

Capability Red

Liz Keogh

@lunivore

September 2014

A story
about
software that matters

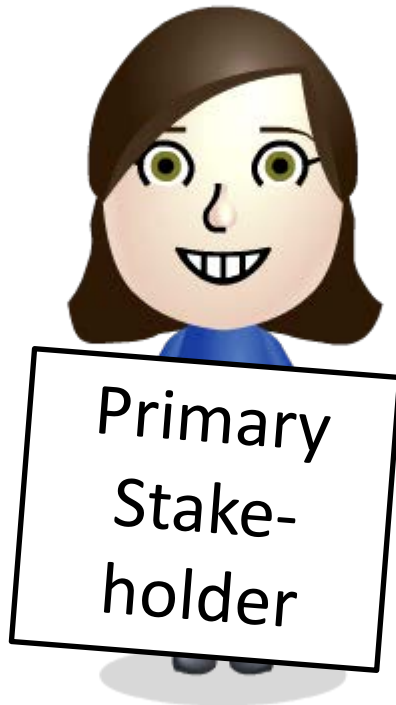
“Hunt the value.”

- Chris Matts,

~~Feature Injection~~

Value Mapping

Vision



Makes money

Saves money

Protects money



Goal



Incidental
Stake-
holder

Needed
to go live

Themes / Feature Sets



A themed
group of
features

Scenario



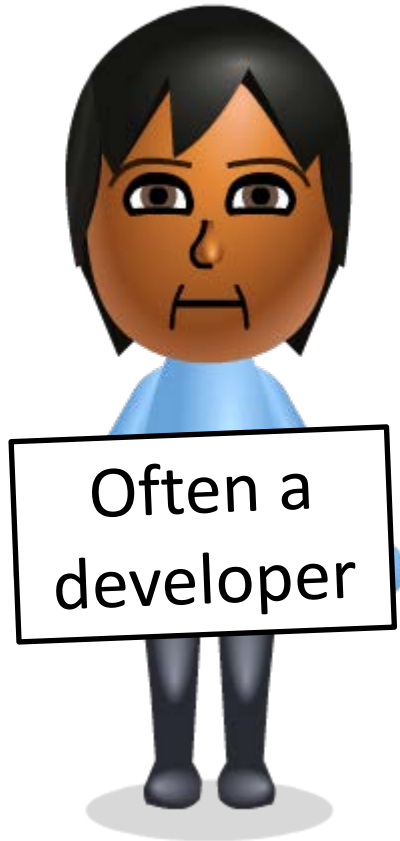
An example
that illustrates
system behaviour
and how
it delivers
value

Feature



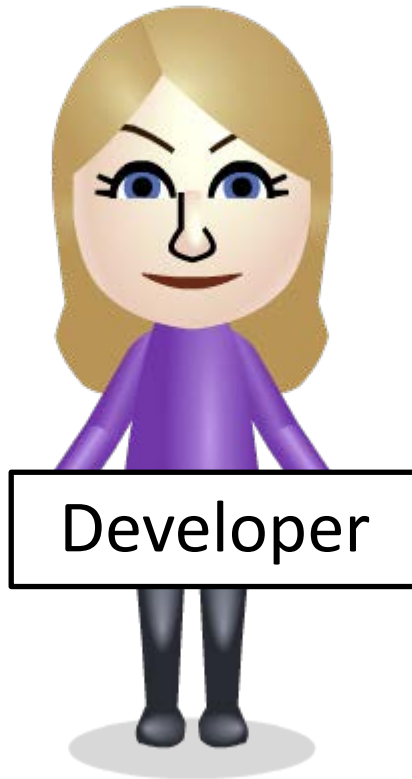
User interface
component
which enables
a capability

Story



A slice through
a feature
to enable
faster
feedback

Code



Ideas turning into
reality

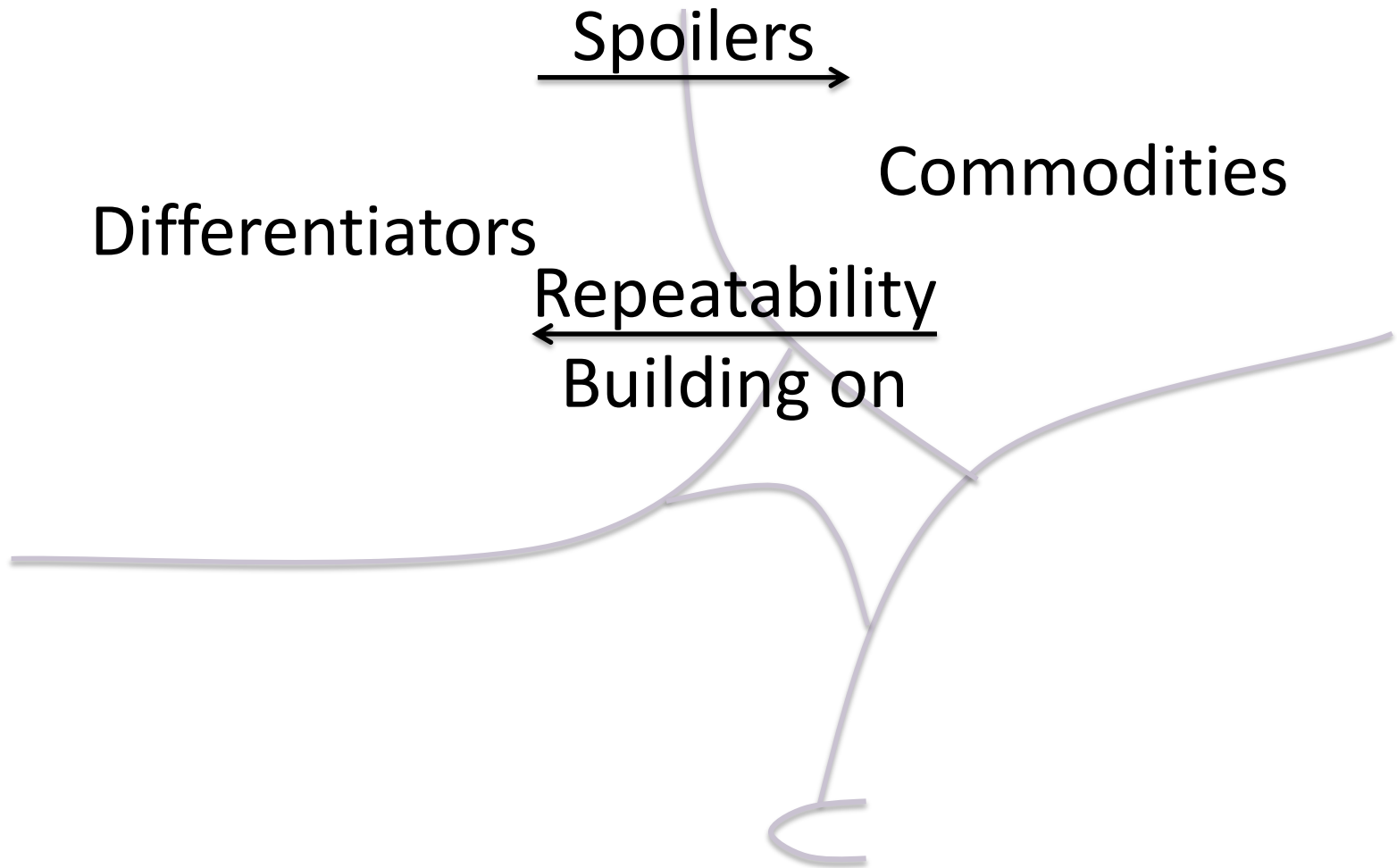
Other ways of asking “Why”?

What will be
different?

What will that get for you
that you
don't already have?

What will you, or the users, or the system,
be able to do
that you can't do already?

The Innovation Cycle



Etymology

***capere* -**

to grasp, lay hold, take, catch;
undertake;
take in, hold;
be large enough for;
comprehend

- etymonline.com

~~Themes / Feature Sets~~



Capability:

The ability to do
something
really well

Continuous vs. Discrete Capabilities

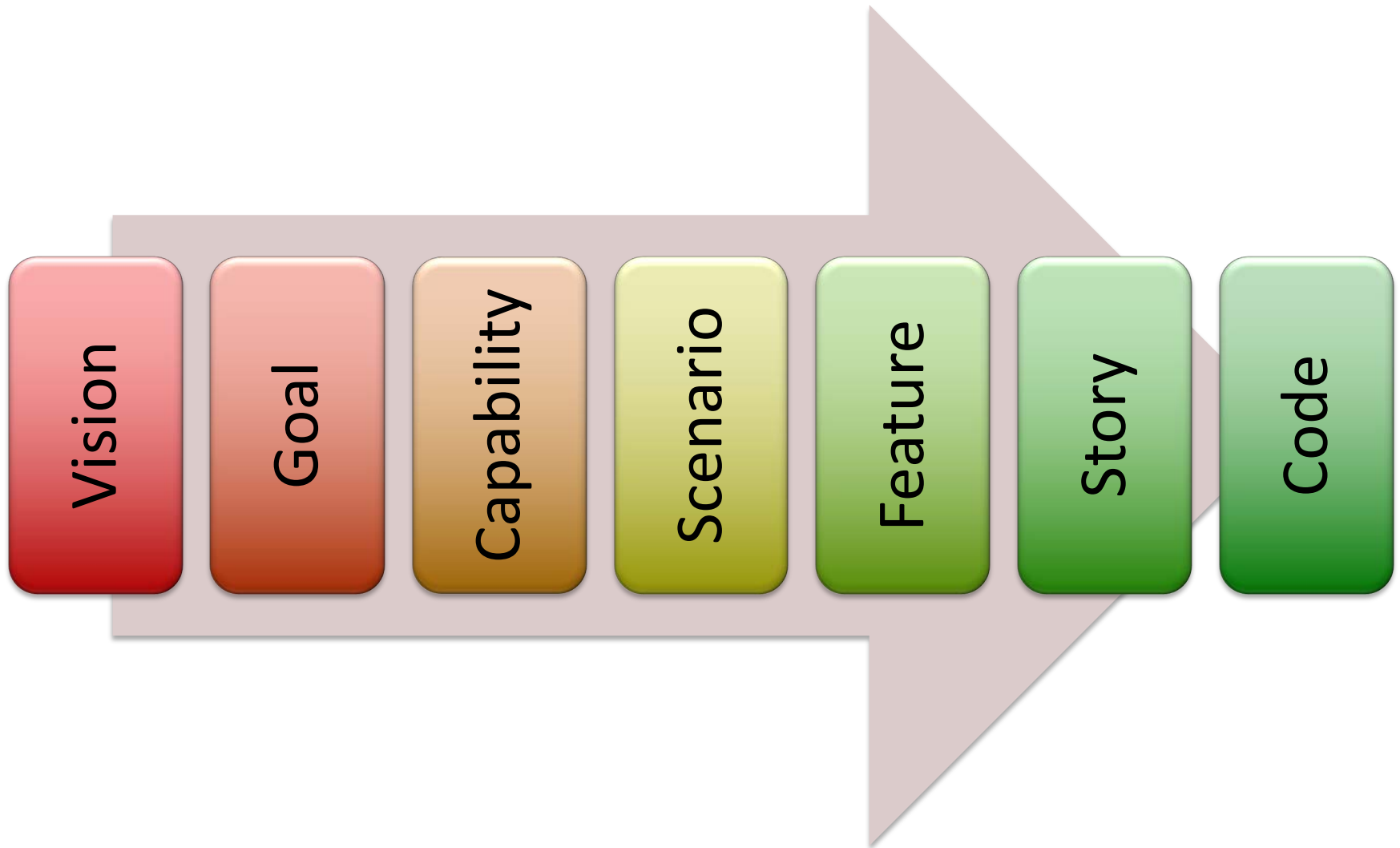
If you can't test it,
monitor it.

