

<https://www.croixlorraine.org>

Working at the Center of the Cyclone

what's on the the other side of deploy

Richard Cook, MD
Adaptive Capacity Labs

Disclaimer & declaration:

The views expressed are those of the speaker alone. The speaker has no conflicts of interest to declare. This presentation does not promote off-label use of any medication or medical device.

Research results

tl;dr: a lot is happening on the other side of ‘deploy’.

- Clarifying **system** changes everything
- Experience with incidents doesn’t fit the paradigm
 - ▶ Everyone is awash in this-might-be-an-incidents, incidents, is-this-big-one?s and OMGs.
 - ▶ Managing incidents is now a *thing*.
 - ▶ Learning from incidents is essential but quite hard.
 - ▶ Ever more dependencies that aren’t yours to deploy.
 - ▶ There is wisdom in incidents.

*This is part of a much larger story.
Not enough time right now.... later* →

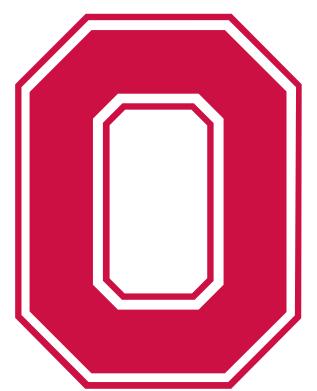
join me Tuesday
at 15:15 in Breakout D

systems research results 1987 though today



Research settings

- surgical operating theaters
- emergency rooms
- intensive care units
- commercial airline cockpits
- military cockpits
- semiconductor wafer fabs
- electrical power generation
- electrical power distribution
- internet-facing business sys



**THE OHIO STATE
UNIVERSITY**

It is *not* surprising that
your system sometimes fails.

What is surprising that
it ever works at all.

RI Cook, 2013

Systems fail.

News

Worldwide airport chaos after computer check-in systems crash



For each event celebrated in the press there are hundreds/thousands more that are known only to a few.

756 automated control systems between 2015 & 16

Every enterprise is experiencing one or more significant events every day.

caused downtime Sunday for HealthCare.gov, the trouble-plagued online insurance marketplace created by the Affordable Care Act.

AWS ↓4/2011
AWS ↓7/2012

The good news is:

Every outage could have been worse.

Work to limit damage / fix the broken
stuff is *remarkably* successful.

The not so good news is:

Something is always broken.

*Broken,
dysfunctional,
& abnormal is
normal here.*

Complexity is increasing.

*All the parts are
changing all the time.
complexity = change*

No one knows what will matter next.

The curse of devops

continuous
deployment requires

- attention
- scrutiny
- *re-calibration*

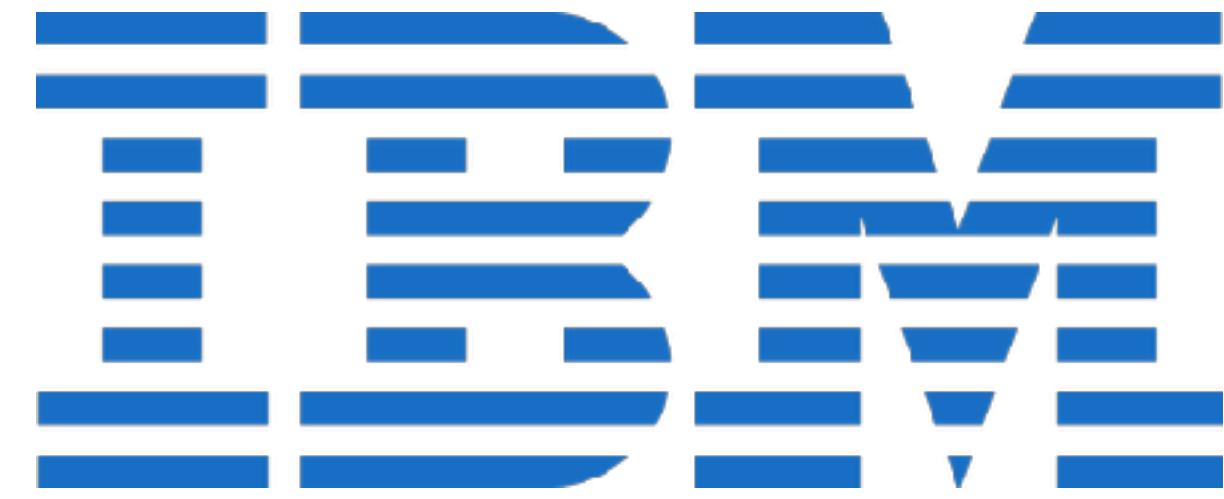
What is re-calibration?
Why do we need it?
How is it being done?
Can we afford it?
How can we do it better?

