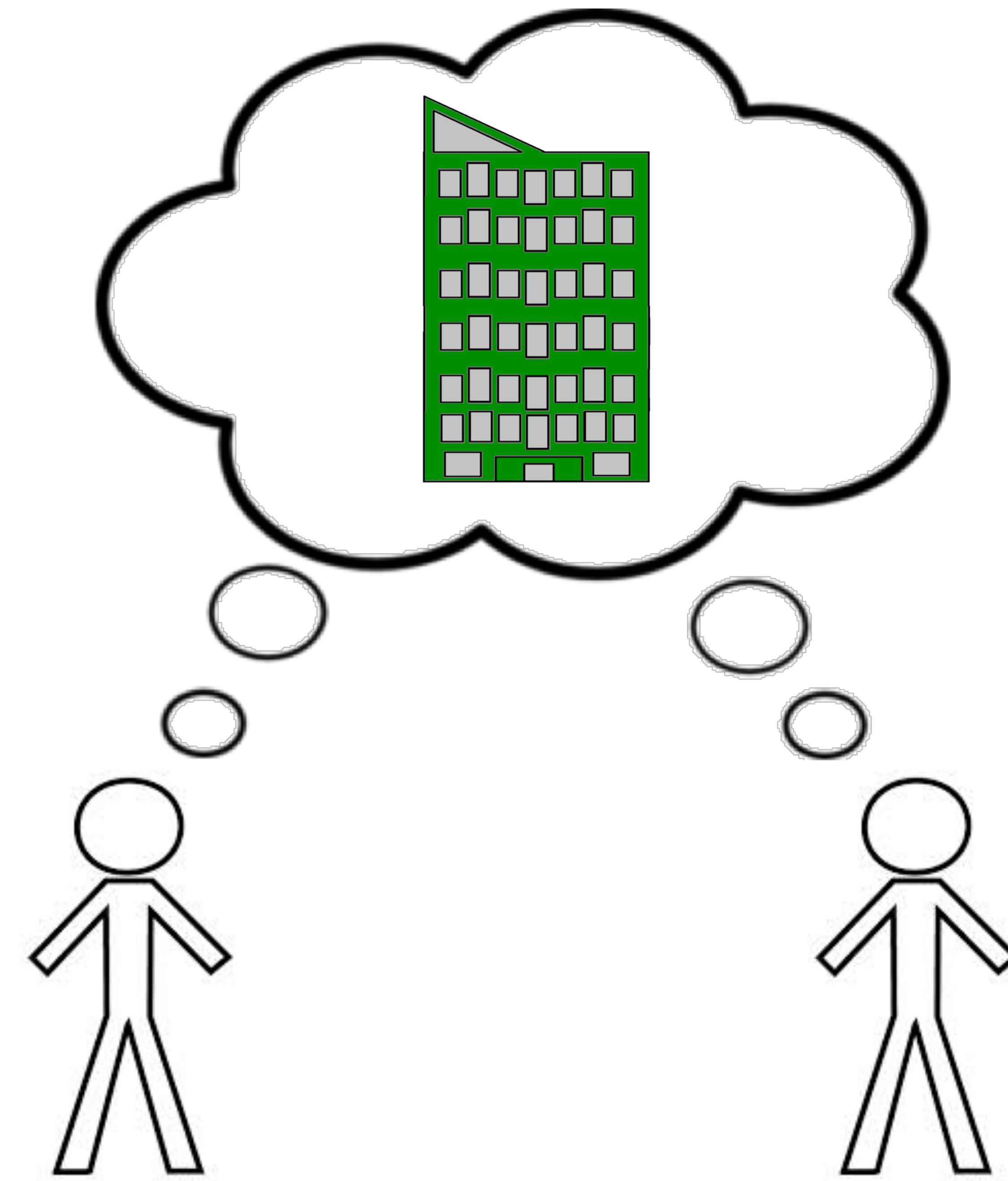
1977





Intelligent systems derive their power
from the **knowledge they possess** rather
than the specific formalisms used.

-Edward Feigenbaum

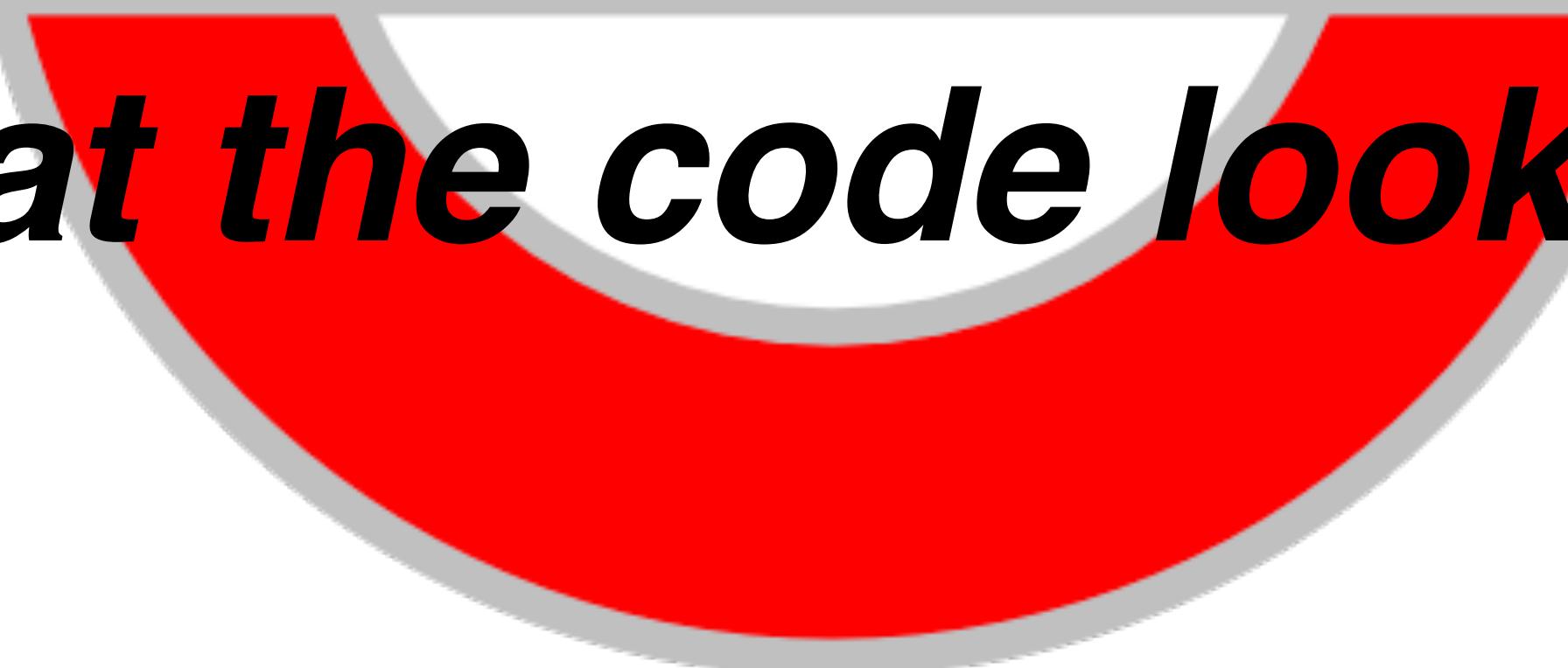




Domain knowledge



MIND THE GAP



What the code looks like

Composable units of knowledge

Hypertension (severity == moderate)

→ recommend-diet-changes

Hypertension (severity == moderate)

Demographics (age > 60)

→ prescribe-bp-meds

1979



**On the Efficient Implementation
of Production Systems**

Charles L. Forgy

February, 1979

**Carnegie-Mellon University
Department of Computer Science**

Hypertension (severity == moderate)

→ recommend-diet-changes

Hypertension (severity == moderate)

Demographics (age > 60)

→ prescribe-bp-meds

Hypertension

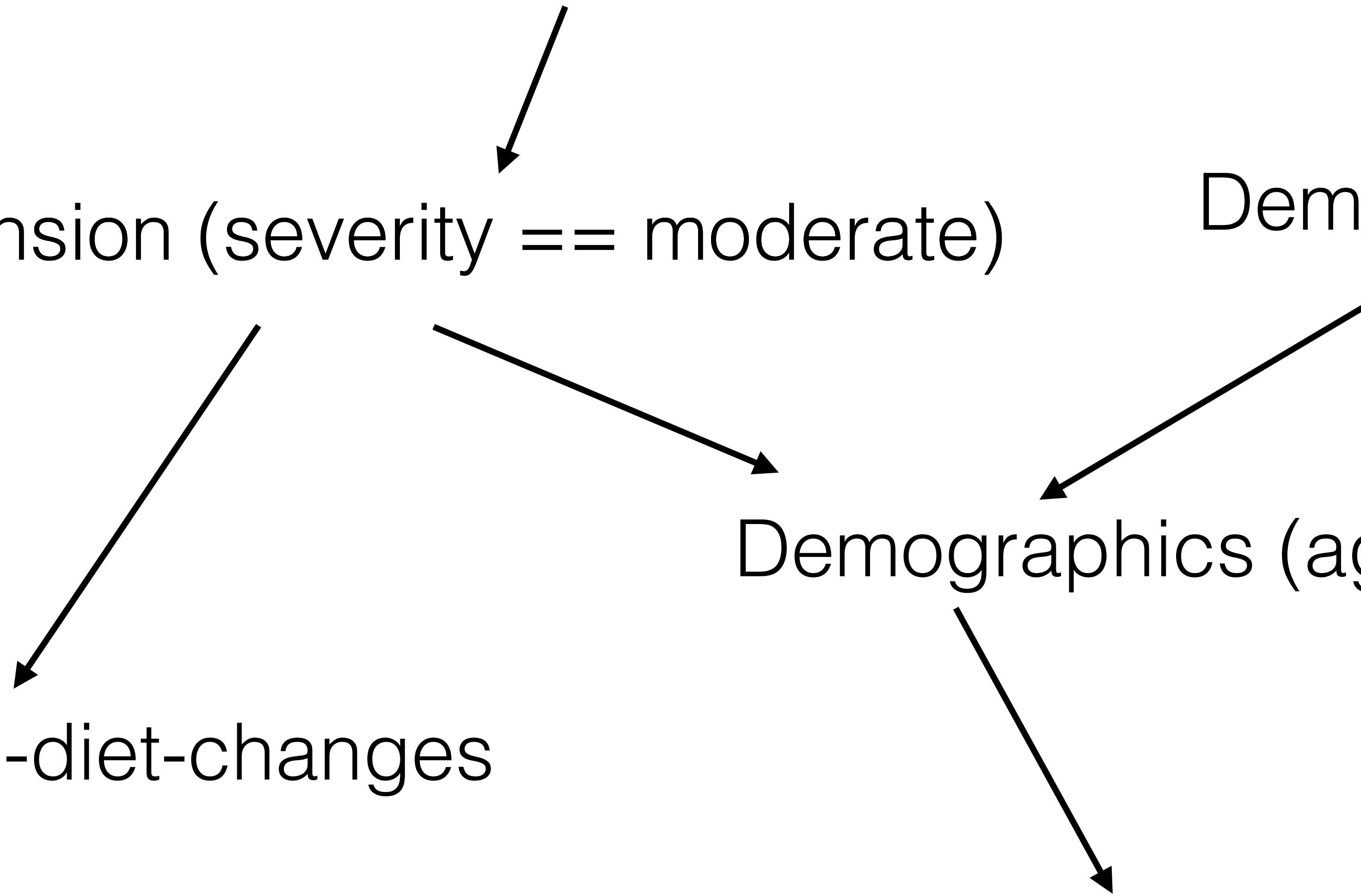
Hypertension (severity == moderate)