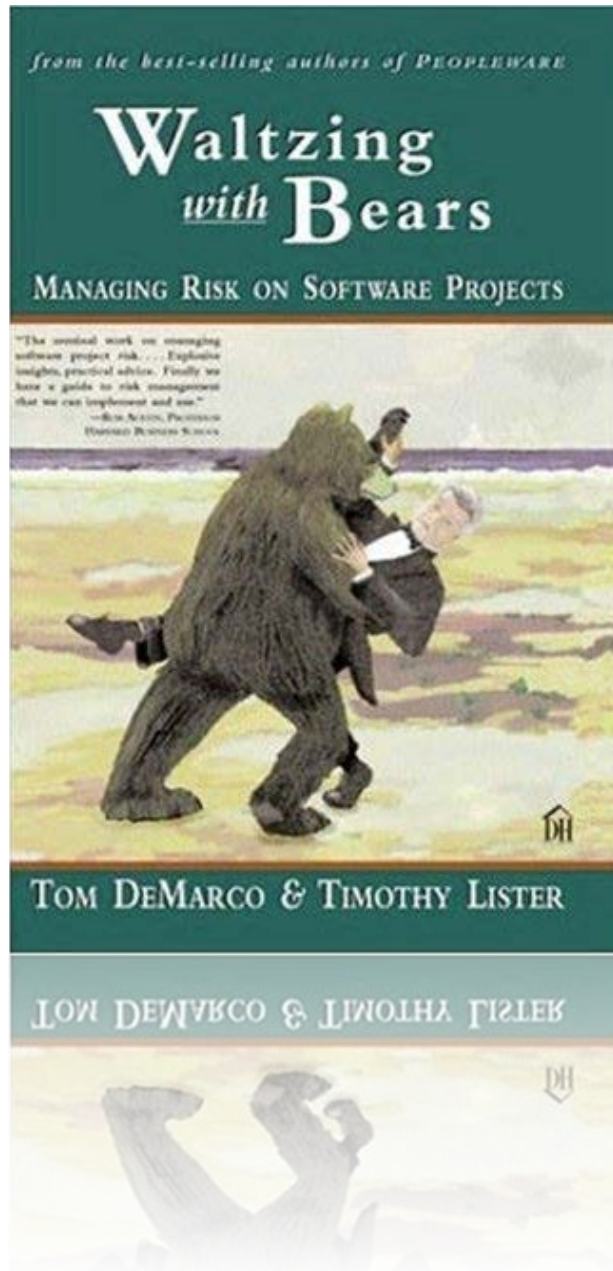
Put some numbers on them!

We want to be able to
update the front page
~~more quickly.~~

in 5 minutes.

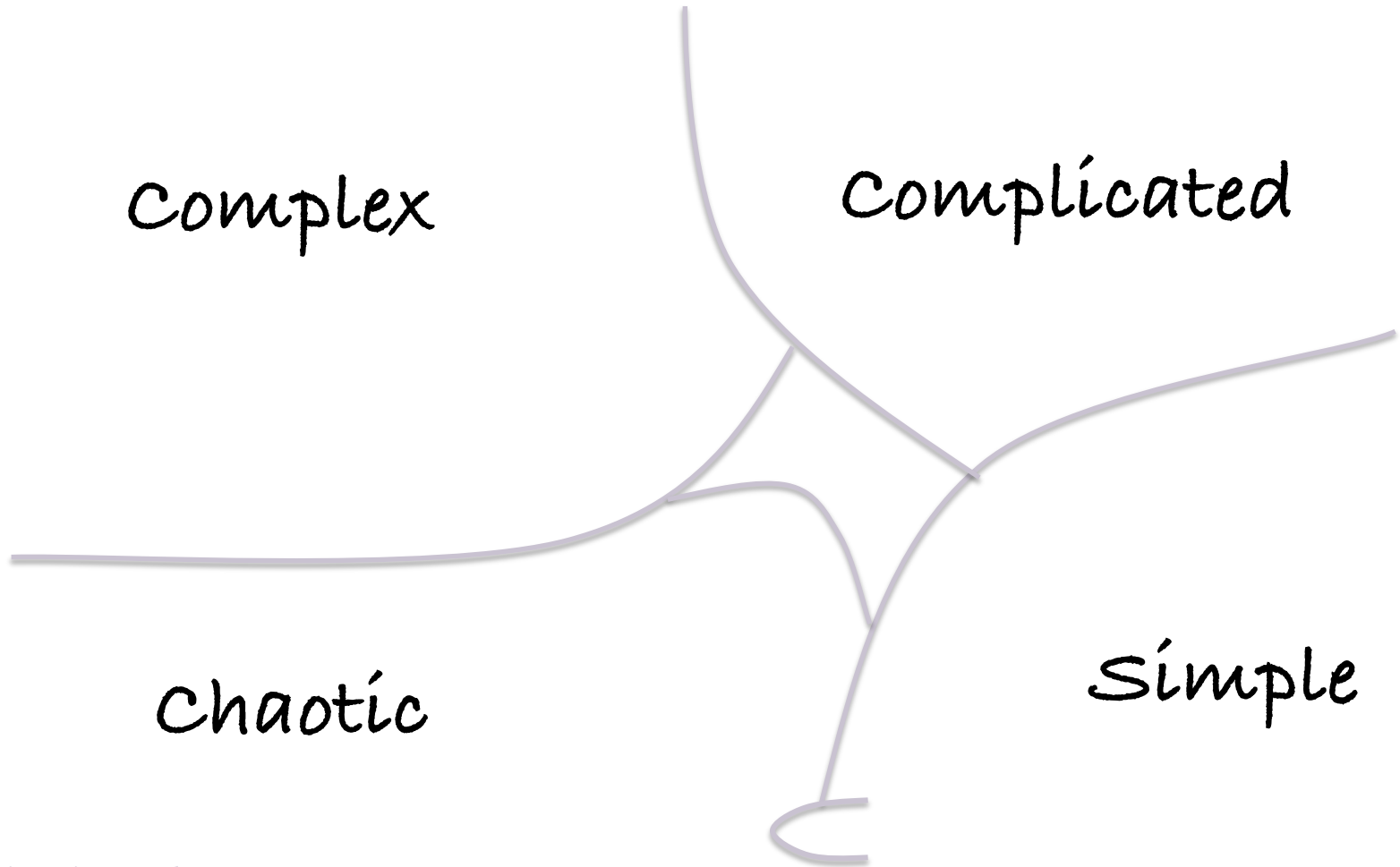
Breaking everything down up-front
is Waterfall!





If a project has
no risks,
don't do it.