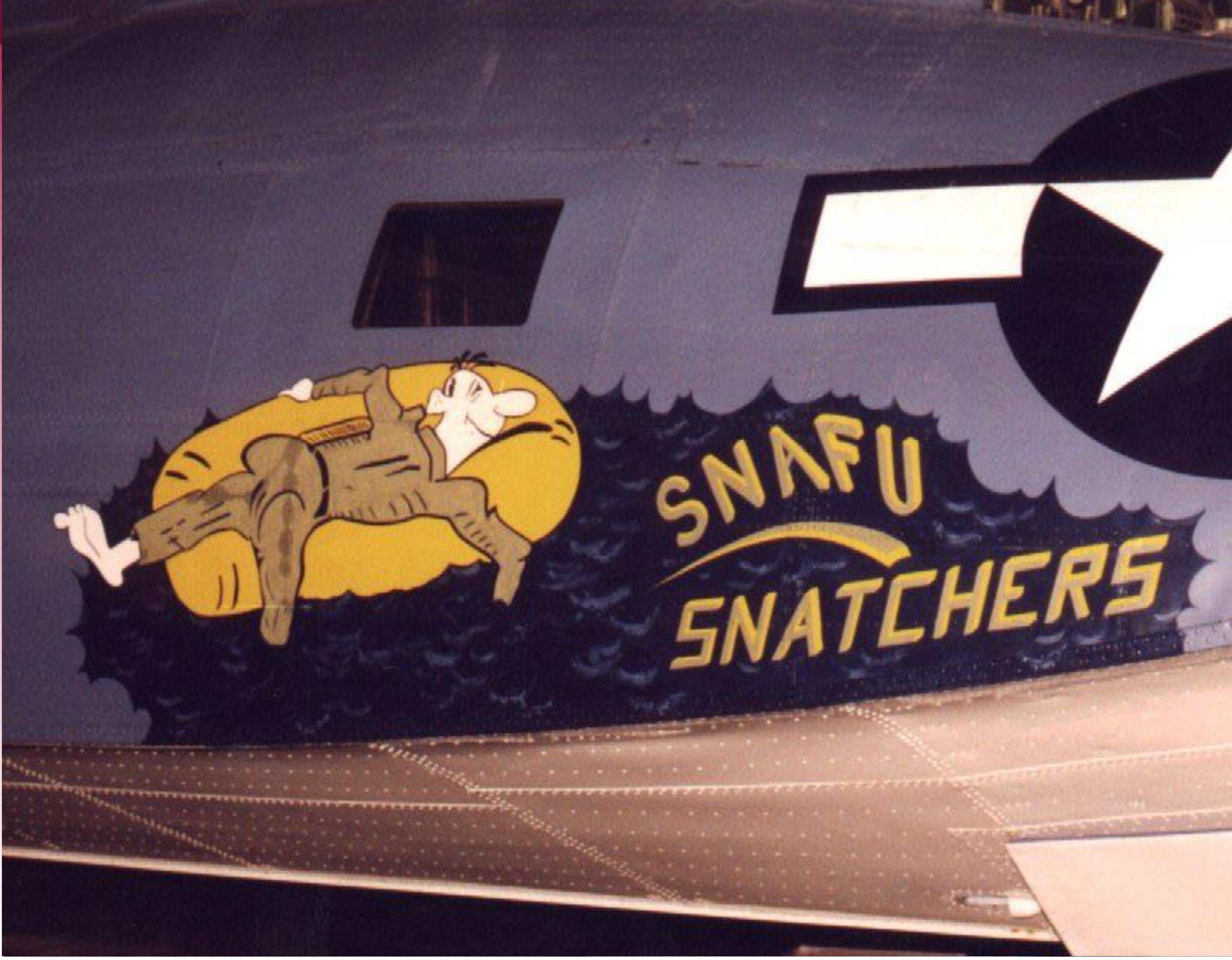
Cause for optimism

- Other domains have similar problems.
- Lots of solid research results available
- Resilience already present & used
- Diverse efforts underway