Demographics

recommend-diet-changes

Demographics (age > 60)

prescribe-bp-meds



1985



1985



1988



1988

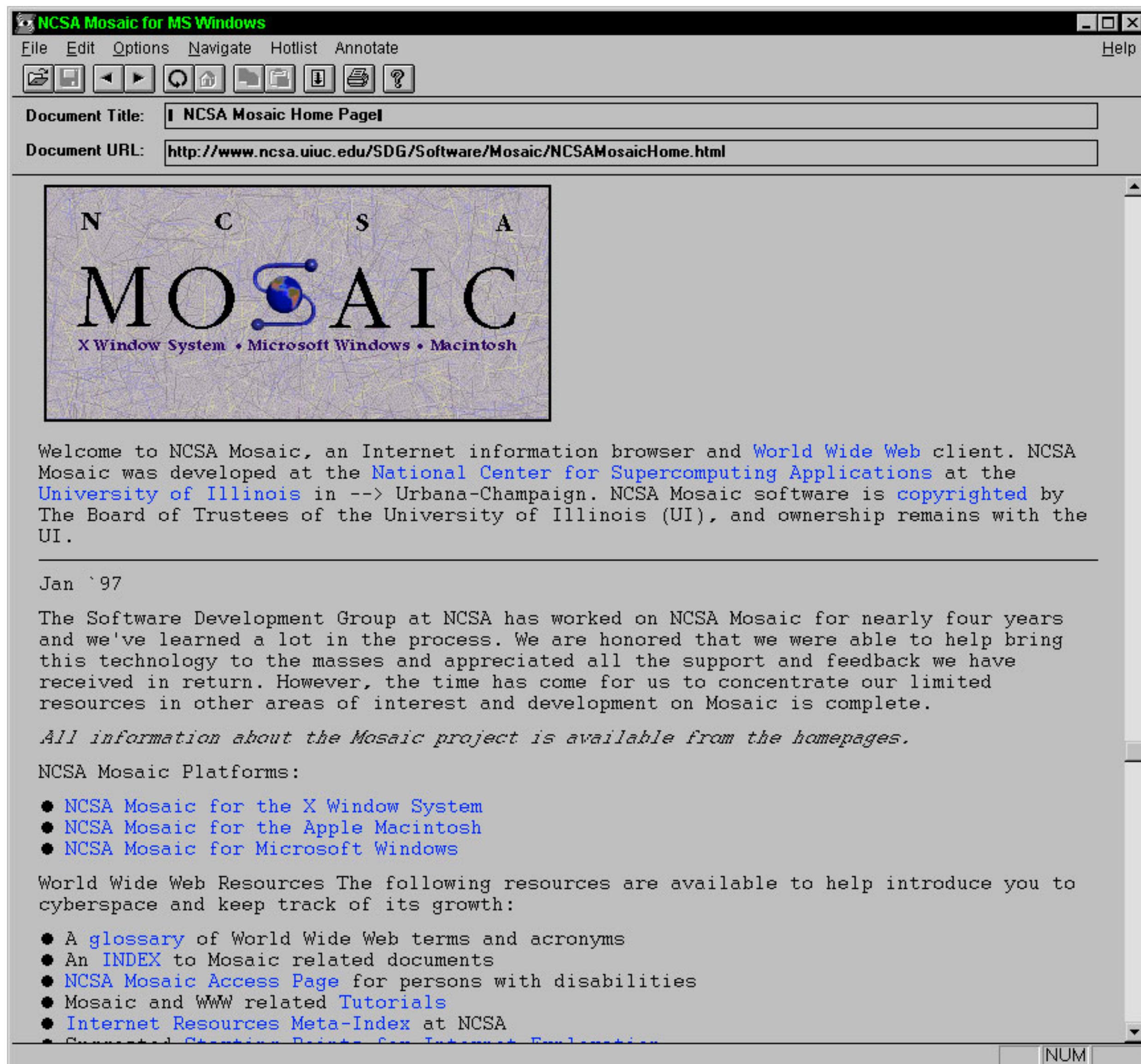


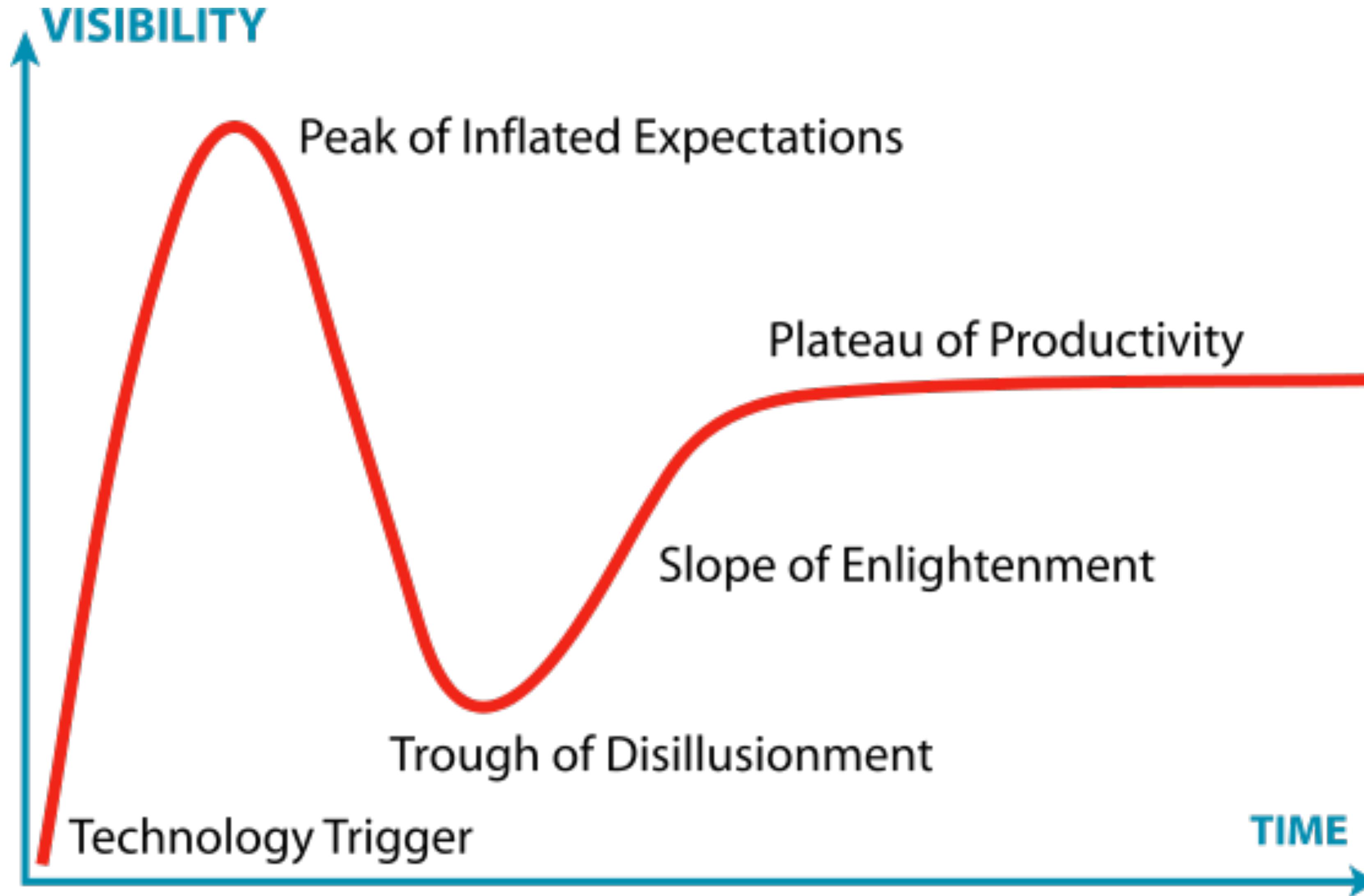
“Frankly, there were a lot of intellectually dishonest promises made.”