Cynefin

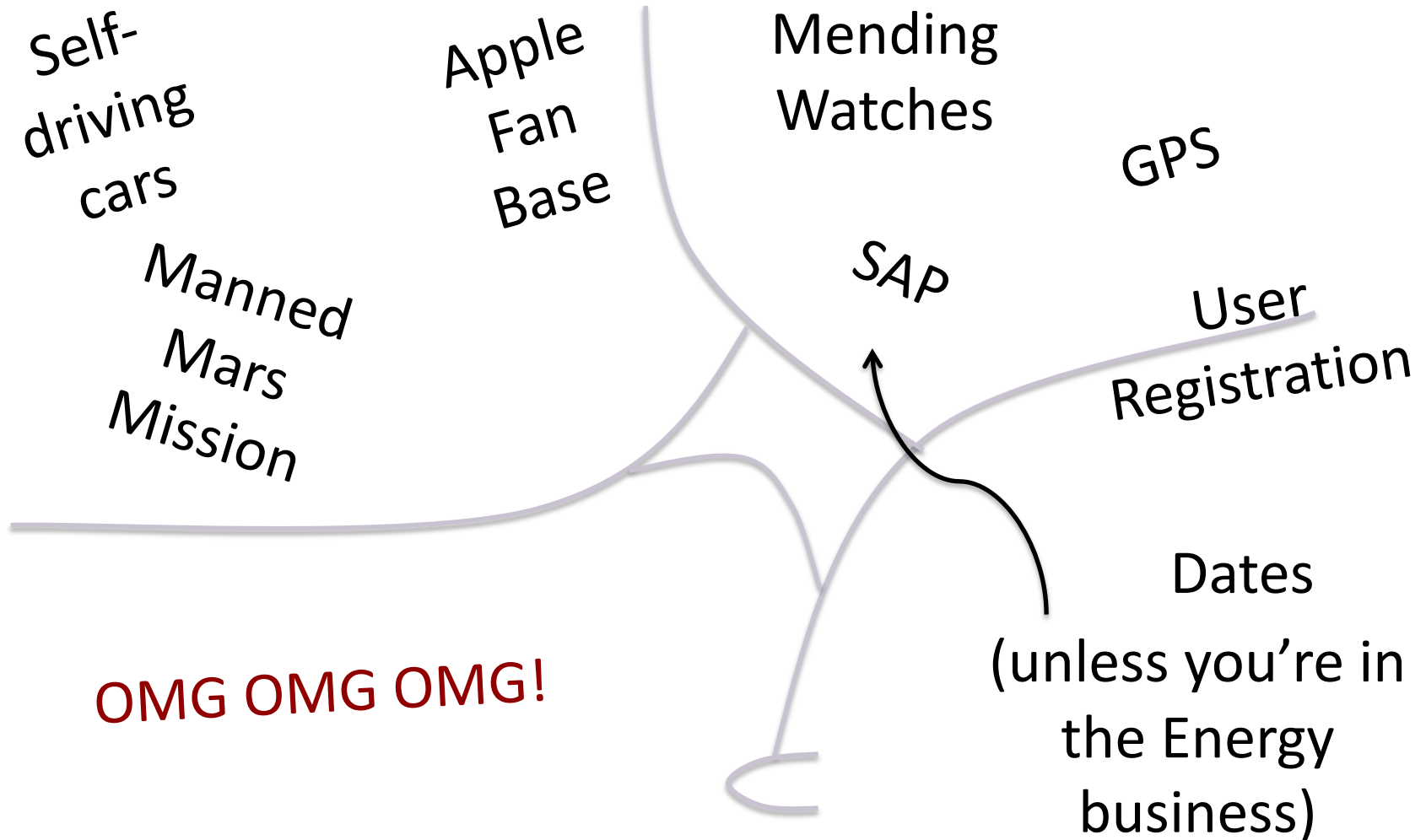


With thanks to
David Snowden and Cognitive Edge

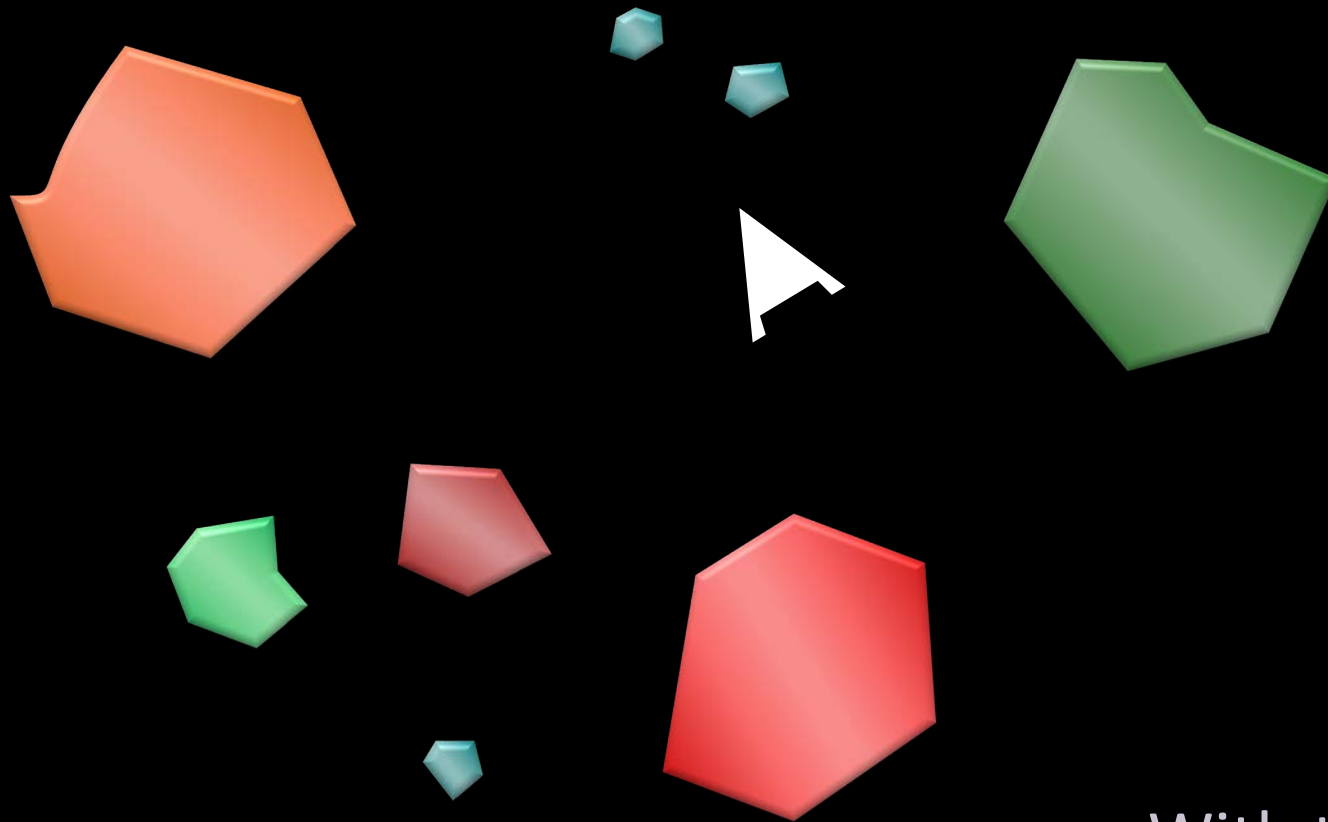
Estimating Complexity

5. Nobody has ever done it before
4. Someone outside the org has done it before
(probably a competitor)
3. Someone in the company has done it before
2. Someone in the team has done it before
1. We all know how to do it.