consortium members





U.S. Air Force Museum, Dayton, OH

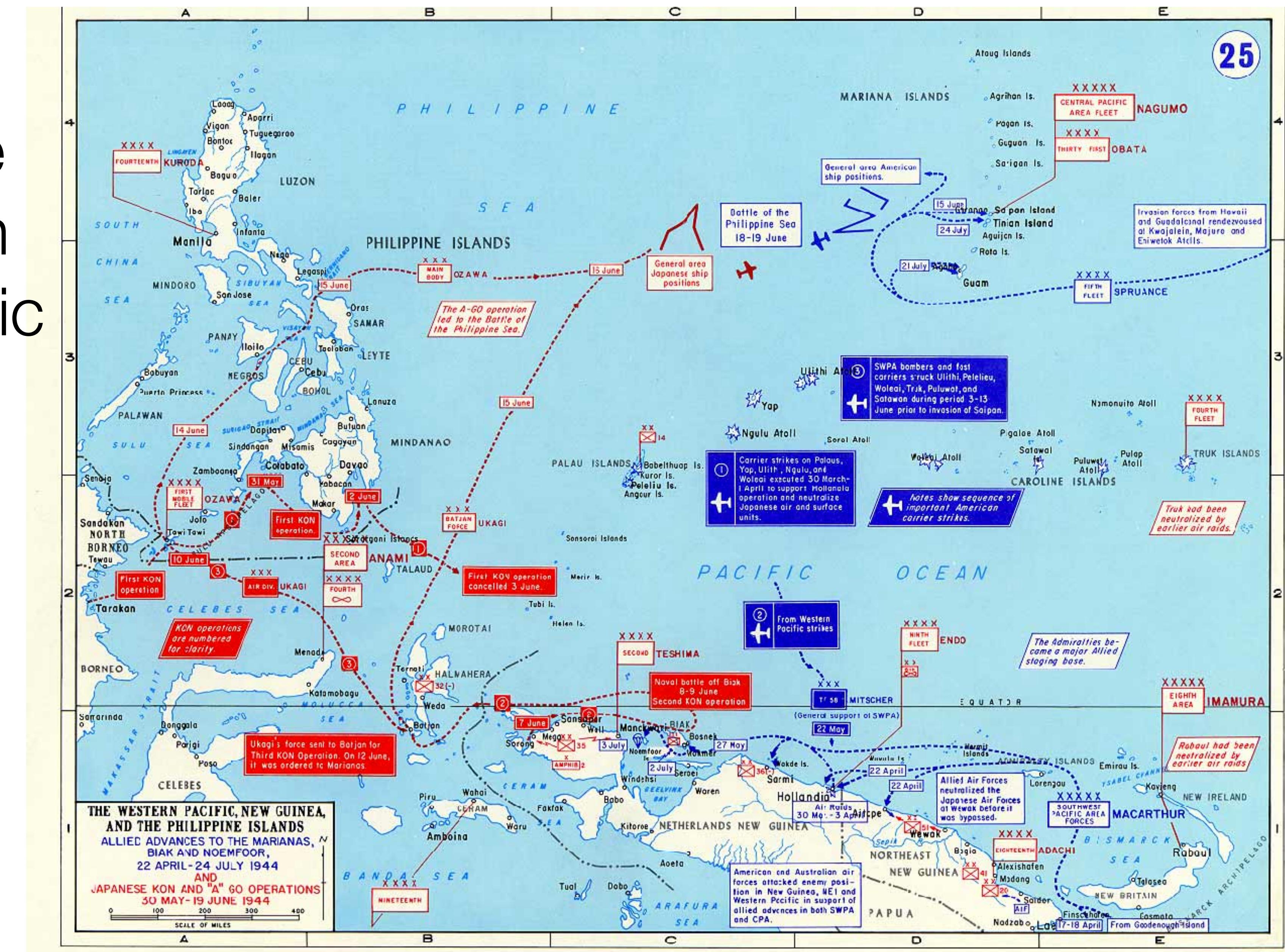
Situation
Normal:
All
F&#\$ed
Up

The 2nd Emergency
Rescue Squadron
SNAFU SNATCHERS

www.pbyrescue.com

Air-sea rescue during WWII in the South Pacific

1944-45



PBY-5 “flying boat” a.k.a. *tooling*



- Expect things will go wrong (SNAFU).
- Build recovery with continuous, rapid learning.



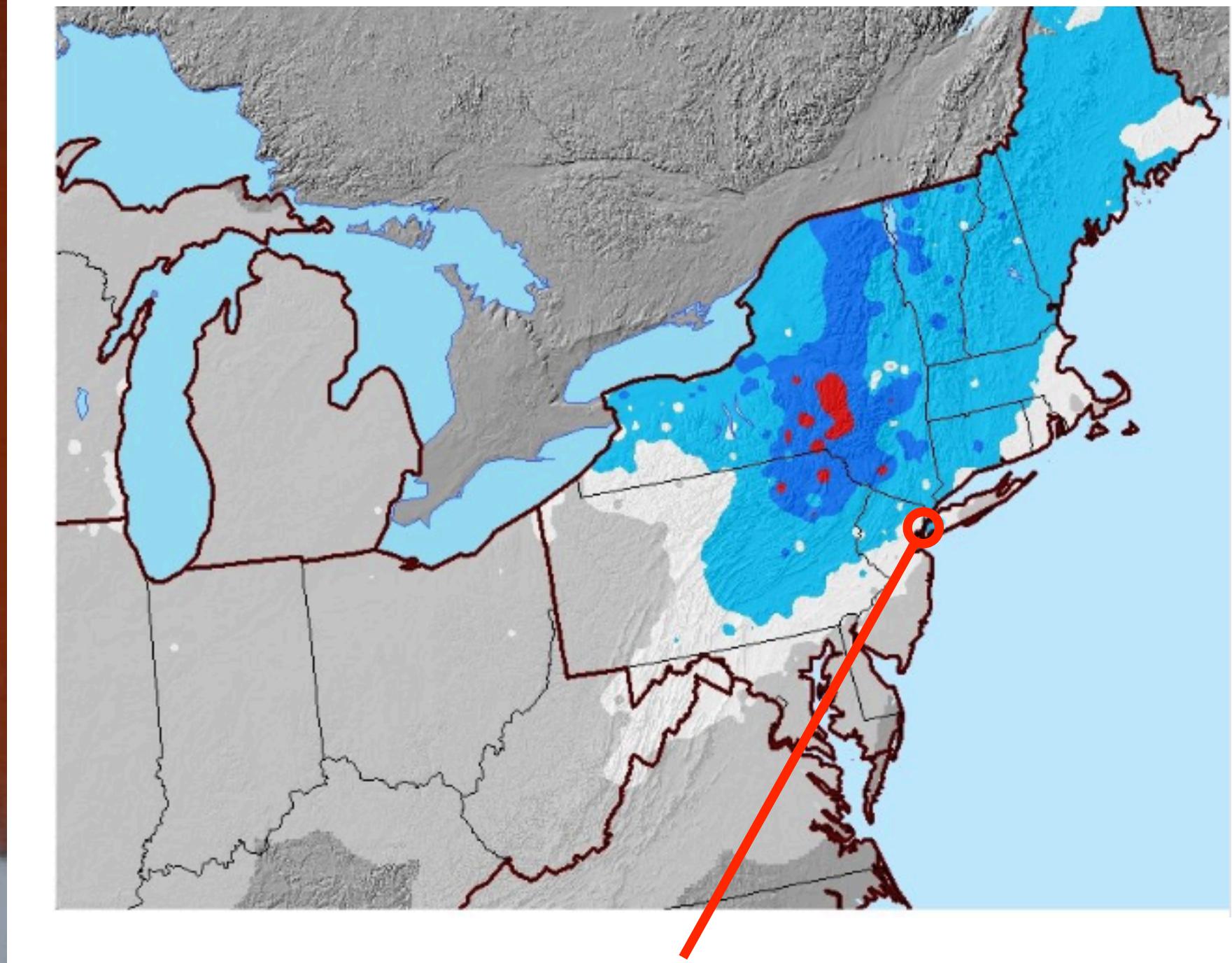
You build things differently
when you expect them to fail!

Making use of the PBY's unique capability required
a whole organization deliberately tuned to the mission.