-Peter Gabel
Bring in the Expert
InfoWorld, October 1990

The AI Winter

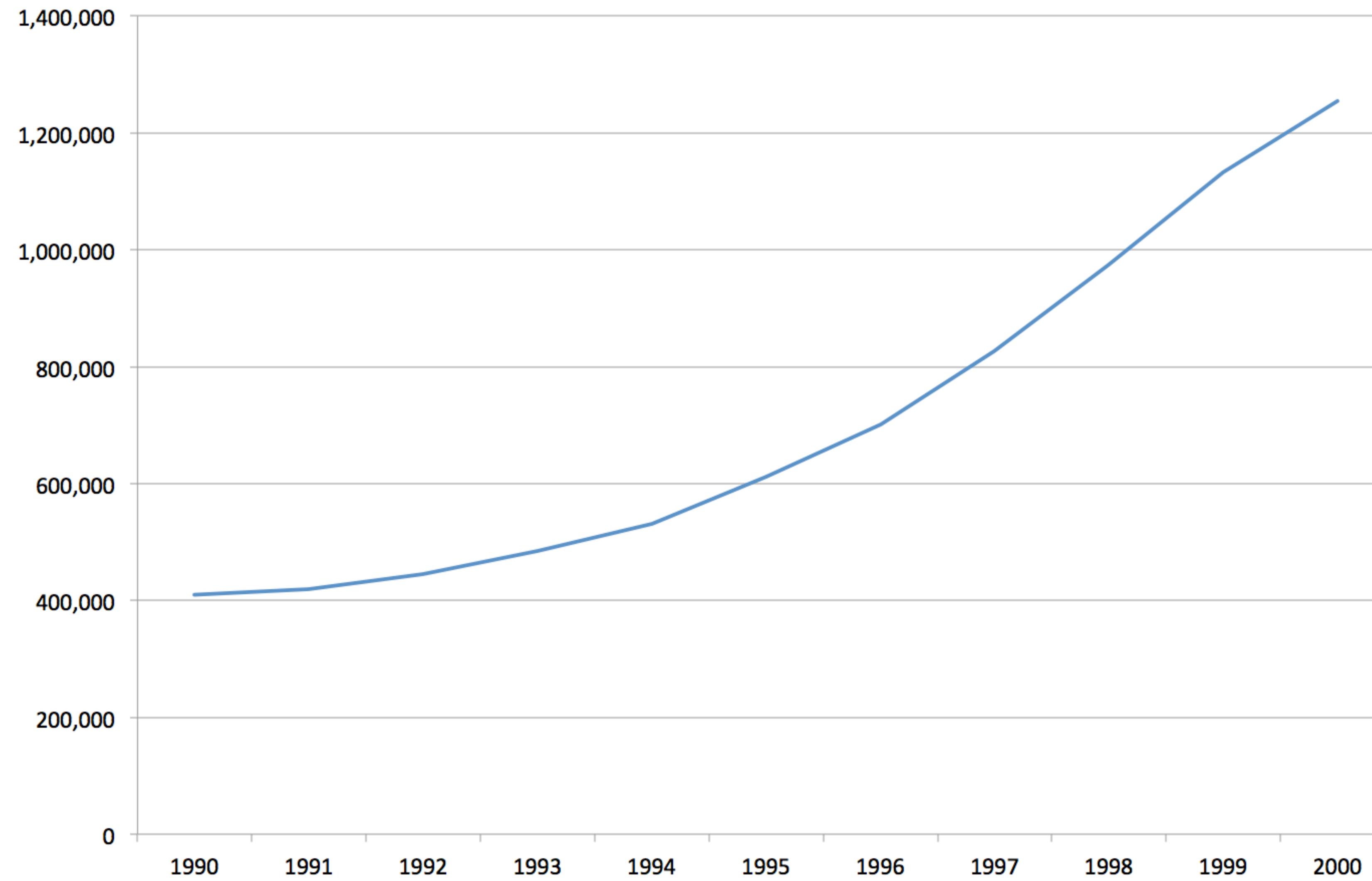
1990s





A Wrong Turn

1990s IT Employment Level



Wanted:
A system that can produce
code without requiring coders

rule “My Rule”

when: This

then: That

Often the central pitch for a rules engine
is that it will allow the **business people**
to specify the rules themselves,
so they can build the rules
without involving programmers.

-Martin Fowler, 2009

Tools Target Business Users

Limited mini languages

Decision tables in spreadsheets

Tools Target Business Users

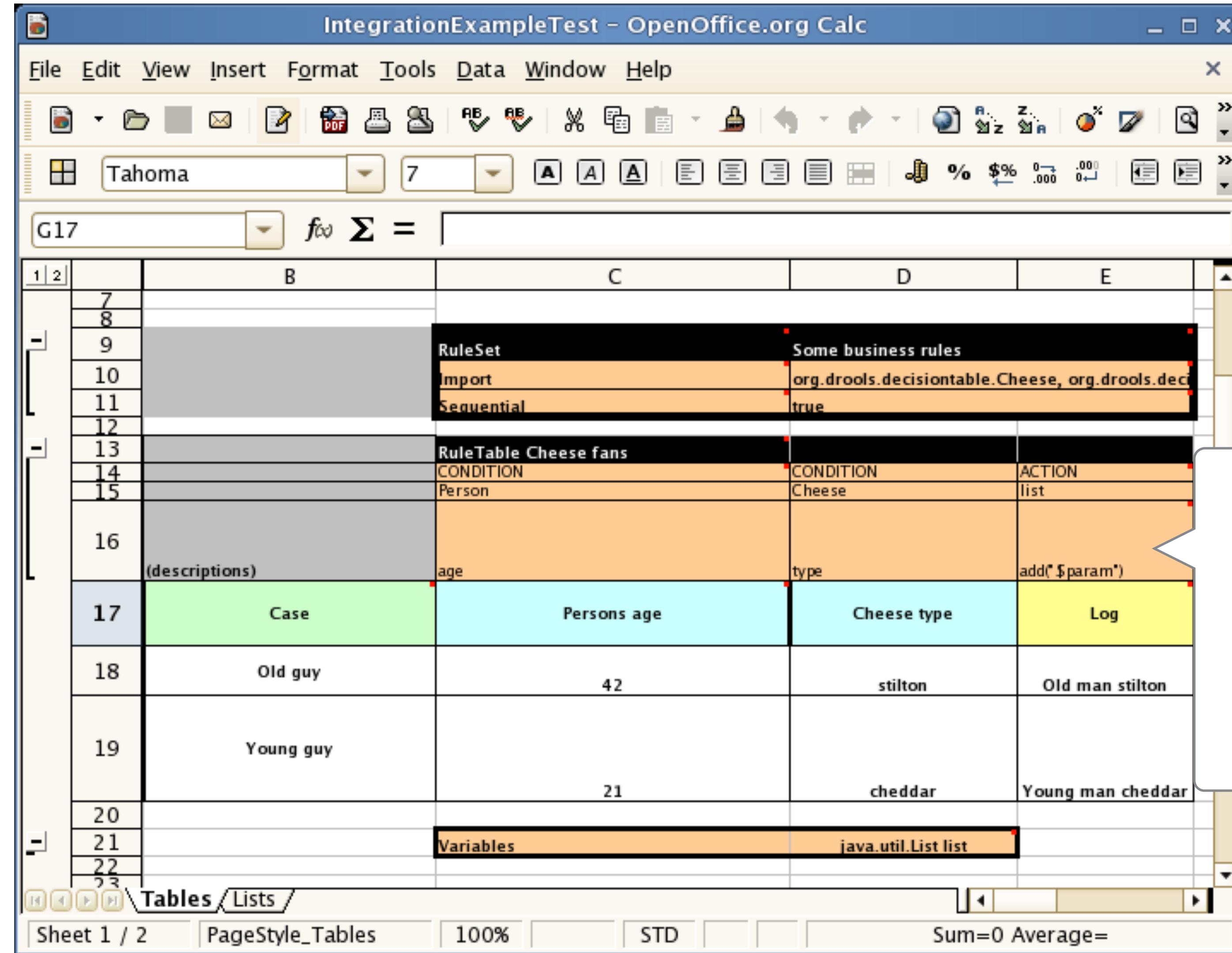
Great for simple problems

But problems don't stay simple

Tools Target Business Users

```
rule "My Rule"  
when:  
    // Pile of Java code  
then:  
    // Another pile of Java code
```