Estimating Complexity

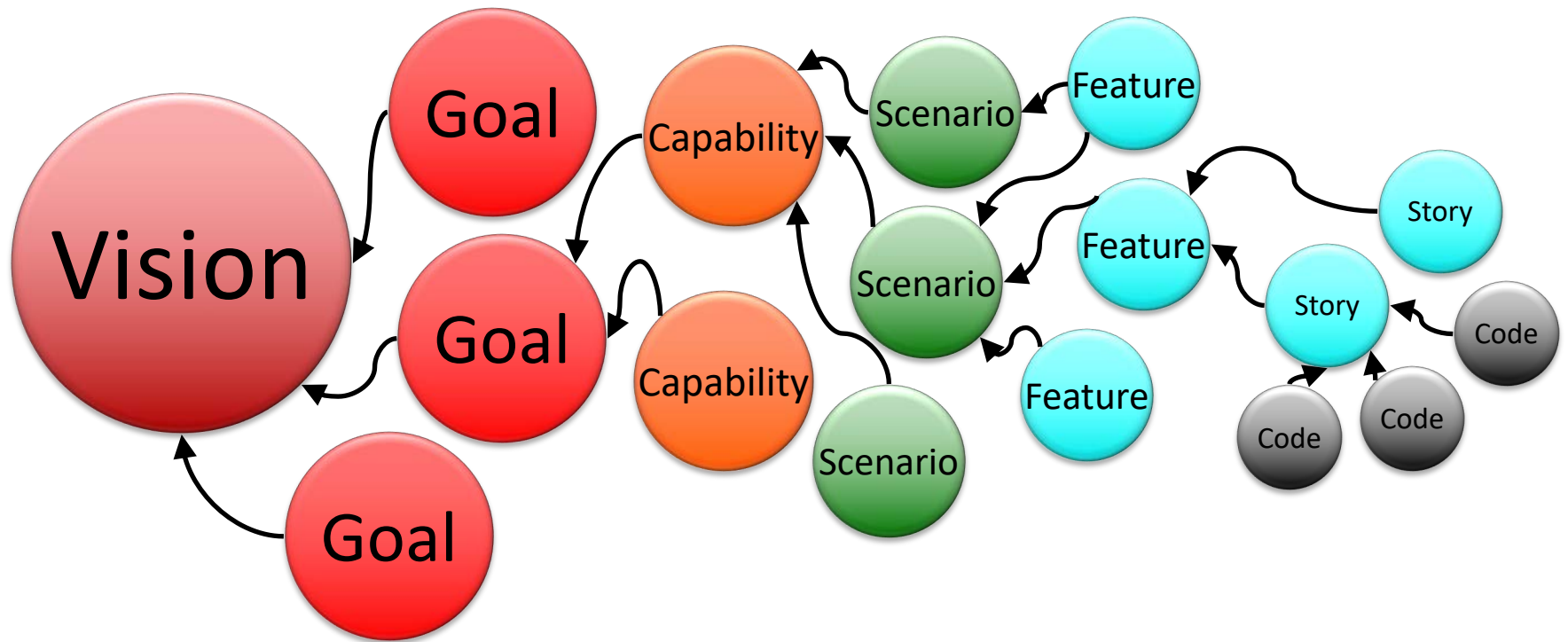


Breaking Things Down

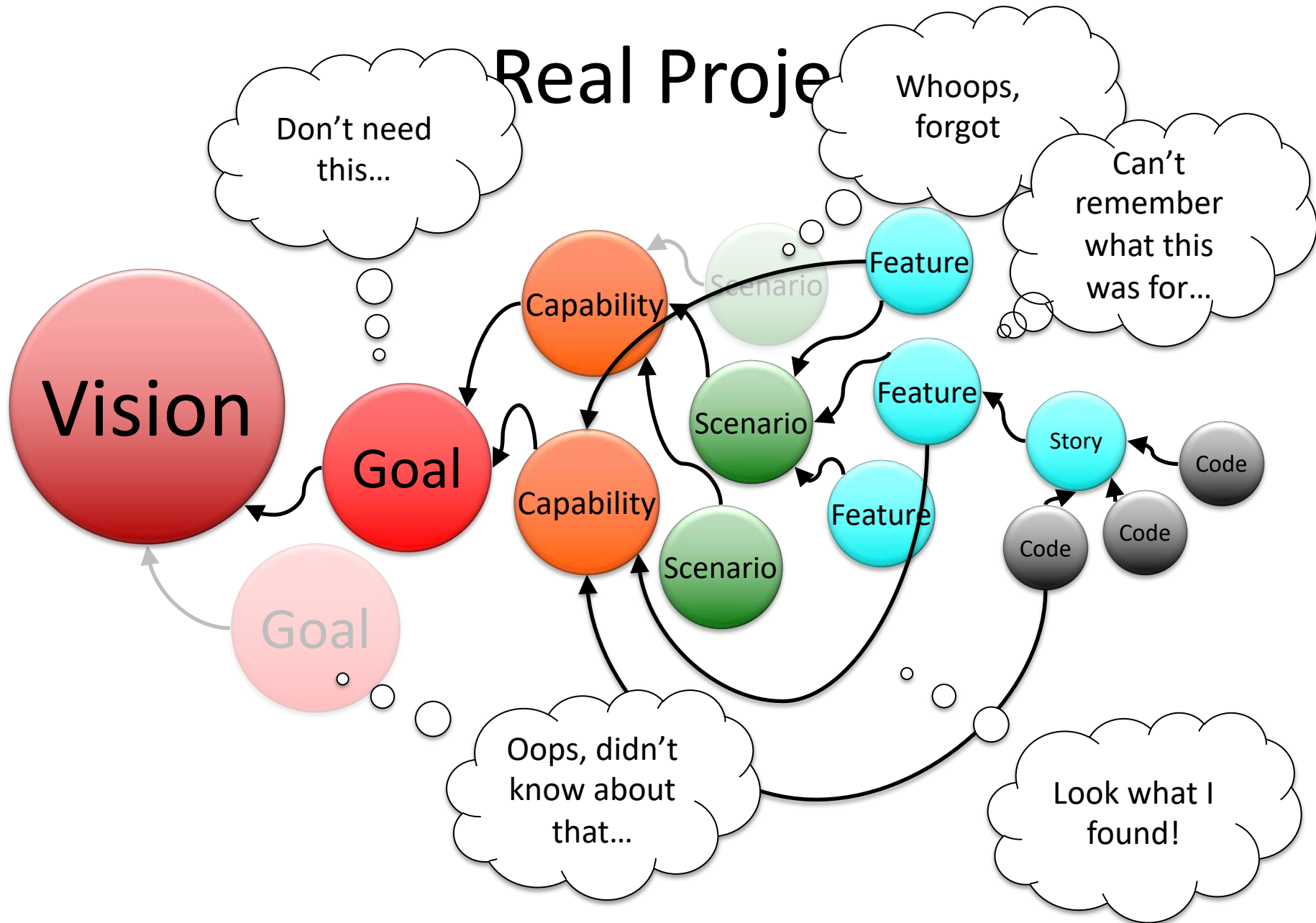


With thanks to
Michael James

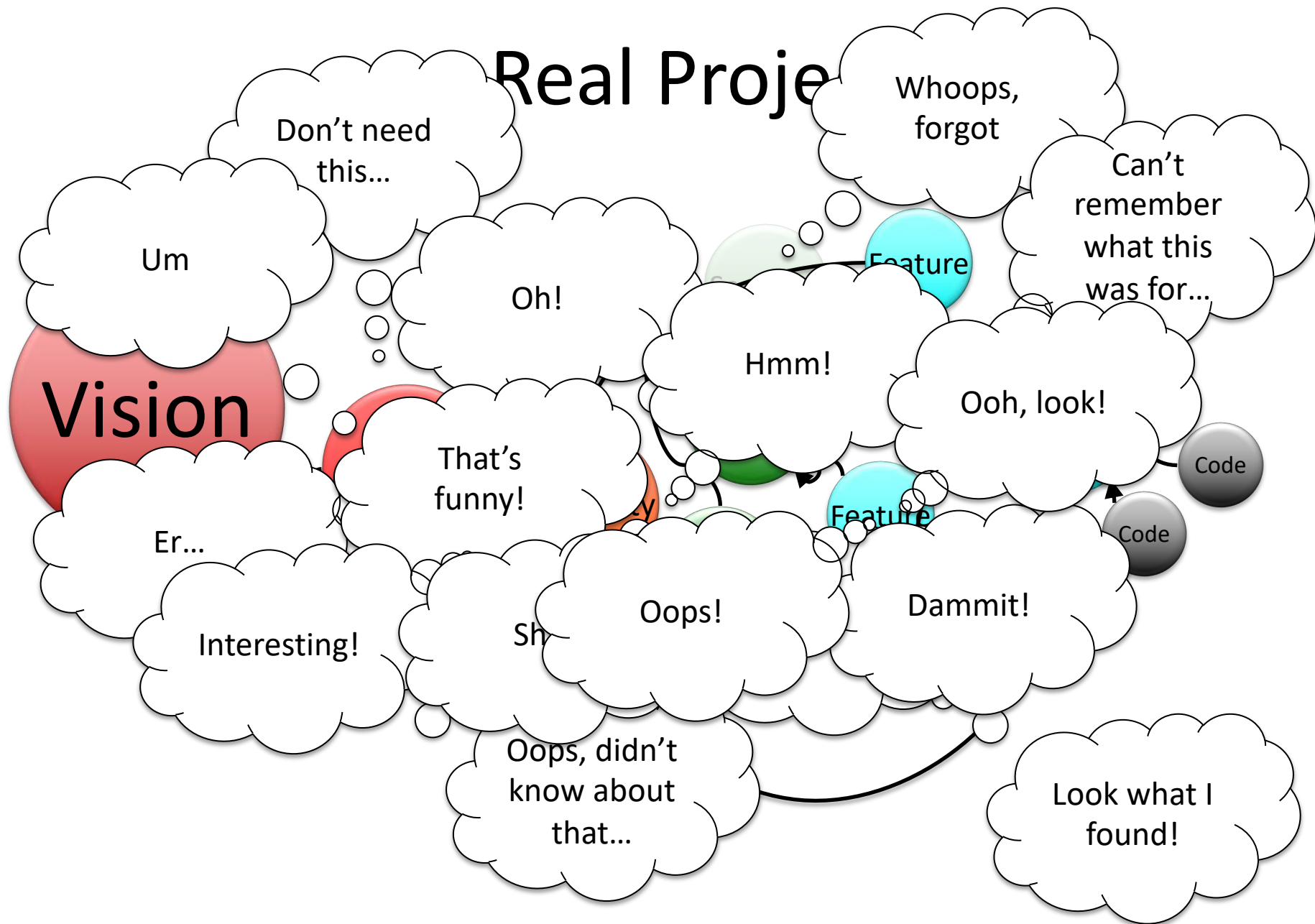
Fractal beauty



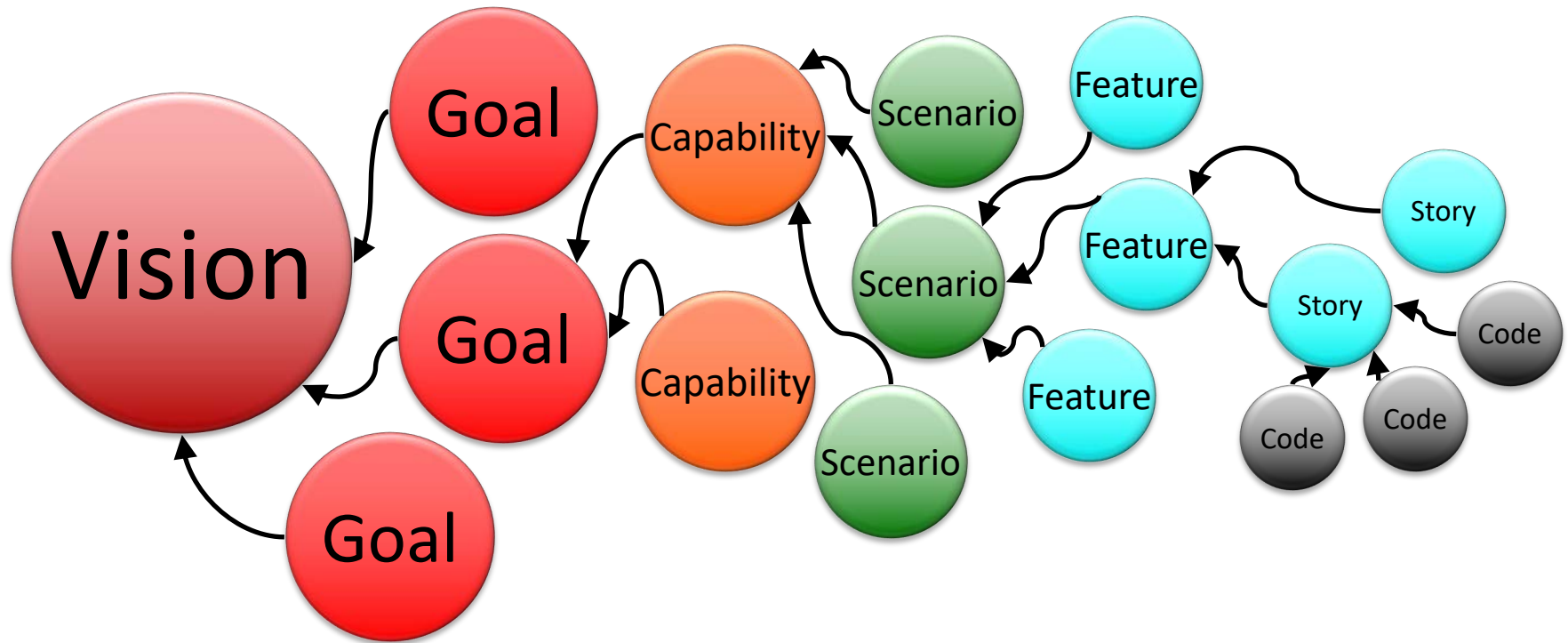
Real Project



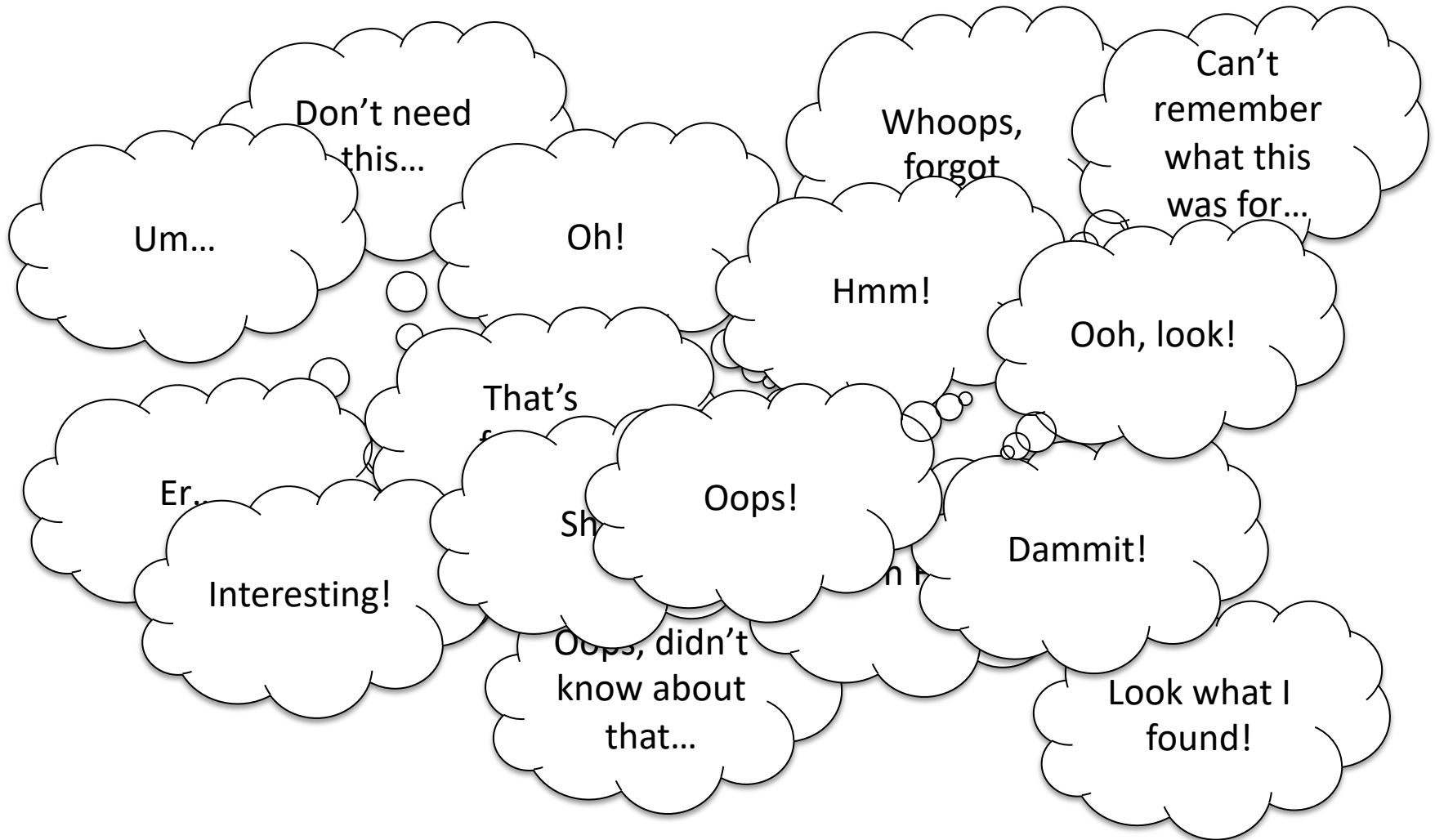
Real Project



~~We are uncovering better ways of
developing software by doing it~~



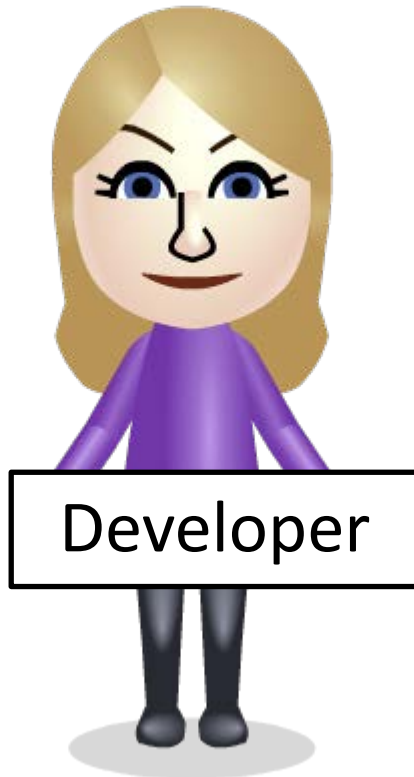
We're discovering how to **discover stuff** by doing it



Different levels of granularity

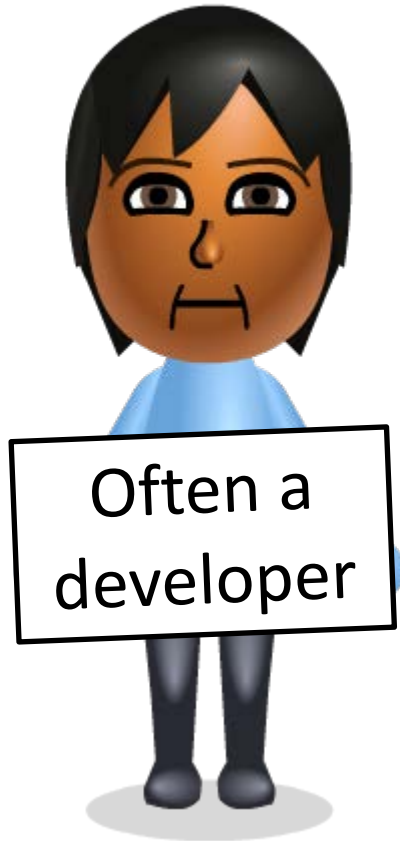


Testing Code