STELLA: Report from SNAFUcatchers Workshop on Coping With Complexity



Winter storm Stella
12-15 March 2017



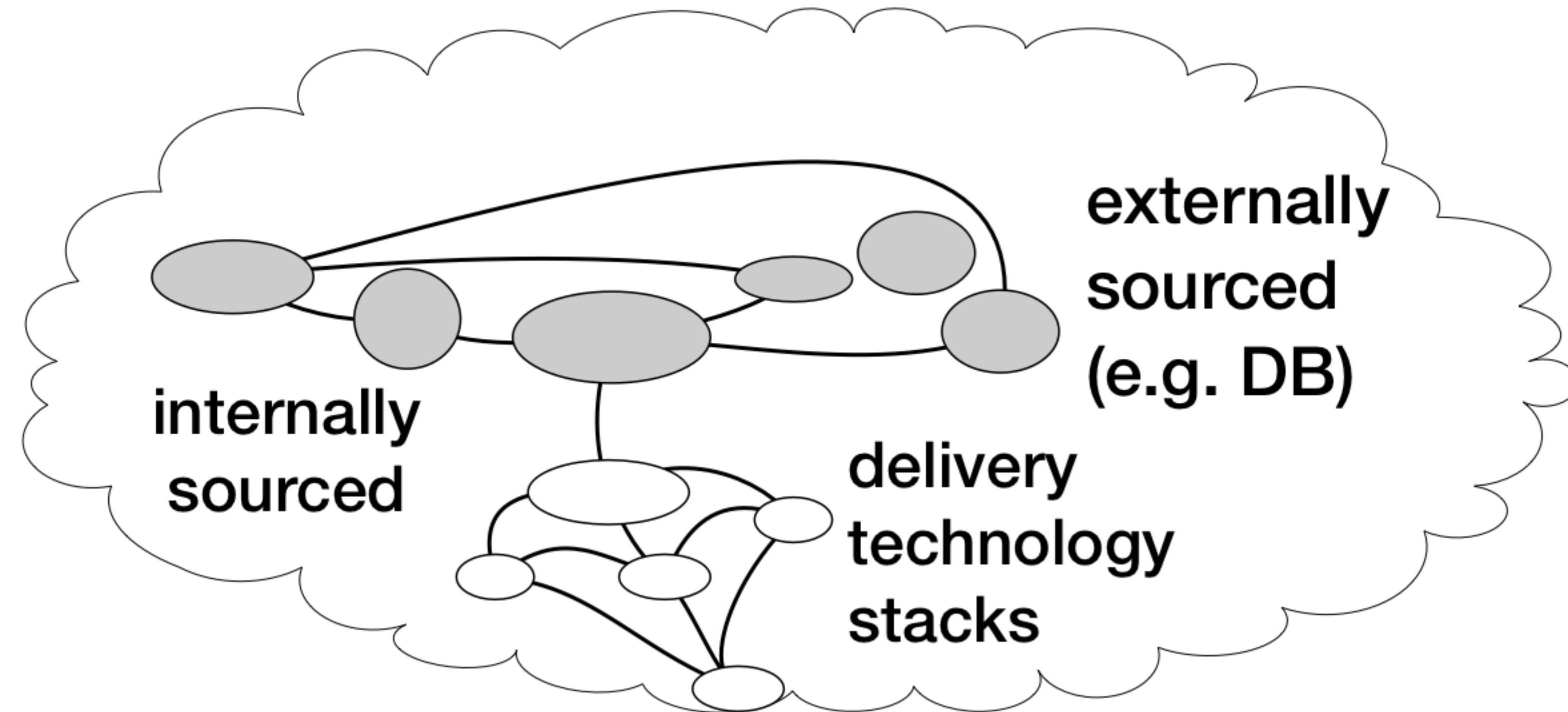
New York city
40 cm snow
 -6°C

Research results

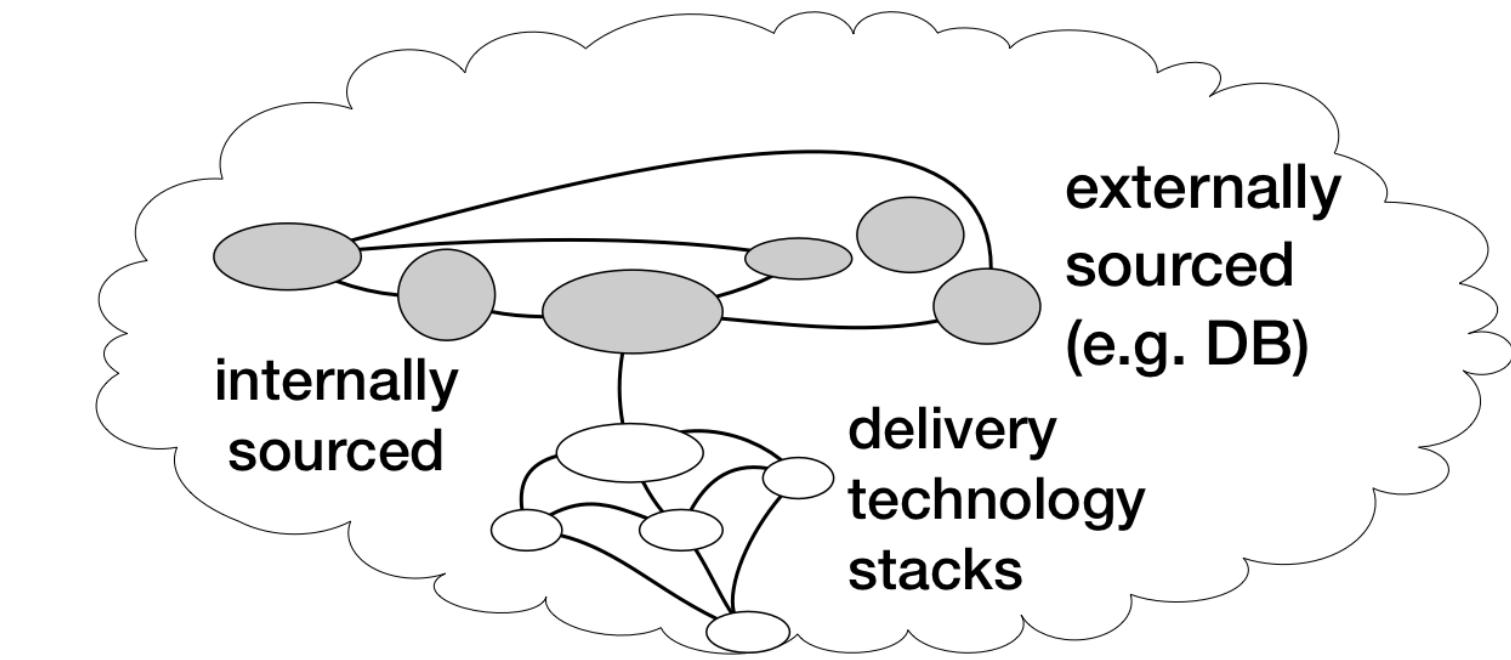
tl;dr: a lot is happening on the other side of ‘deploy’.