Tools Target Business Users



Write
Code
Here

Rule authoring is development,
whether we call it that or not

So let's approach it as code

A

B

f(A,B) => C

g(B,C) => D

h(A,C,D) => E

A

B

f(A,B) => C

g(B,C) => D

h(A,C,D) => E

A = Travel Expenses

B = Contract Budget

C = Over Budget Expenses

D = Required justifications

E = Expense report summary

A

B

$f(A,B) \Rightarrow C$

$g(B,C) \Rightarrow D$

$h(A,C,D) \Rightarrow E$

A = Customer Demographics

B = Order History

C = VIP status

D = Current Offerings

E = Sales Package

A

B

$f(A,B) \Rightarrow C$

$g(B,C) \Rightarrow D$

$h(A,C,D) \Rightarrow E$

A = Diagnosis history

B = Lab test results

C = Complication risks

D = Interventions

E = Plan of Care

A

B

f(A,B) => C

...or many other domains

g(B,C) => D

h(A,C,D) => E

A

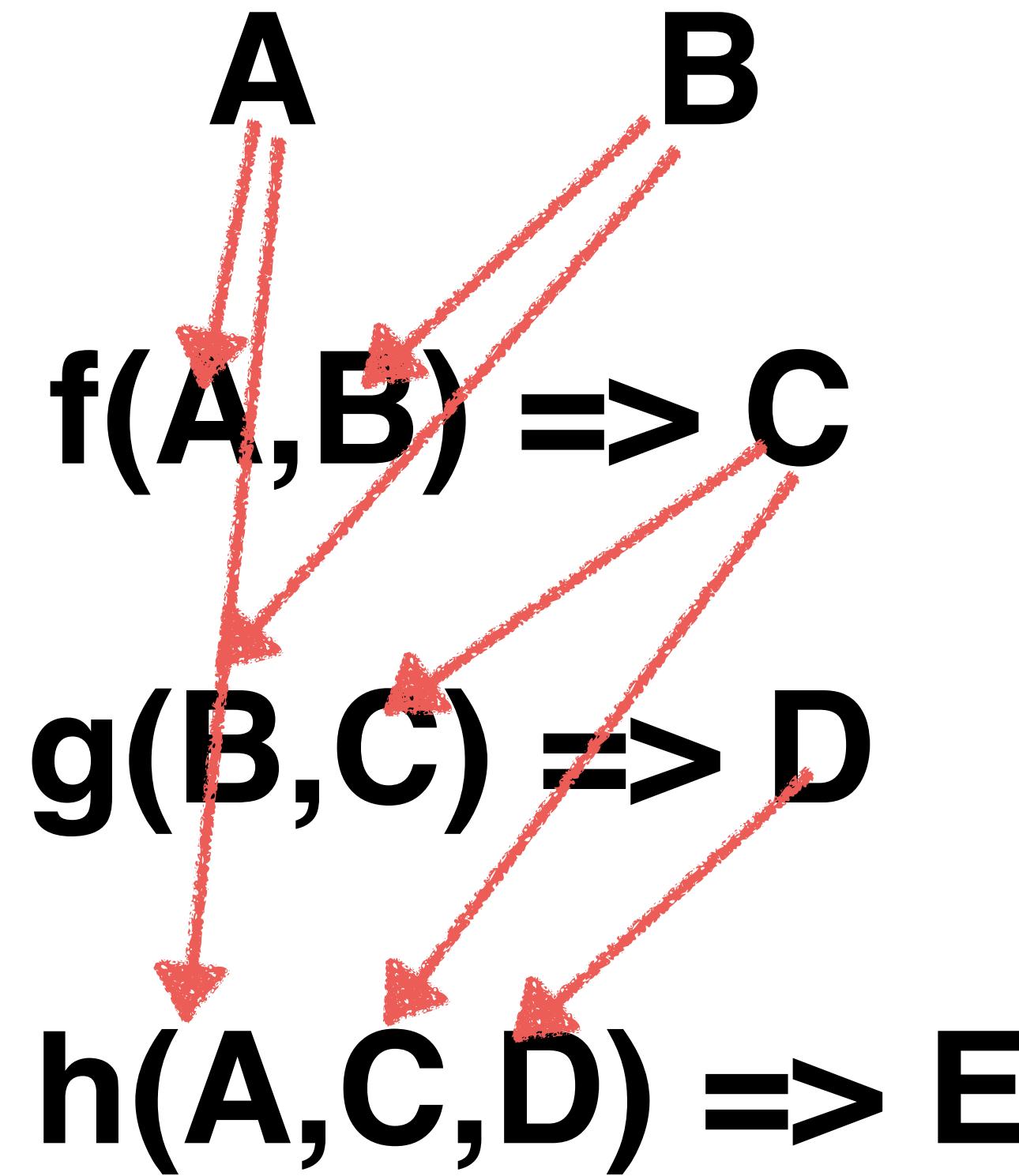
B

$f(A,B) \Rightarrow C$

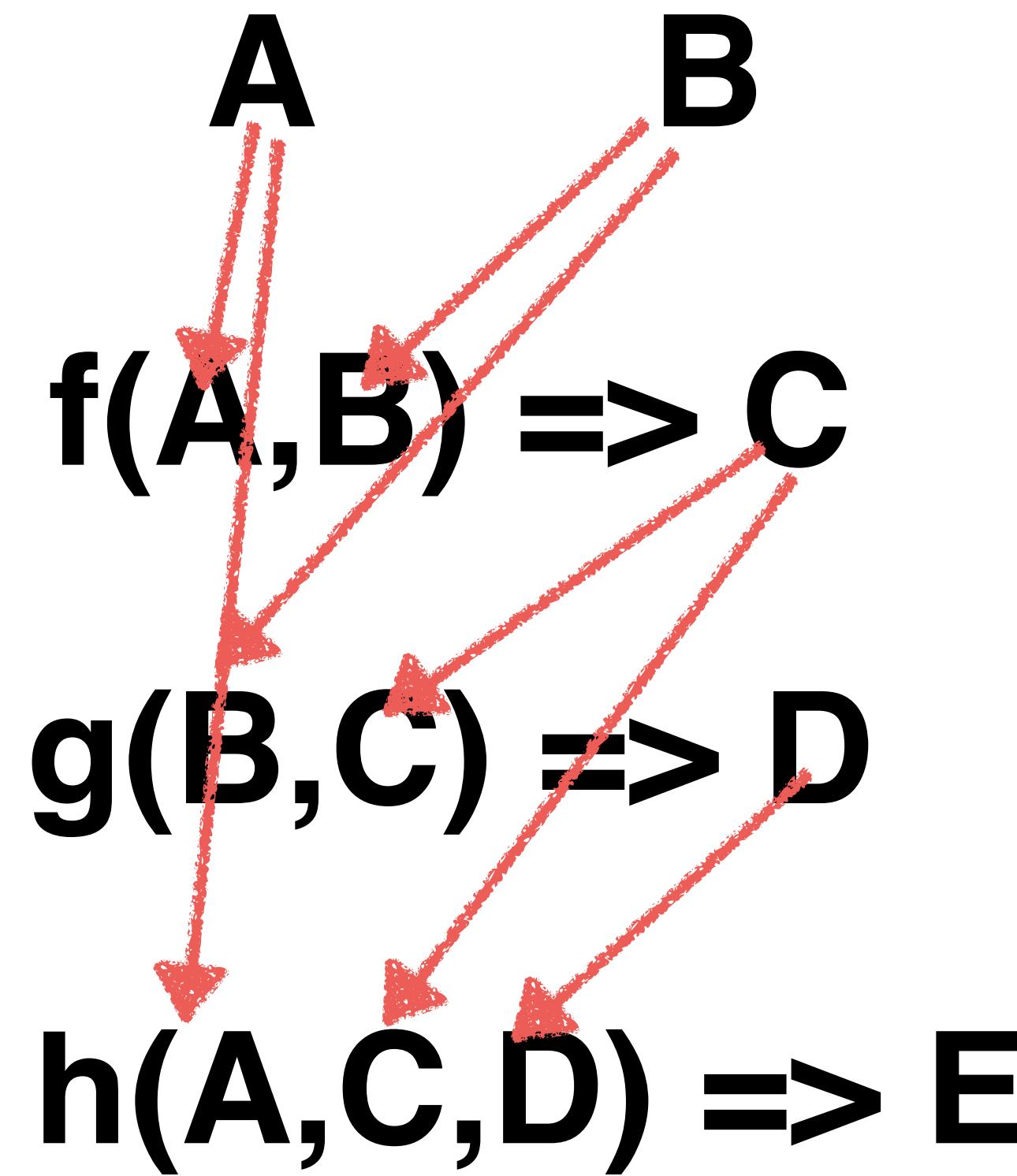
$g(B,C) \Rightarrow D$

$h(A,C,D) \Rightarrow E$

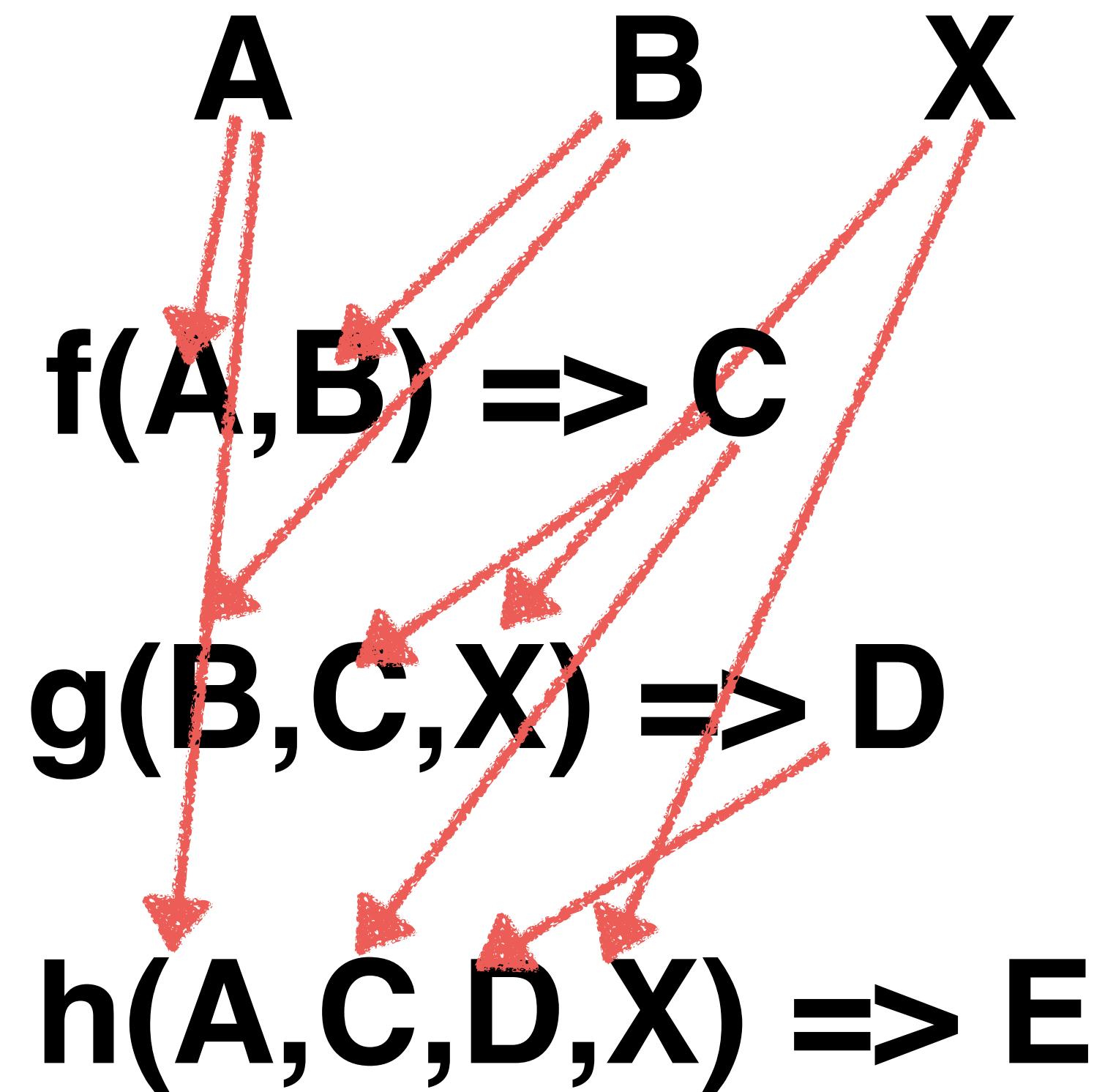
But we must wire things together!