Do the examples
now work?

Testing a Story



What do our
stakeholders
and users
want to change
about this?

Testing a Feature



Is this usable?
Does it
deliver the
capability?

Testing Scenarios



Does the
application
produce
the outcome
we expected?

Testing Capability



Can we do
what we
wanted to do,
for the contexts
in which
we wanted to do it?

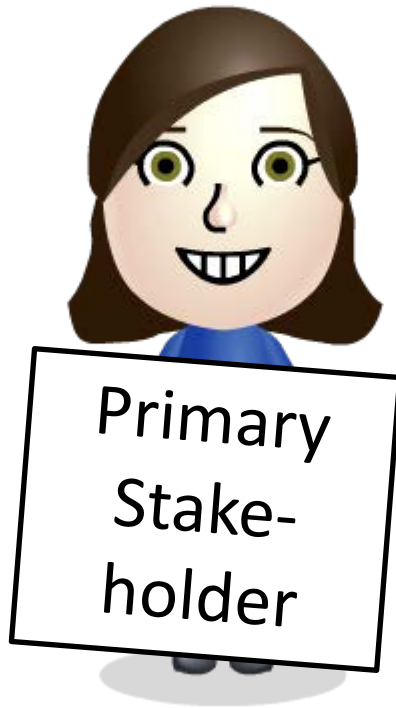
Testing a Goal



Incidental
Stake-
holder

Does this
meet our
stakeholder's
goal?

Testing a Vision



Did we:

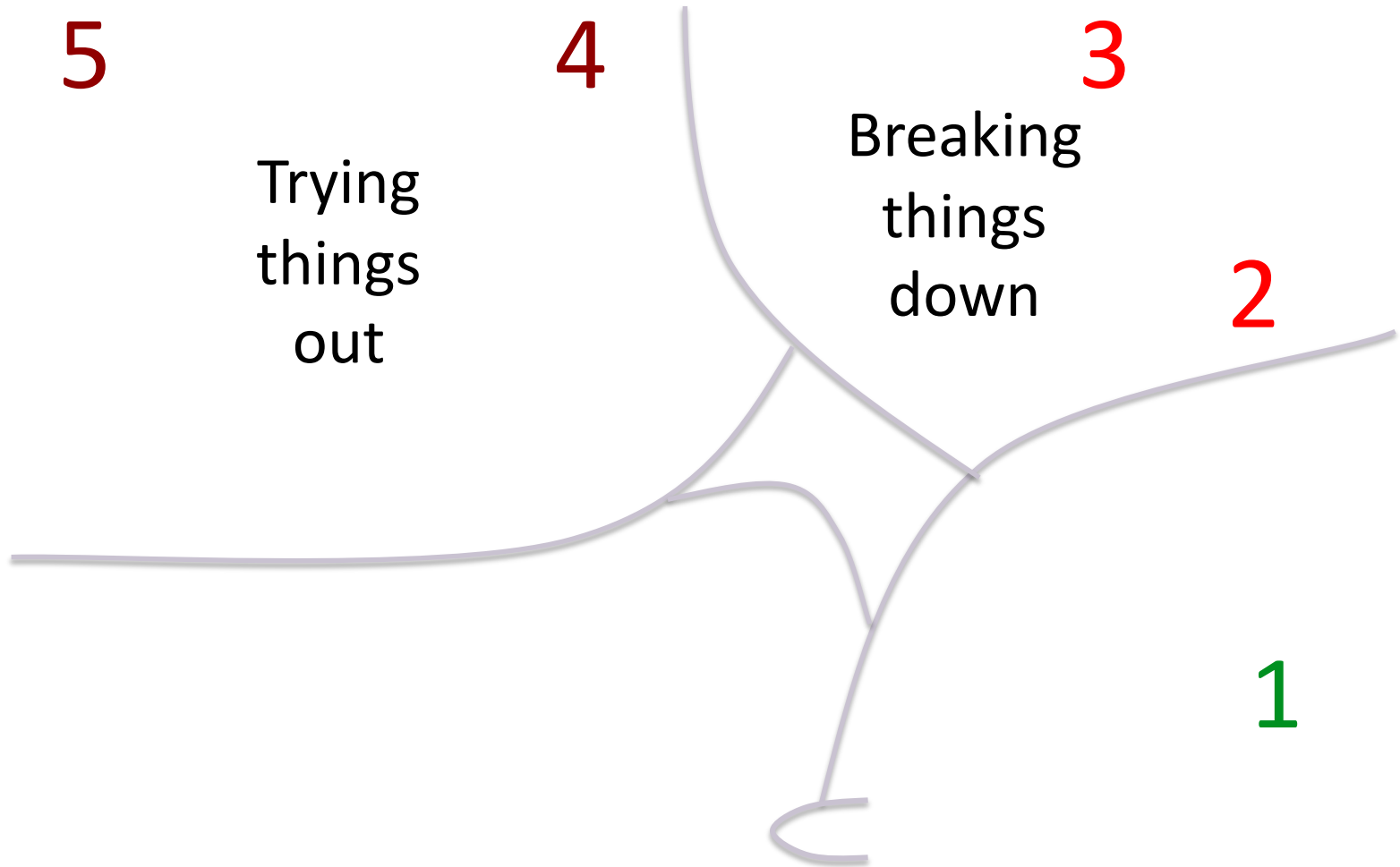
Make money

Save money

Protect money?