- Clarifying **system** changes everything
- Experience with incidents doesn’t fit the paradigm
 - ▶ Everyone is awash in this-might-be-an-incidents, incidents, is-this-big-one?s and OMGs.
 - ▶ Managing incidents is now a *thing*.
 - ▶ Learning from incidents is essential but quite hard.
 - ▶ Ever more dependencies that aren’t yours to deploy.
 - ▶ There is wisdom in incidents.
- Complexity *is* change!

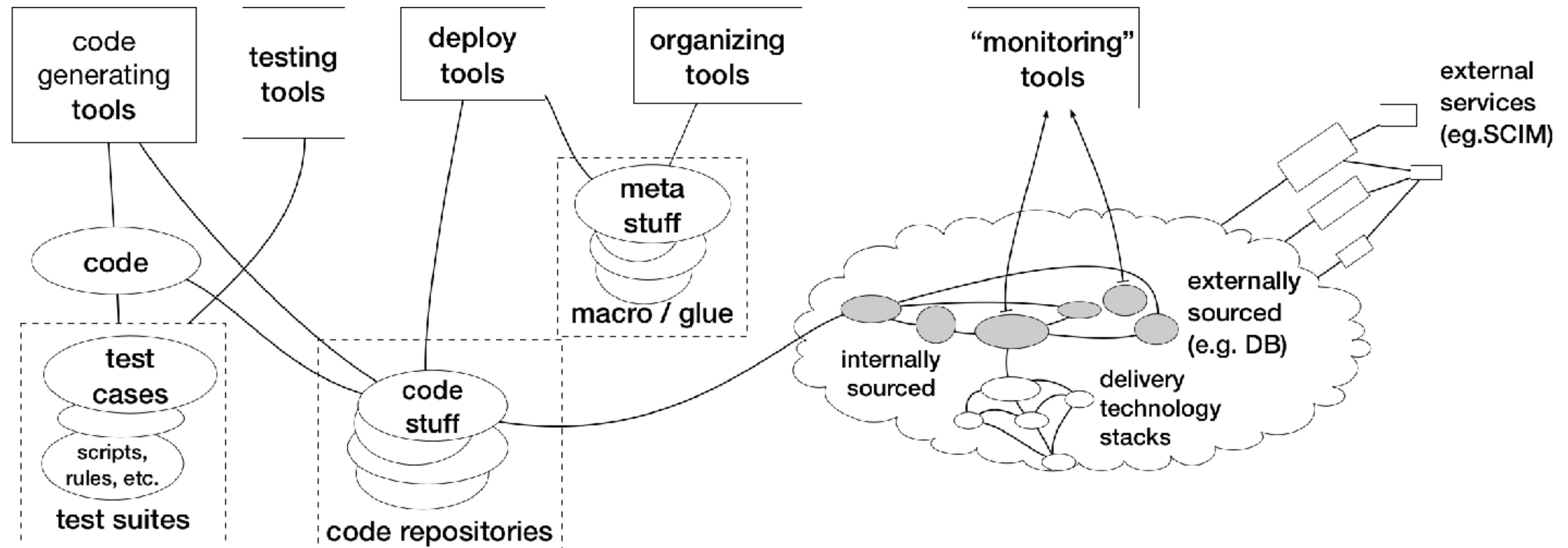
the system



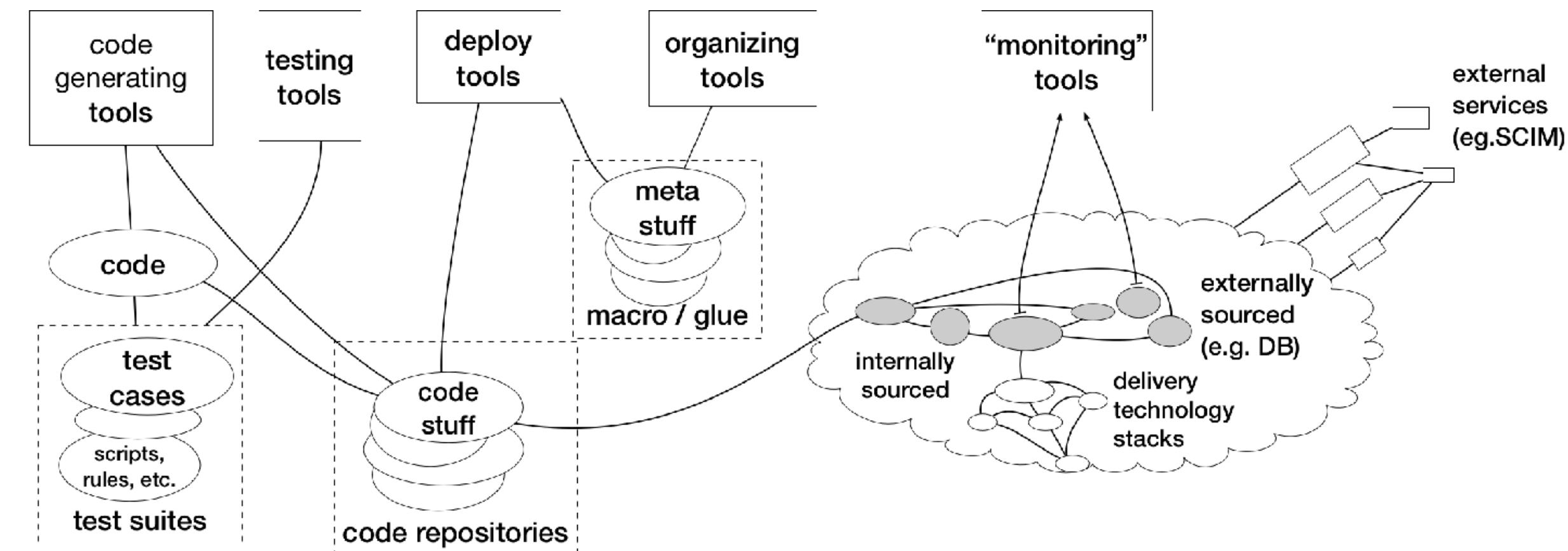
the system



the system

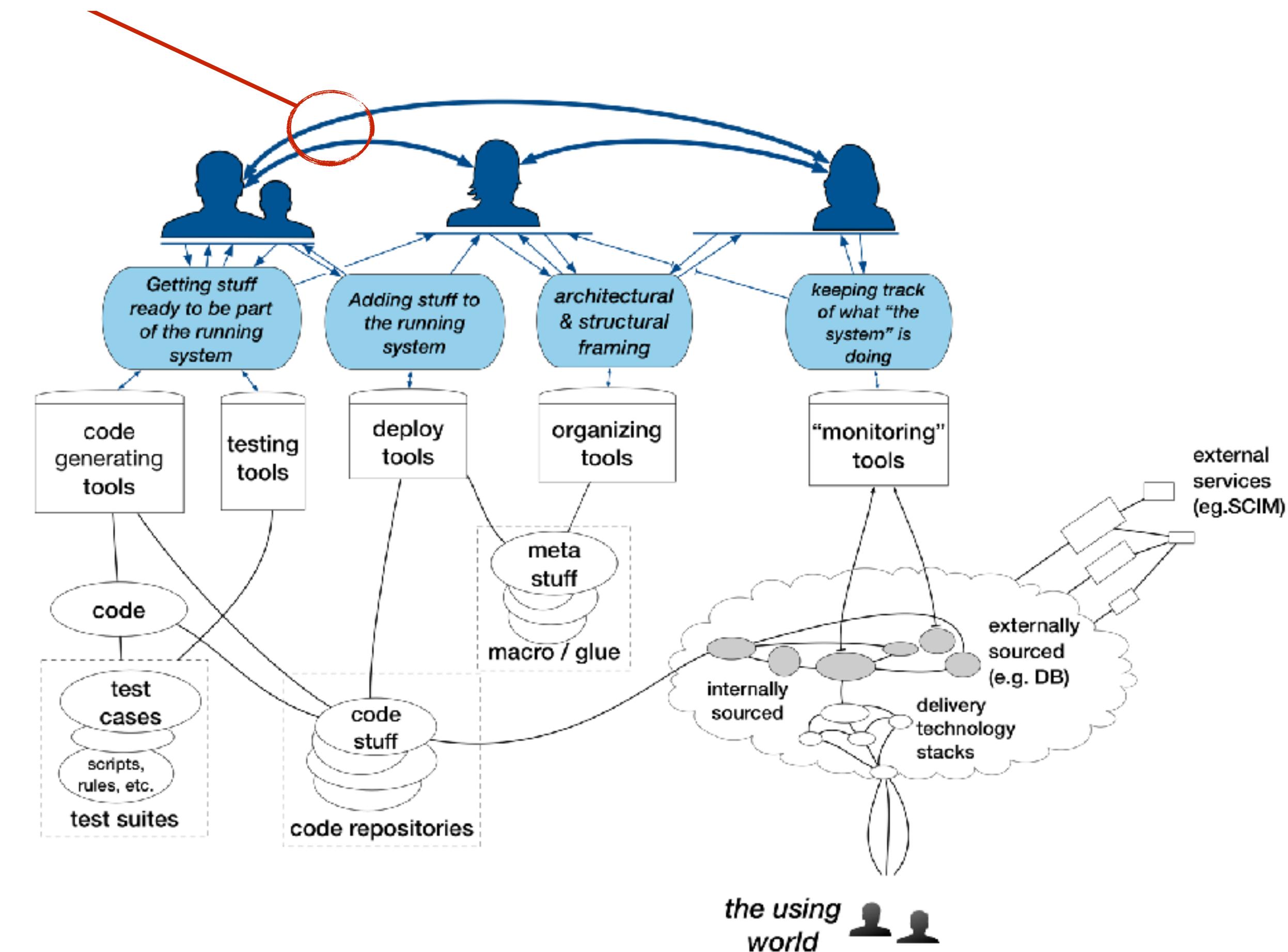


the system



the system

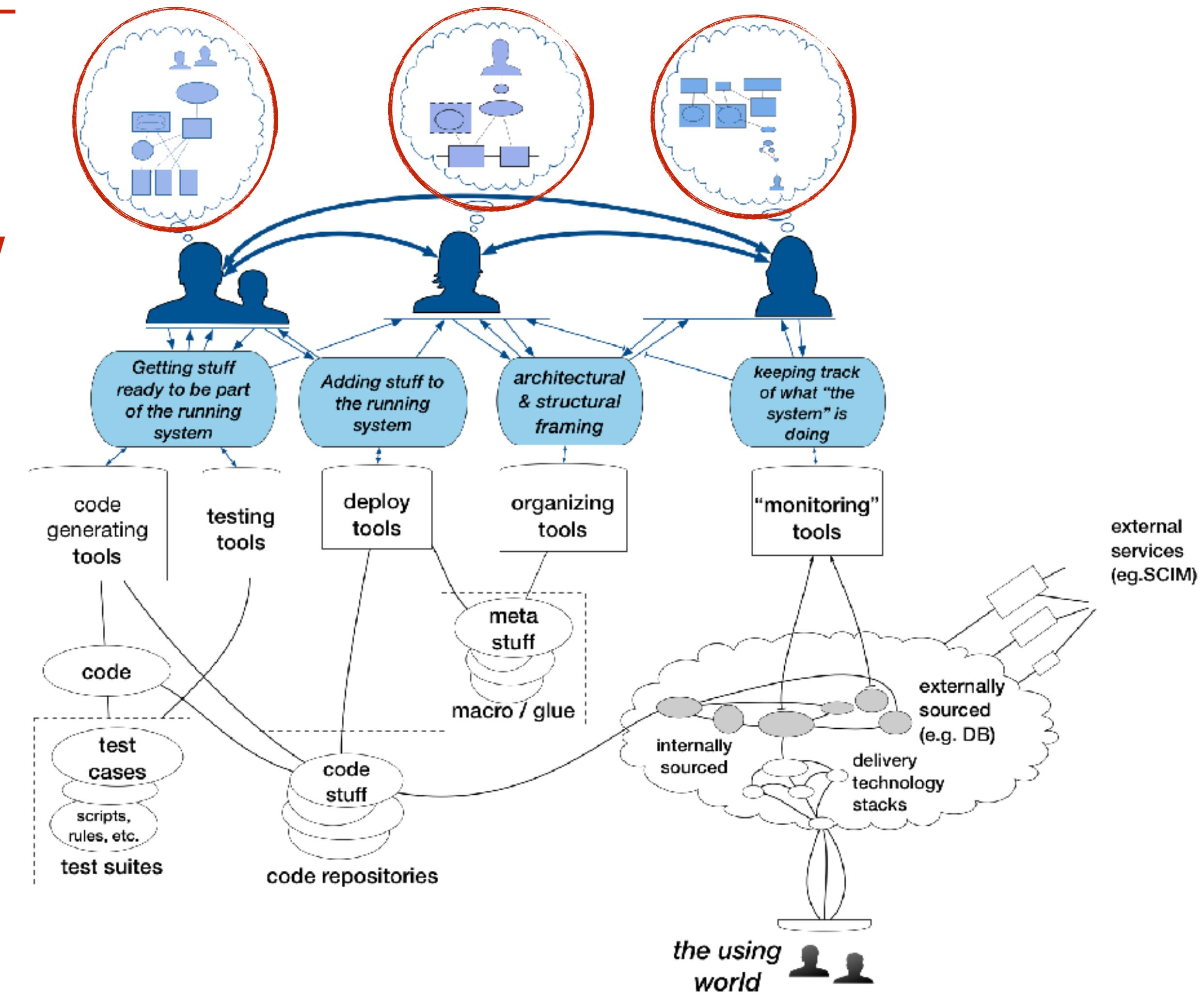
*lots of communications
between problem solvers*



the system

mental models

- unique
- sometimes deep
- sometimes shallow
- always incomplete
- constantly tested
- actively calibrated