But we must wire things together!



A = Diagnosis history
B = Lab test results
C = Complication risks
D = Interventions
E = Plan of Care



A = Diagnosis history
B = Lab test results
C = Complication risks
D = Interventions
E = Plan of Care
X = Family History

Now try this with thousands of requirements from different sources

Explicit wiring is complex and error prone

So let's wire them automatically

A

B

f(A,B) => C

g(B,C) => D

h(A,C,D) => E

How do we automatically wire this?

A

B

f(A,B) => C

g(B,C) => D

h(A,C,D) => E

unit-of-logic

Type[constraints]

Type[constraints]

=>

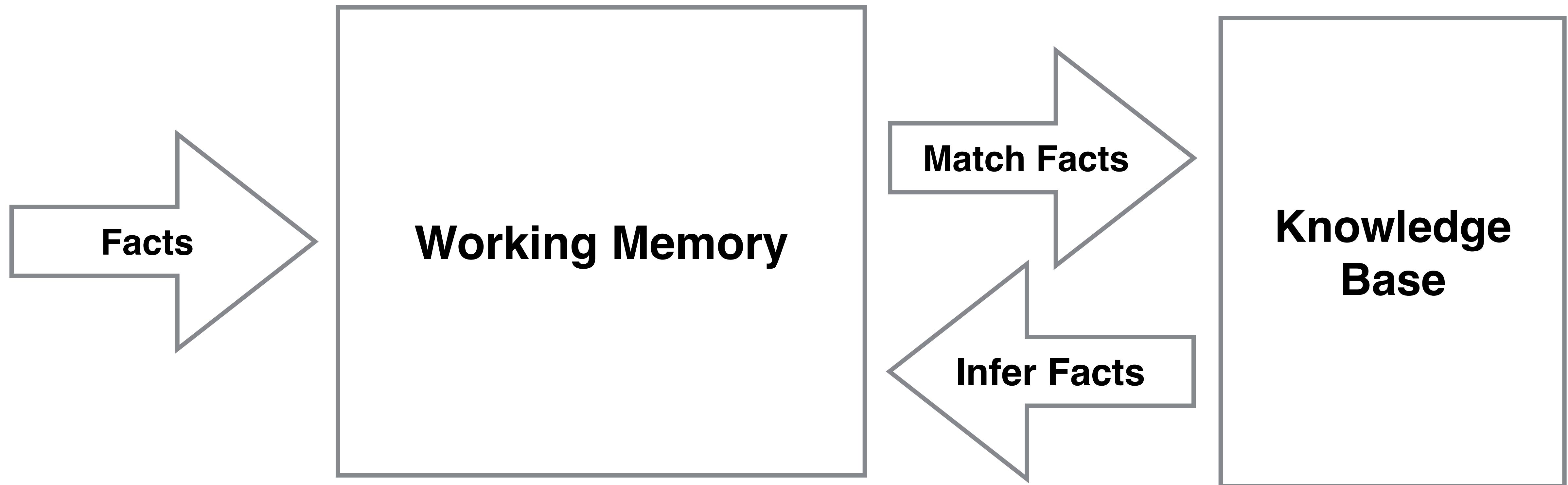
action

Rules as a Control Structure

Rules as a Control Structure

A Principled Design

Start with a proven model



Simple structure

unit-of-logic

Type[constraints]

Type[constraints]

=>

action

Type system of the host

unit-of-logic

Type[constraints]

Type[constraints]

=>

action

Simple expressions

unit-of-logic

Type[**constraints**]

Type[**constraints**]

=>

action

Unification

unit-of-logic

Type[**?a == field1**]

Type[**?a == field2**]

=>

action

Arbitrary actions

unit-of-logic

Type[constraints]

Type[constraints]

=>

action

Truth Maintenance

if A assert B

if ~~X~~ retract B

Rules as a first-class participant
in development

Nools (JavaScript)

Wongi (Ruby)

Clara (Clojure)

Nools (JavaScript)

Wongi (Ruby)

Clara (Clojure)

Why Clojure?

Data over objects

Equality means something

Reach

Can grow the language

(Hello, macros!)

```
(defrule diet-recommendation  
  [Hypertension (= :moderate severity)]  
  =>  
  (recommend-diet-changes))
```

```
(defrule senior-bp-management  
  [Hypertension (= :moderate severity)]  
  [Demographics (> age 60)]  
  =>  
  (prescribe-bp-meds))
```

Hypertension

[Hypertension (= :moderate severity)]

(recommend-diet-changes))

Demographics

[Demographics (> age 60)]

(prescribe-bp-meds))



Rules Demo

And the usuals..