Estimating Complexity



There's always one
Red

How to do Capability Red

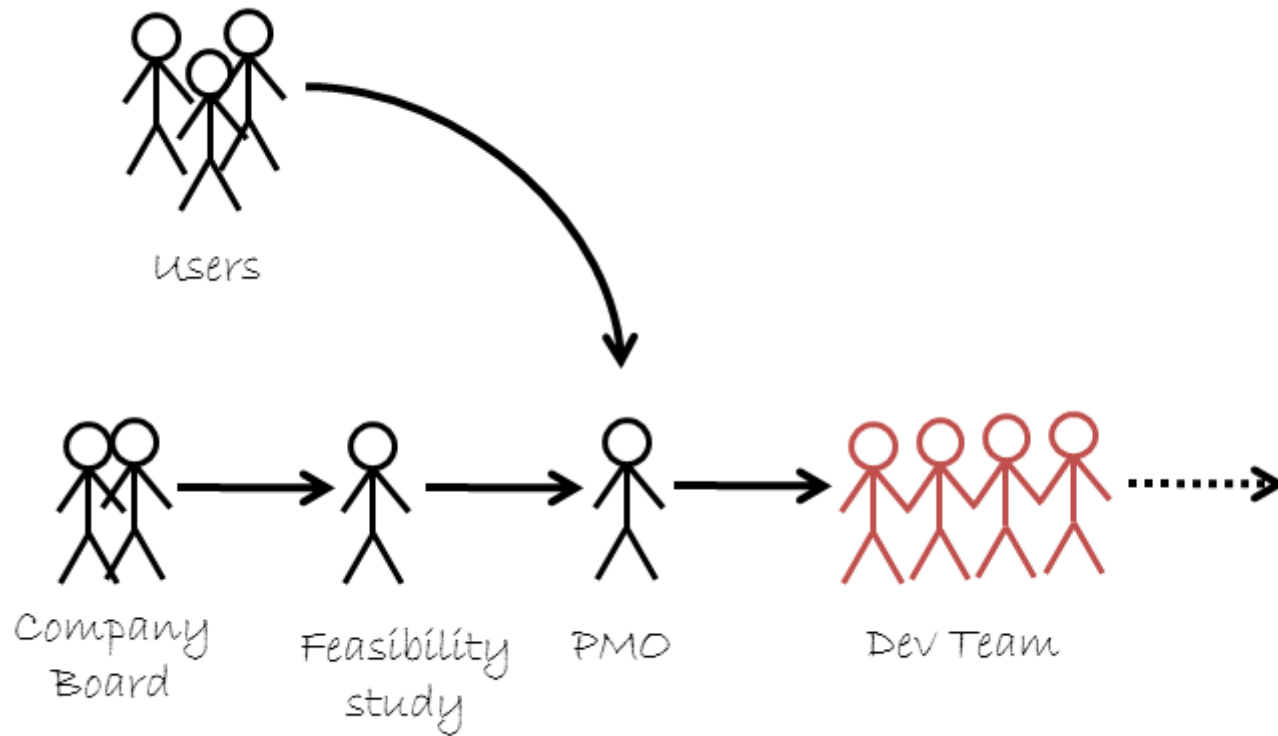
Before you ask

why,

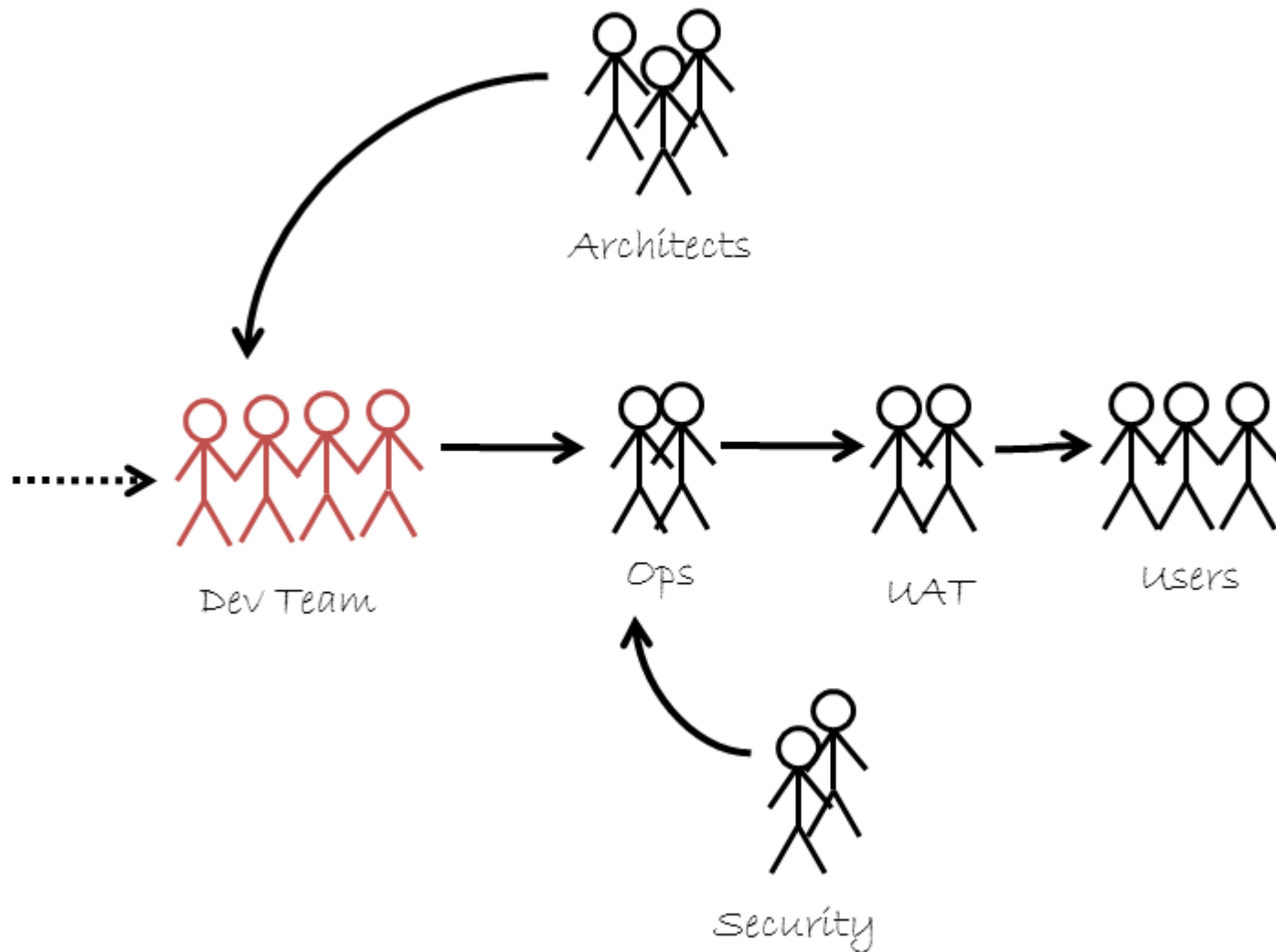
ask

who

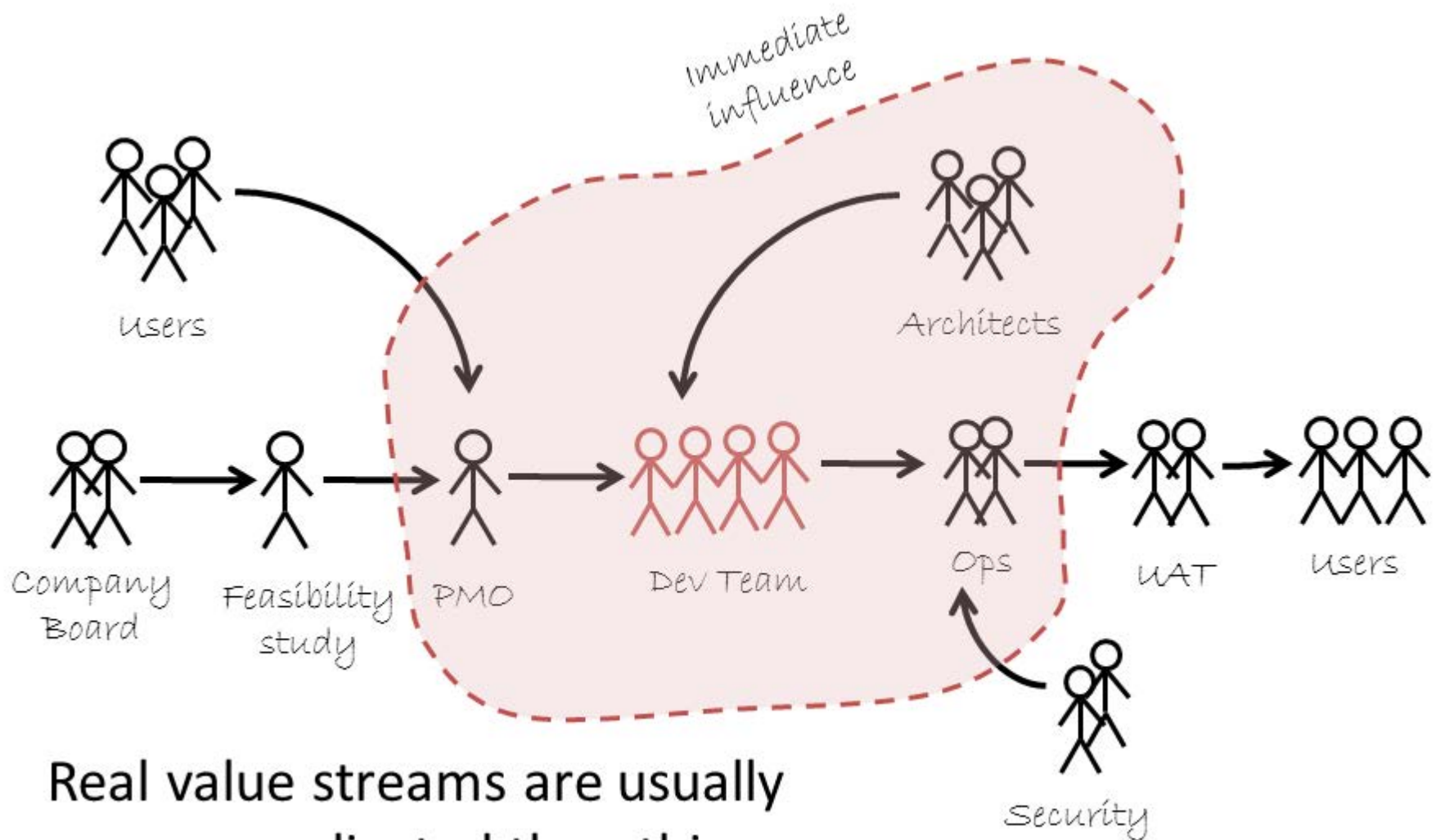
Identify your Stakeholders



Gatekeepers vs. Educators

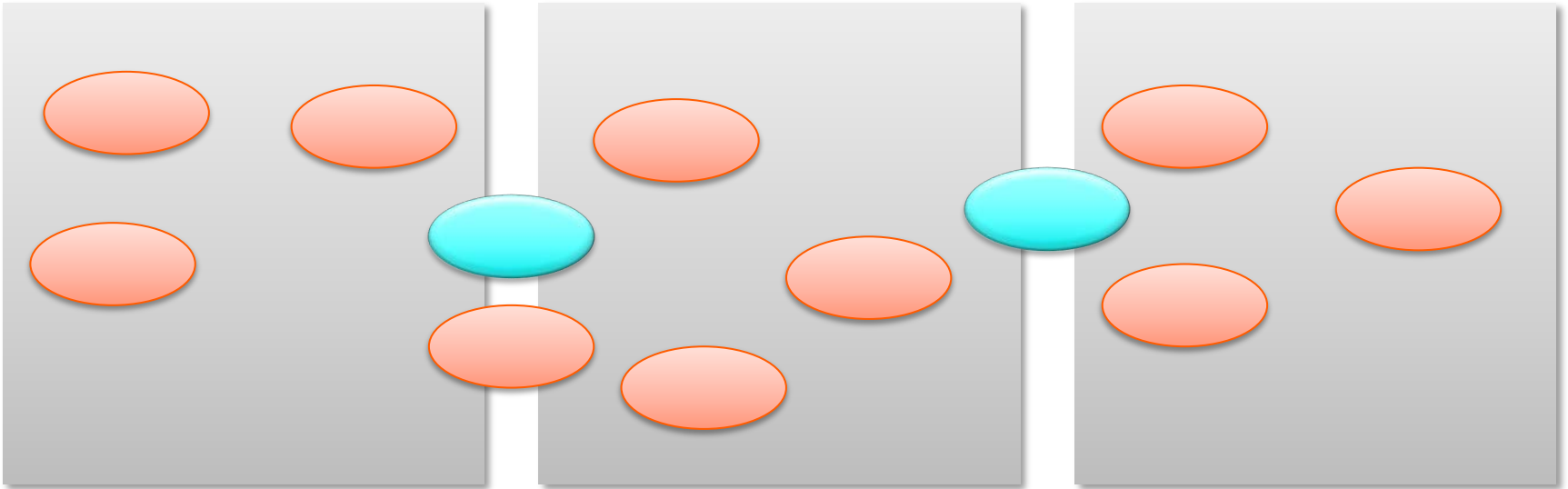


Value Streams



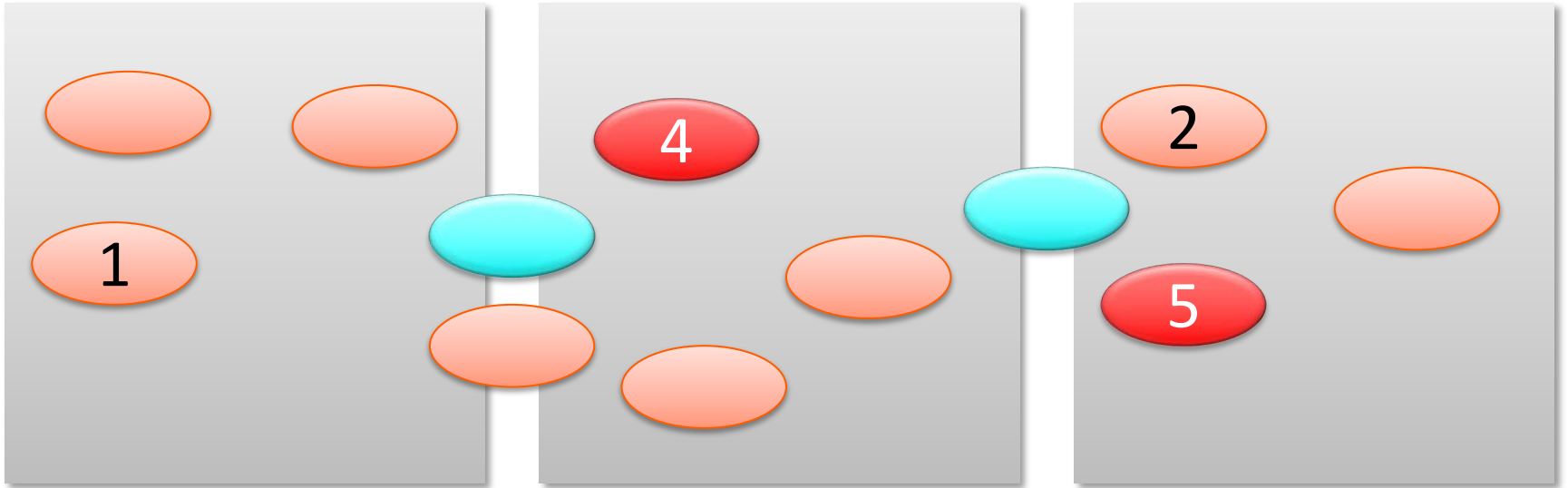
Real value streams are usually more complicated than this.

Draw up the capabilities



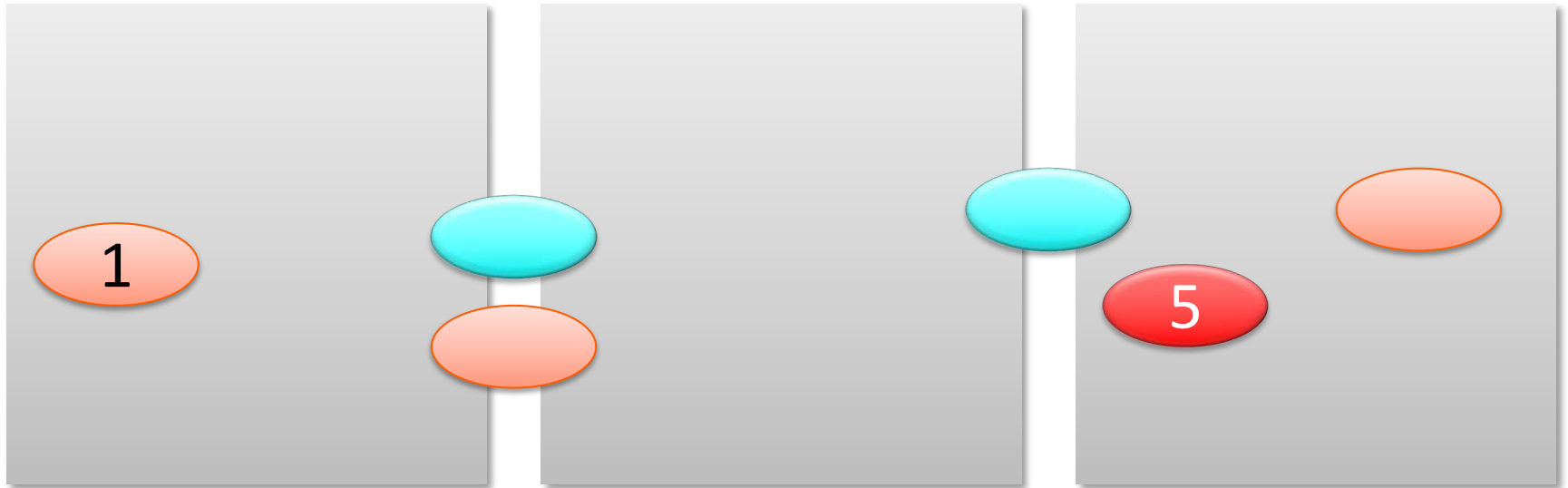
Don't forget the integration points!