Arbitrary expressions

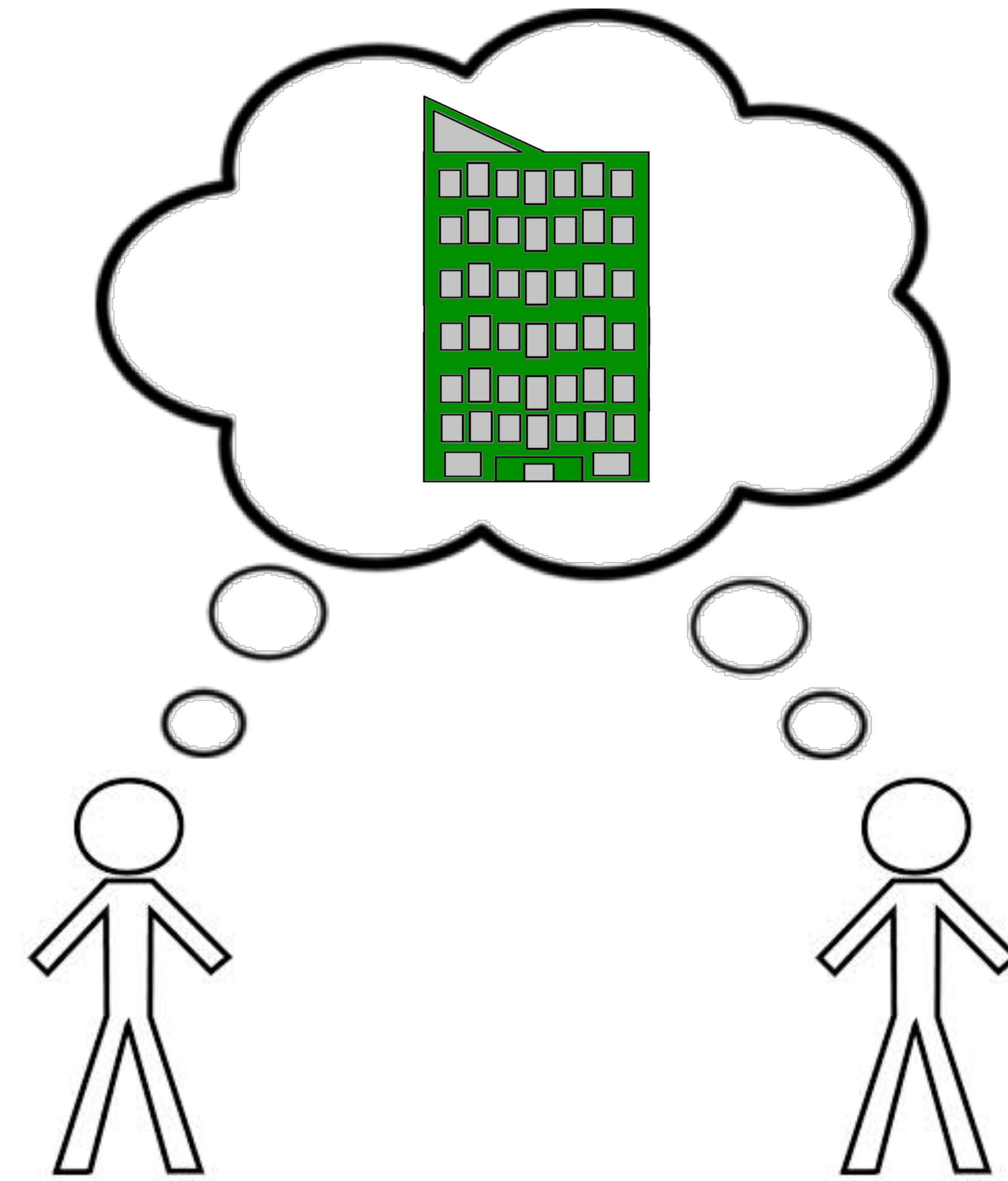
Accumulators

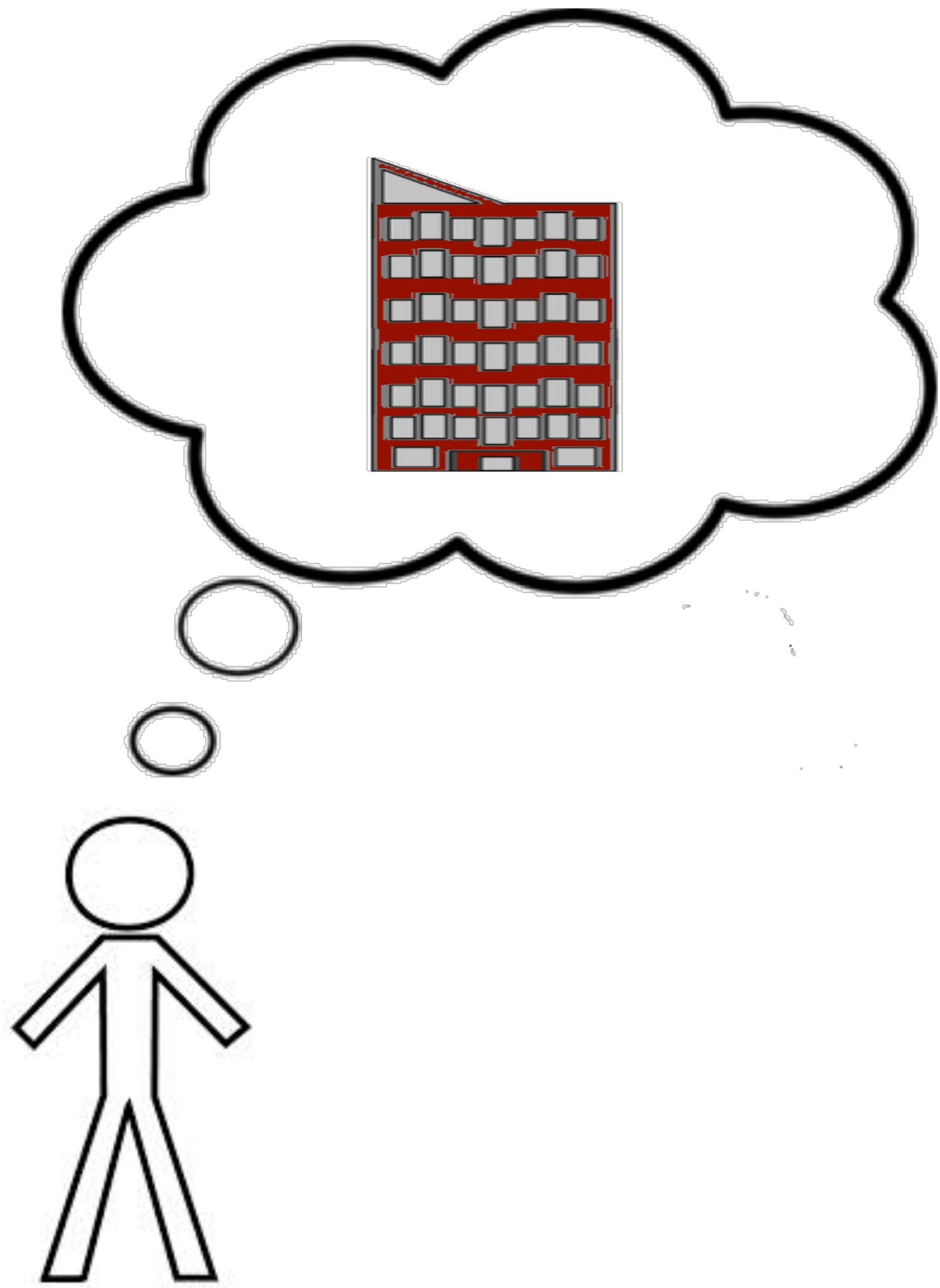
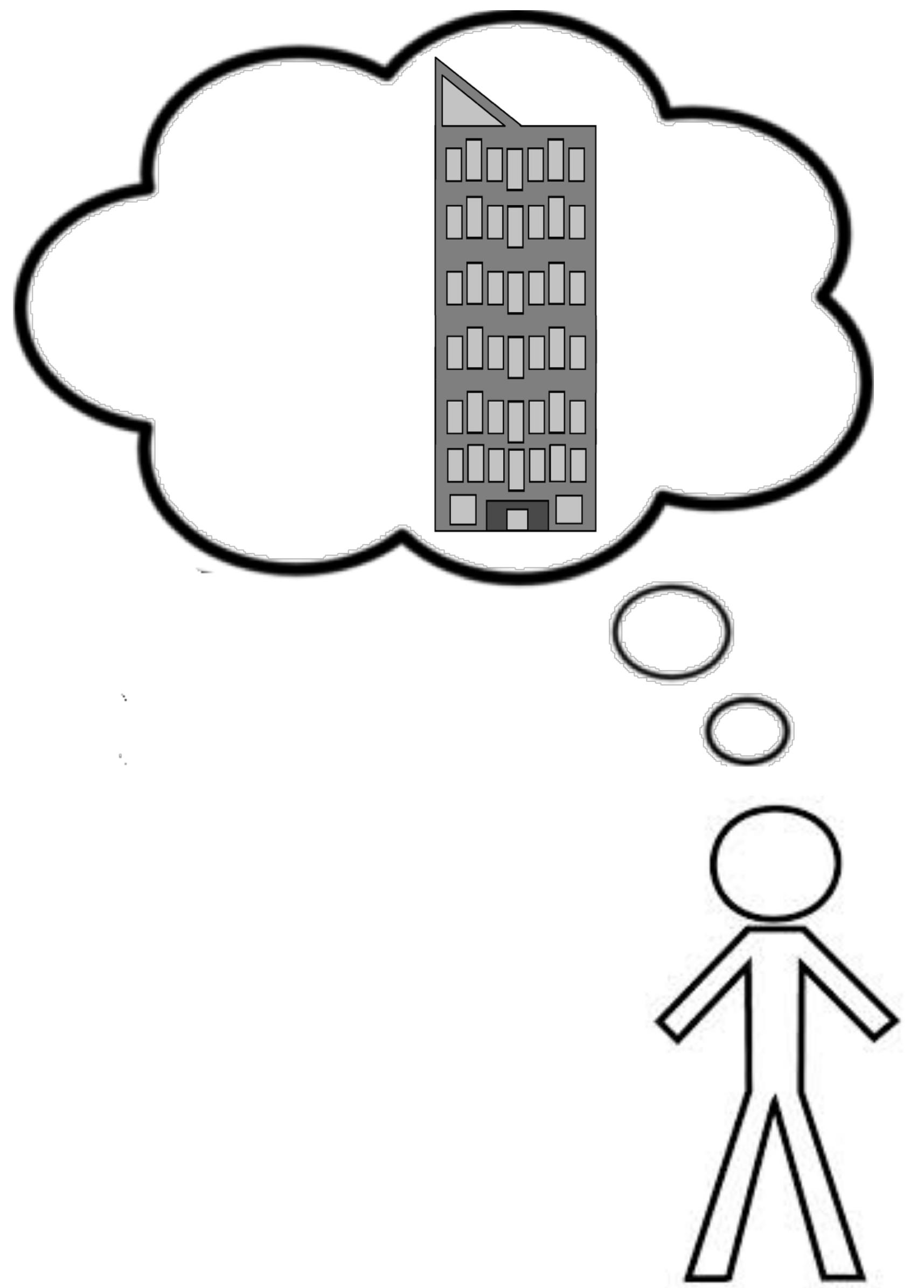
Session durability

Tracing

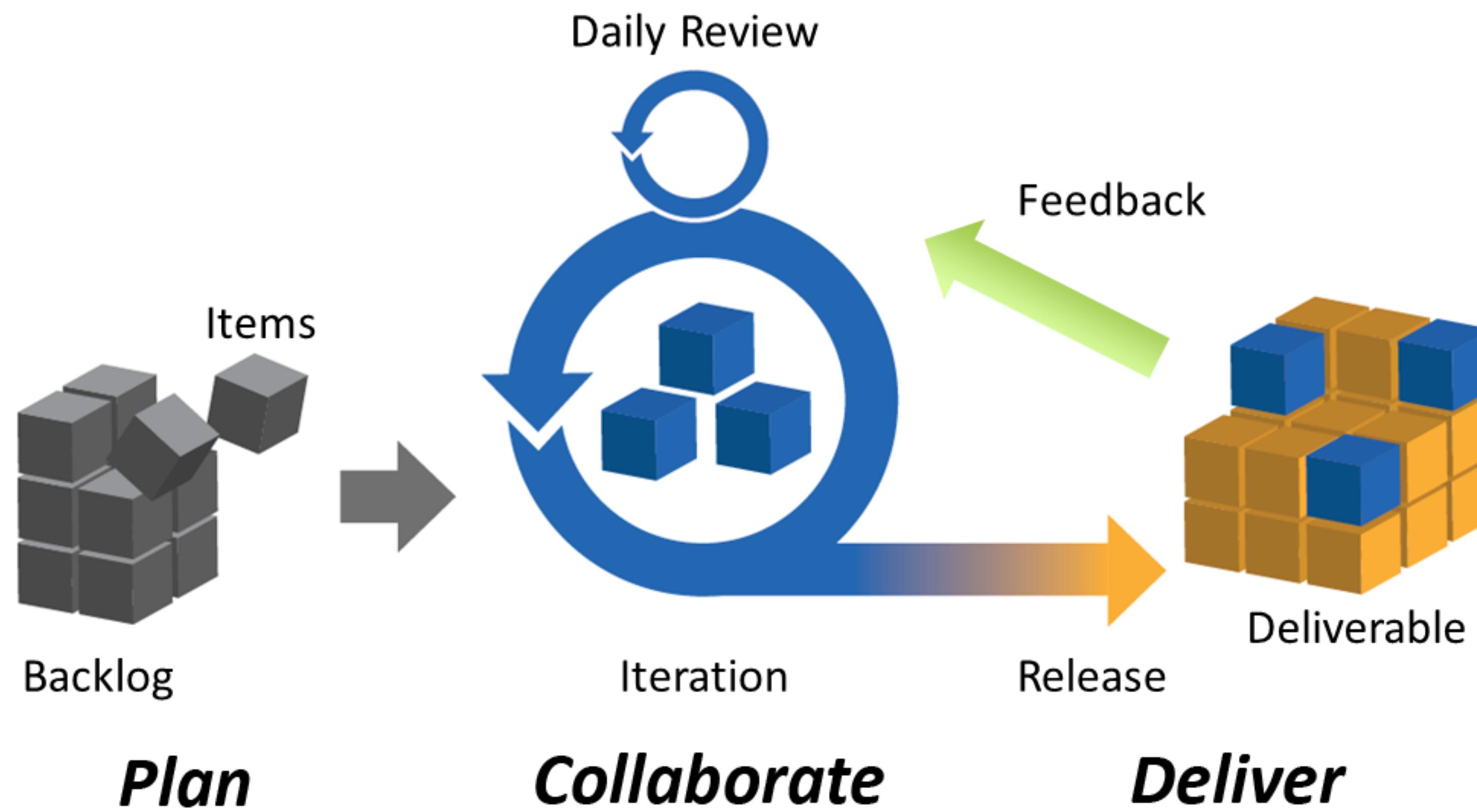
Intelligent systems derive their power
from the **knowledge they possess** rather
than the specific formalisms used.

-Edward Feigenbaum



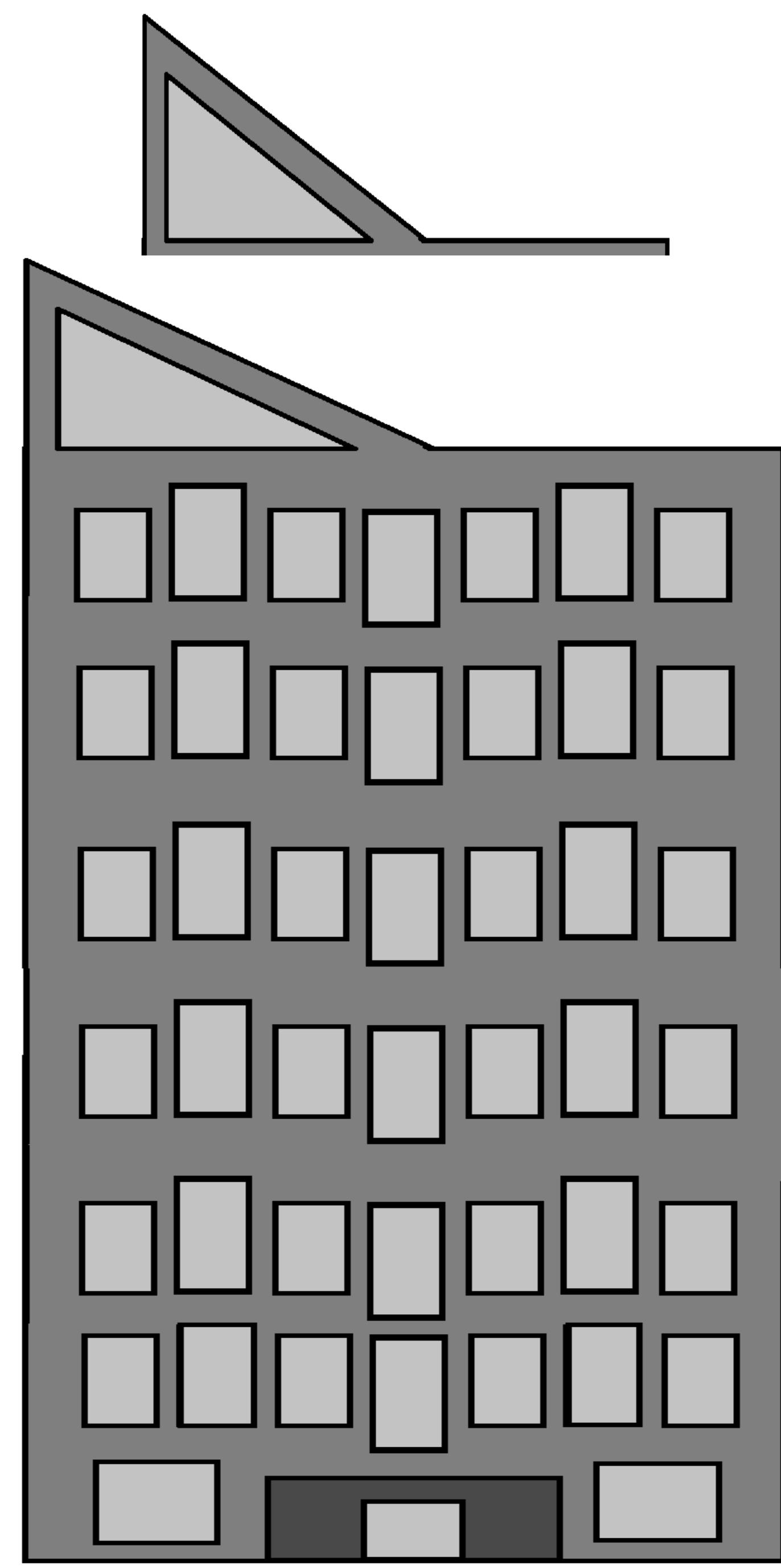


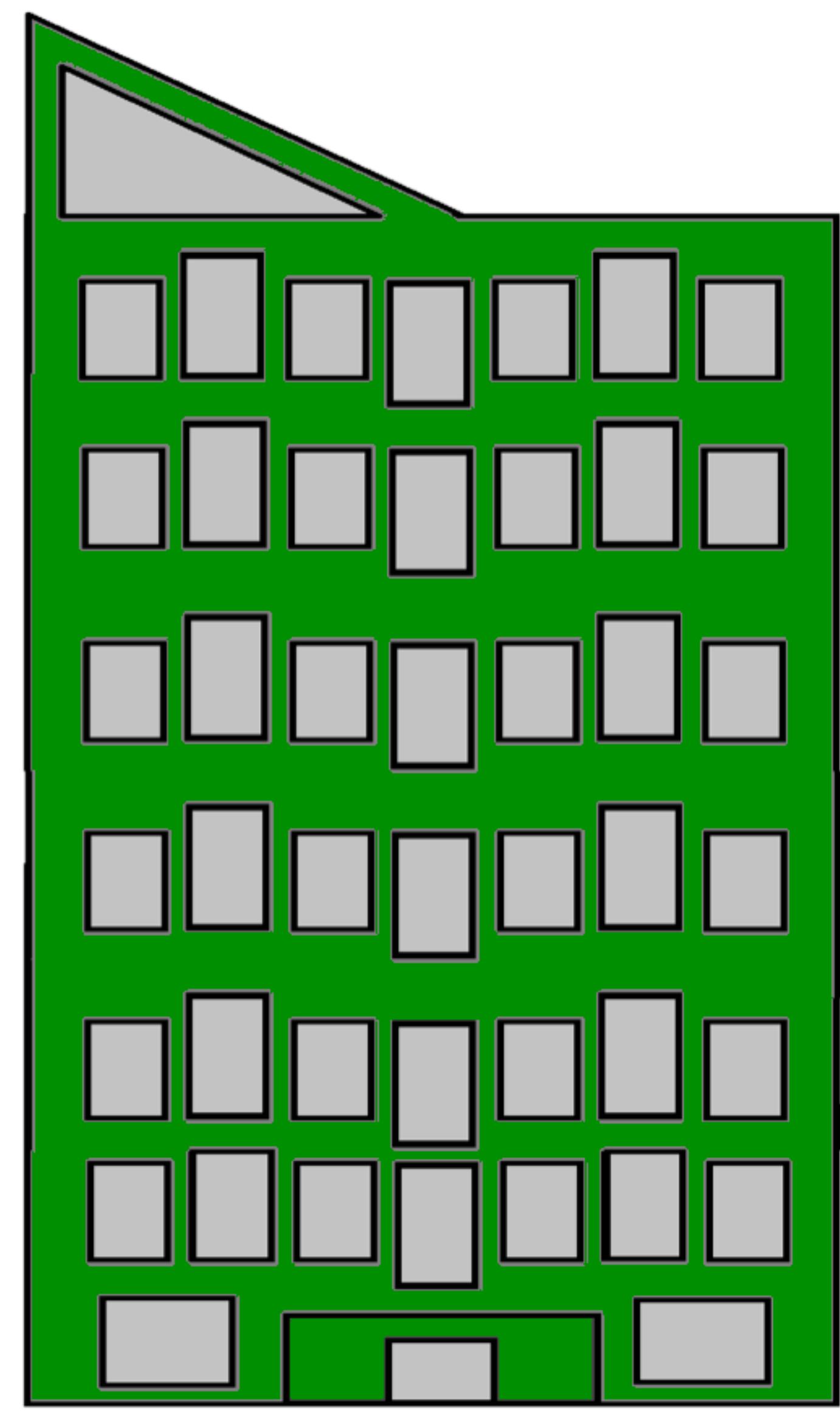
Agile Development!



Agile Project Management: Iteration

Imagine if other industries used
agile development....