Put complexity estimates on



Pay particular attention to 4s and 5s

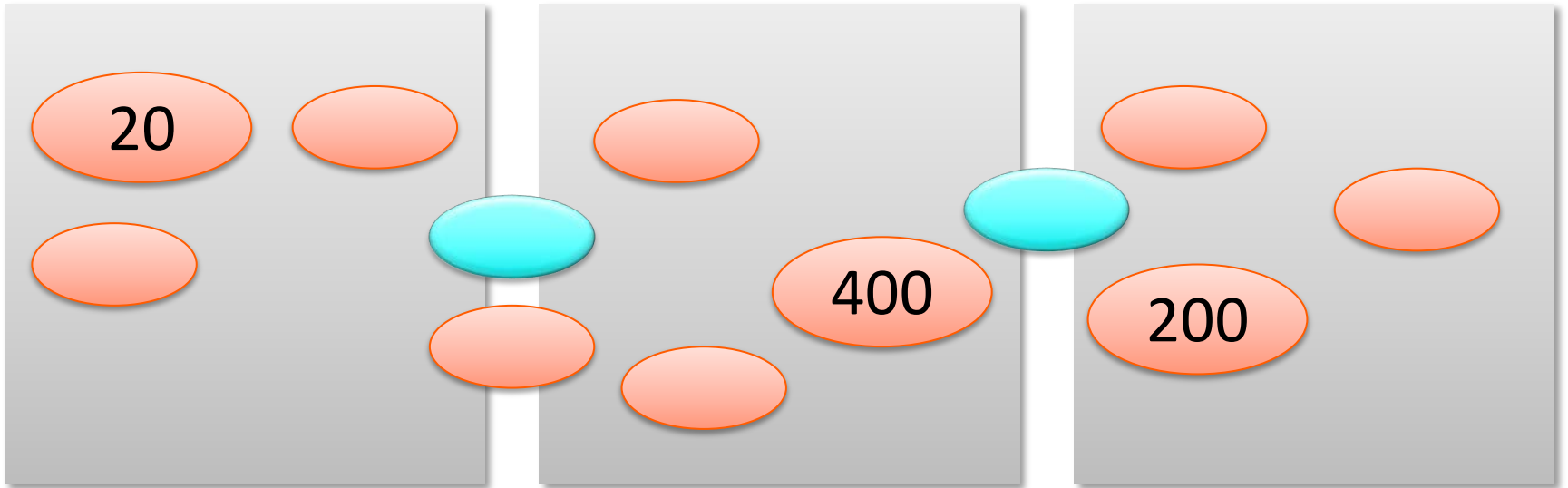
See if you can narrow the MVP



Only one 4 or 5.

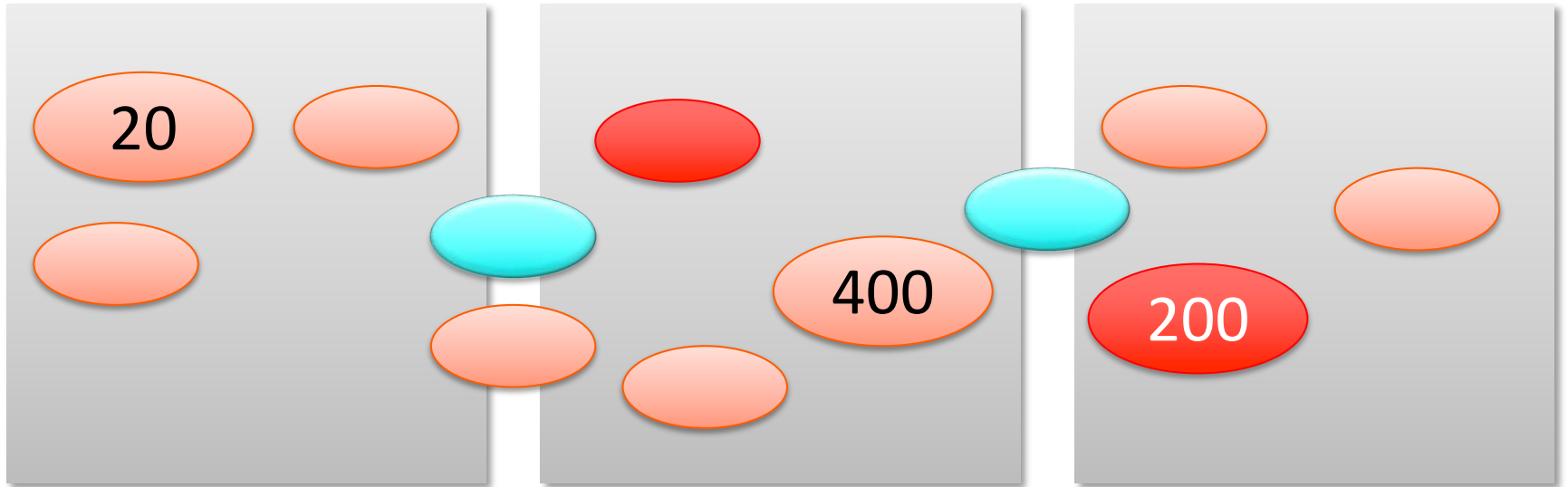
Companies with high transaction costs
for going live might not be able to do this.

Estimate, if it's useful



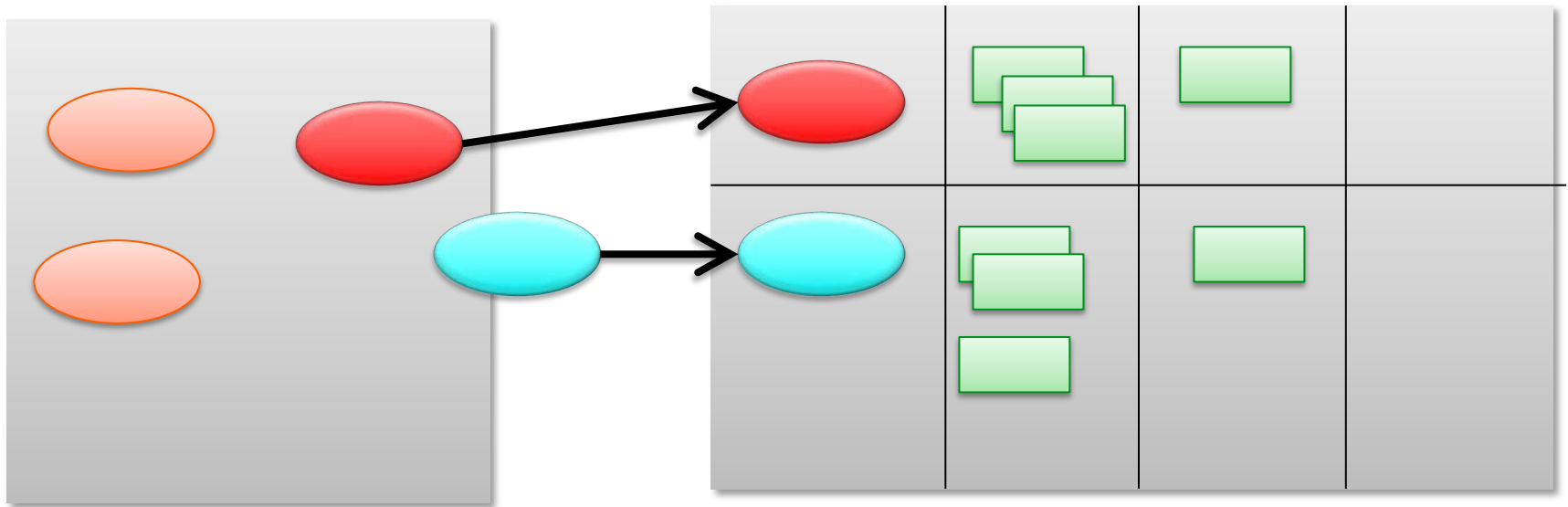
Keeps the PMO happy

Double the 4s and 5s

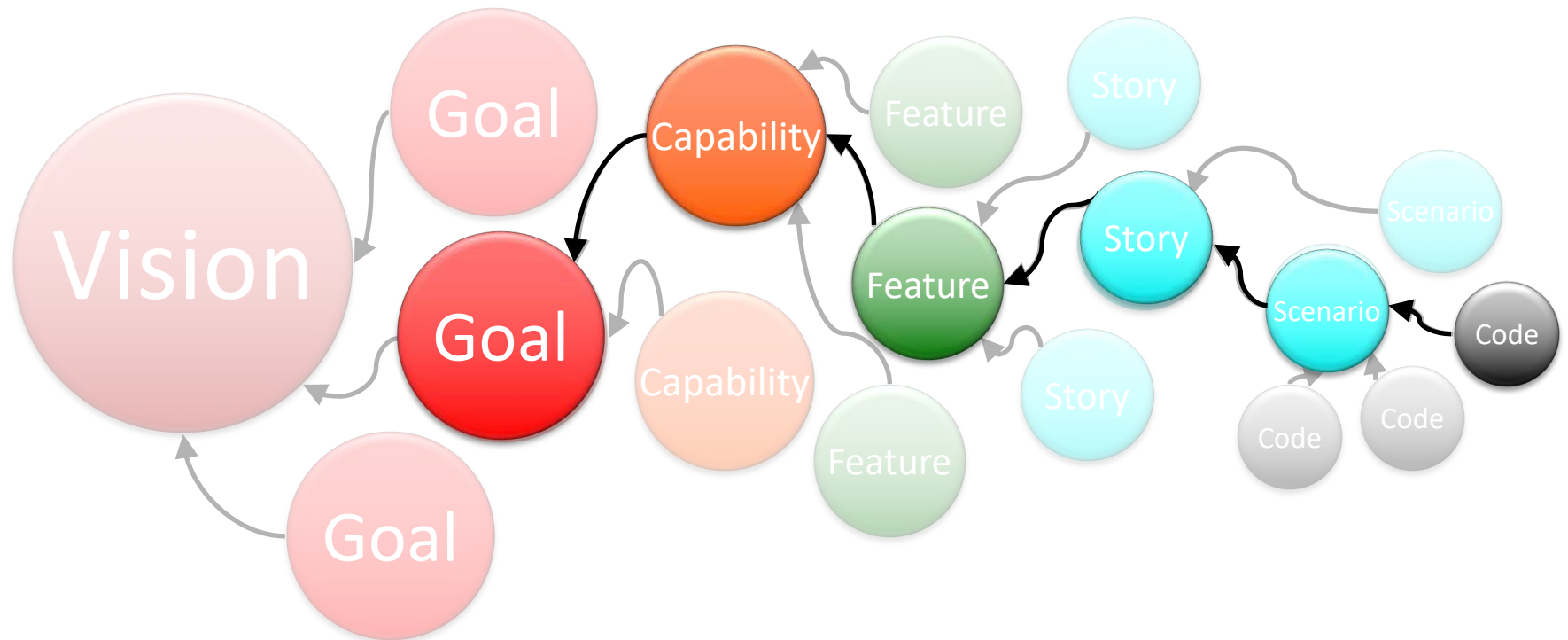


Unless you can explain why not!

High-level, risk-first



Risk (Newest Stuff) First



If your stakeholders
don't trust you,
that's
your biggest risk

Liz Keogh

<http://lizkeogh.com>

@lunivore