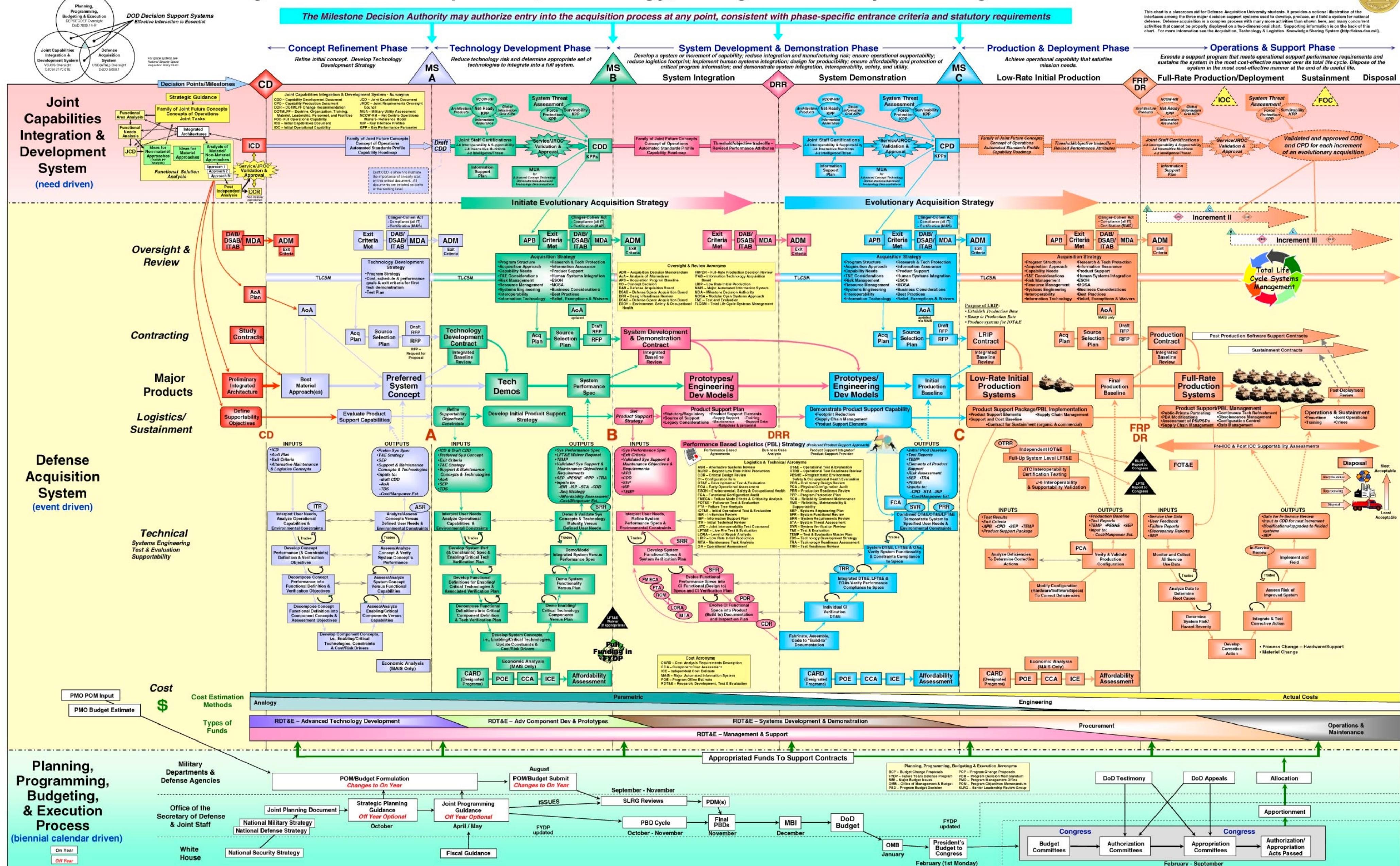


Agile works when iteration is cheap,
but iteration isn't always cheap.

Integrated Defense Acquisition, Technology, & Logistics Life Cycle Management Framework

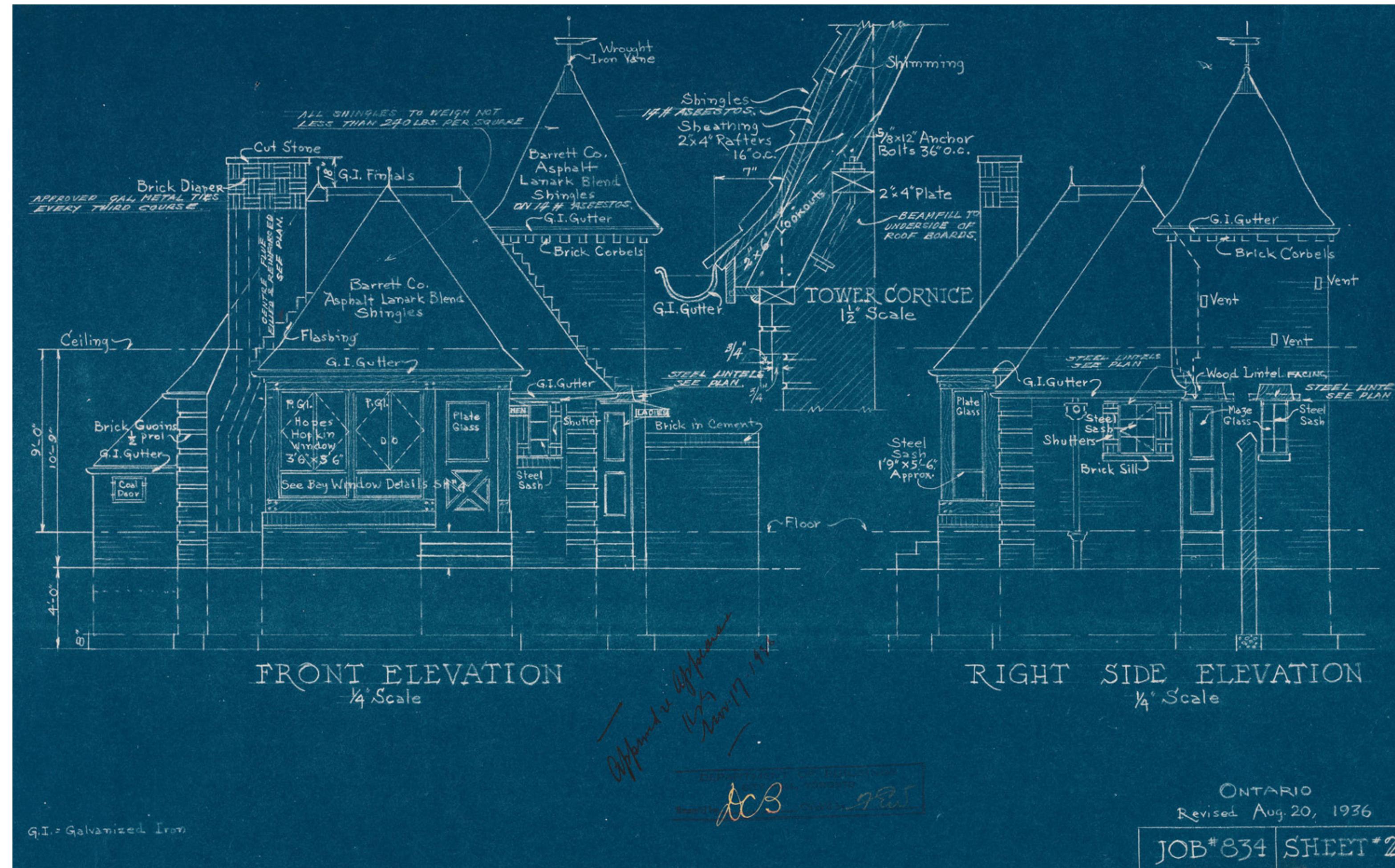


al
ent
his
(au-mil).

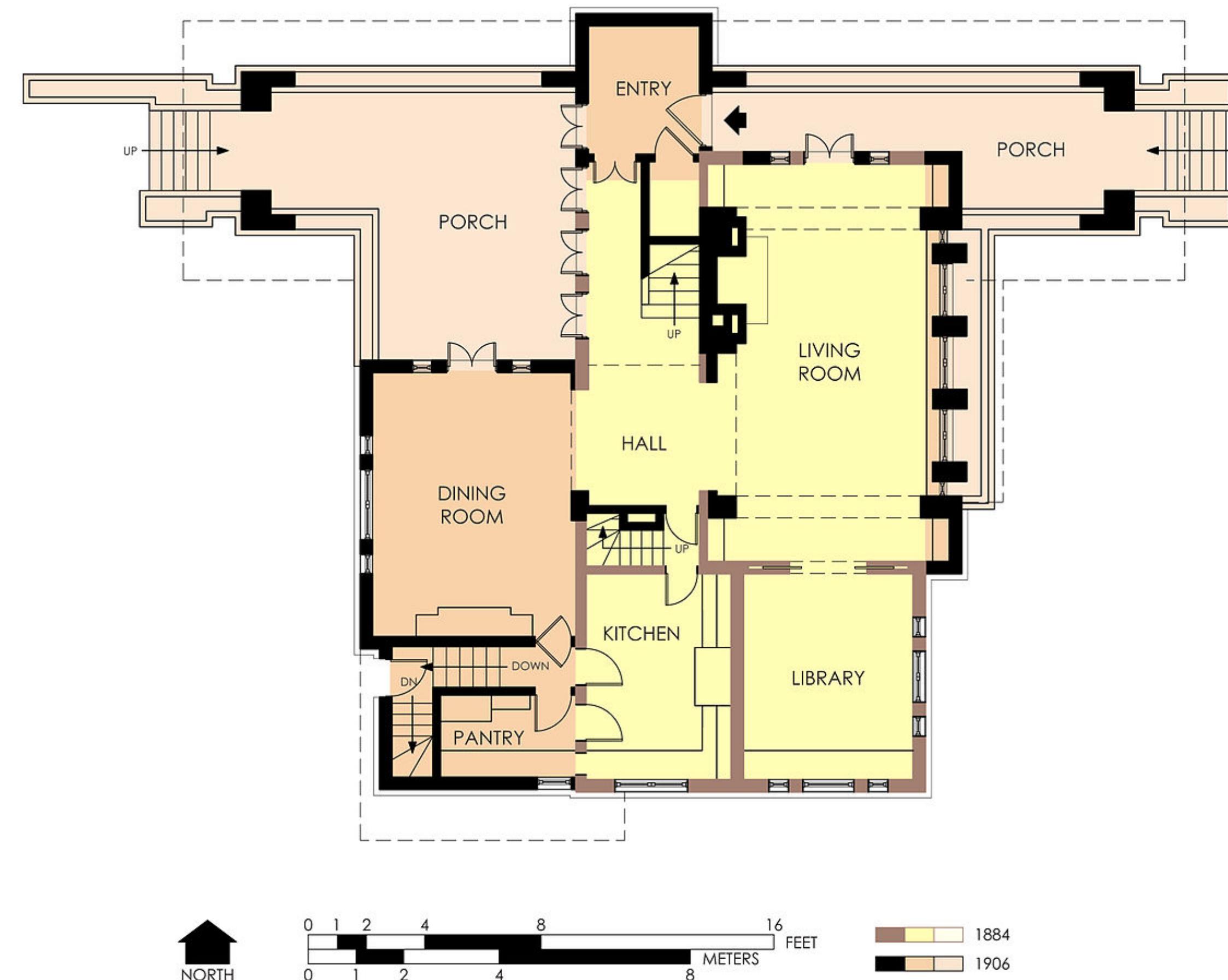


What can we learn from industries
where agile doesn't work?

Blueprints



Floor plans



Floor Plan Demo

A New Role for Rules

A tool for **developers**

A means to minimize the **semantic gap**

Leverage for a **shared understanding**

(Just don't write Java code in Excel.)

Questions?

@ryanbrush



<http://engineering.cerner.com>

<http://github.com/rbrush/clara-rules>

Image Sources

- [http://en.wikipedia.org/wiki/Waterfall_model#mediaviewer/
File:Waterfall_model.svg](http://en.wikipedia.org/wiki/Waterfall_model#mediaviewer/File:Waterfall_model.svg)
- <http://www.hpl.hp.com/hpjournal/pdfs/IssuePDFs/1977-03.pdf>
- <http://www.nsf.gov/od/lpa/news/03/images/mosaic.6beta.jpg>

Backup Slides