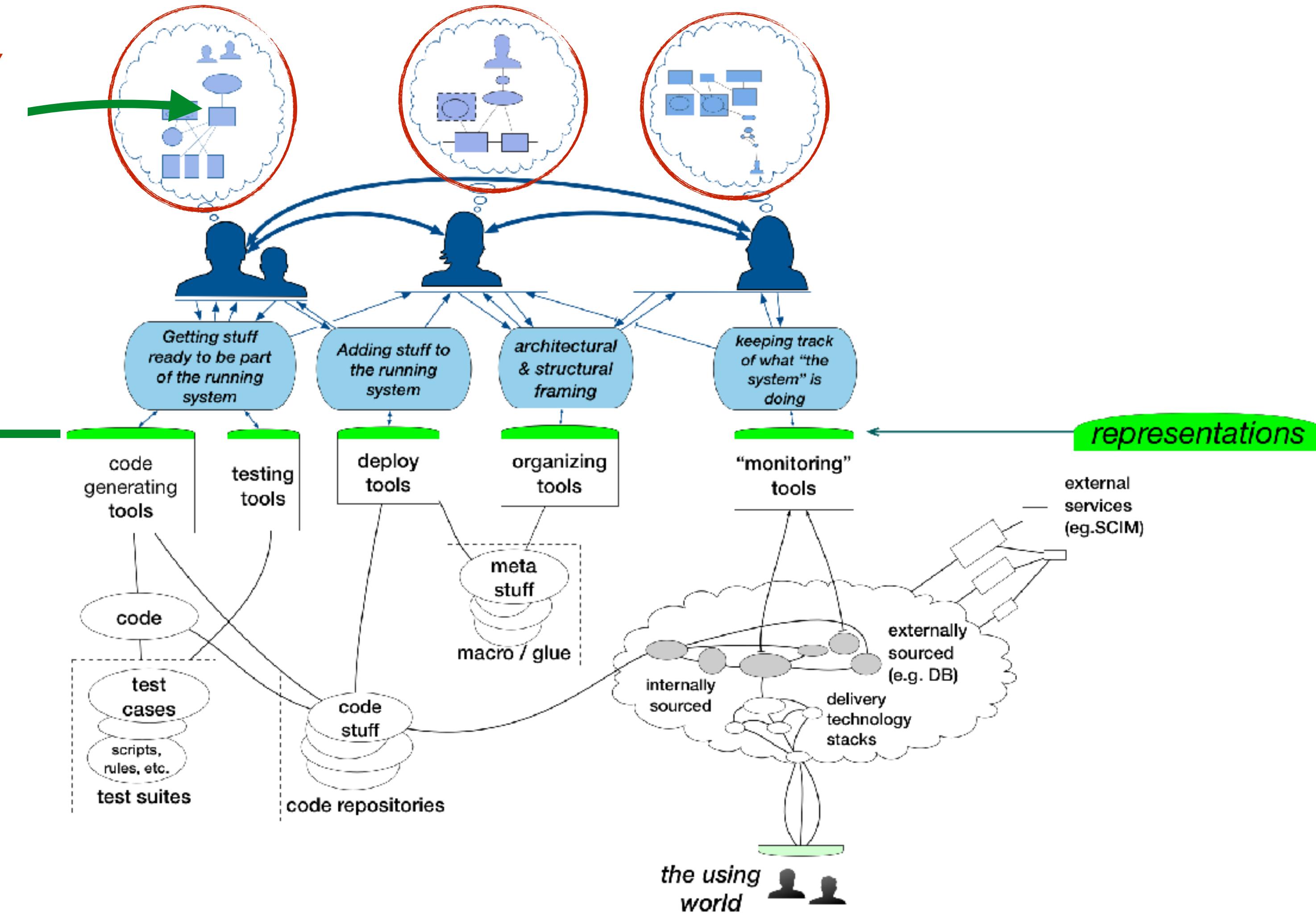


the system

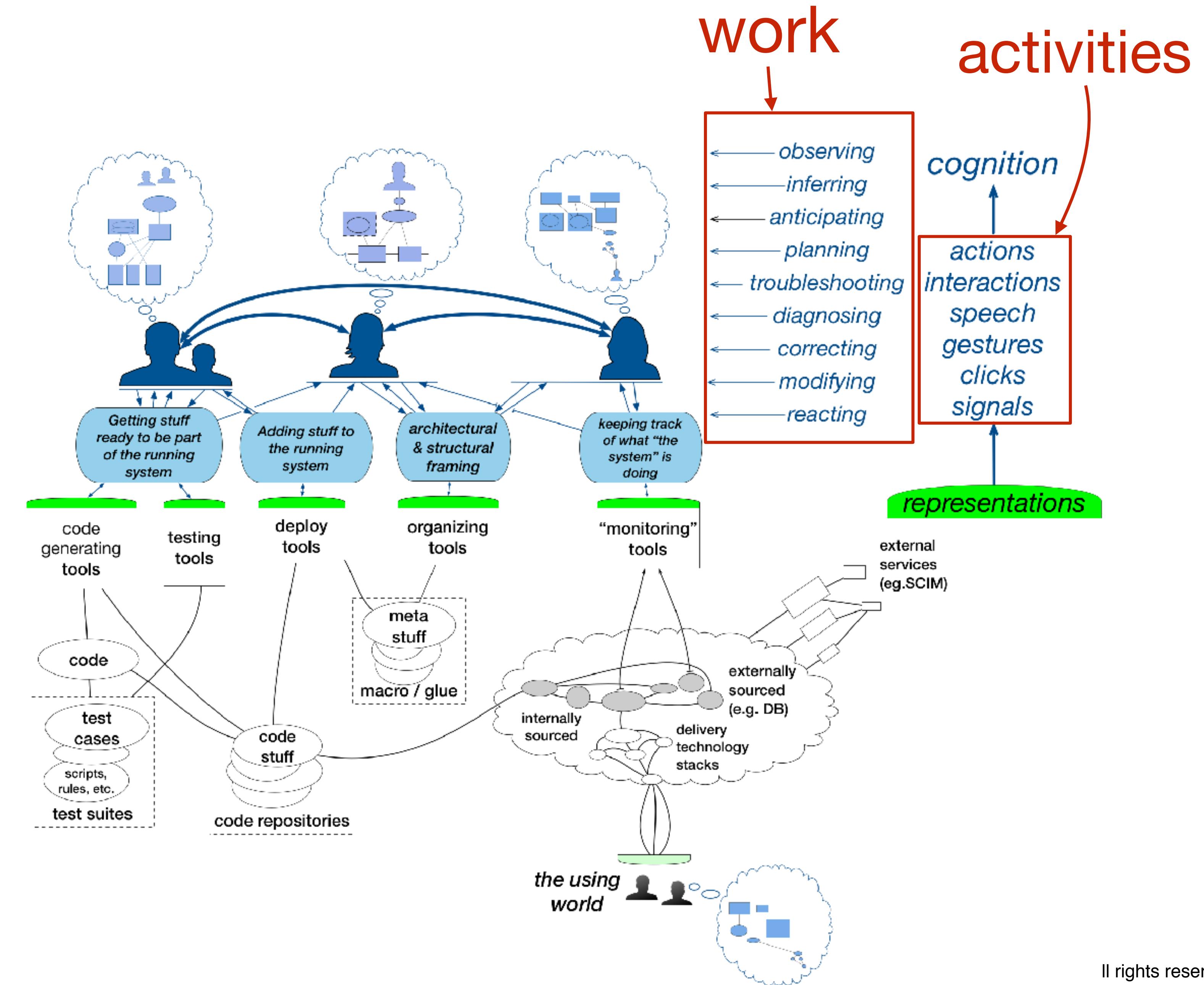
*mental models
are informed by
representations*

```
Mon Apr 15 7:44 PM, 2018
INDU INDUSTRIES (TM) TERMINLINK PROTOCOL
ENTER PASSWORD NOW
2 ATTEMPT(S) LEFT: 1 :
0x3400 d0gsth0: 0x55F0 f4044d00
0x5700 d0717e0 0x57F0 j1X5200
0x5800 g1<4sds 0x59F0 (~>4"ad
0x5900 0>8001 0x59F0 <=>0000
0x5800 _(>)F073 0x59F0 asdha75h
0x5800 (A>0000 0x59F0 k1F5C0KF
0x5800 0x5217L 0x5CF0 (<>)>000G
0x5800 )(<017: 0x5DF0 R10004000

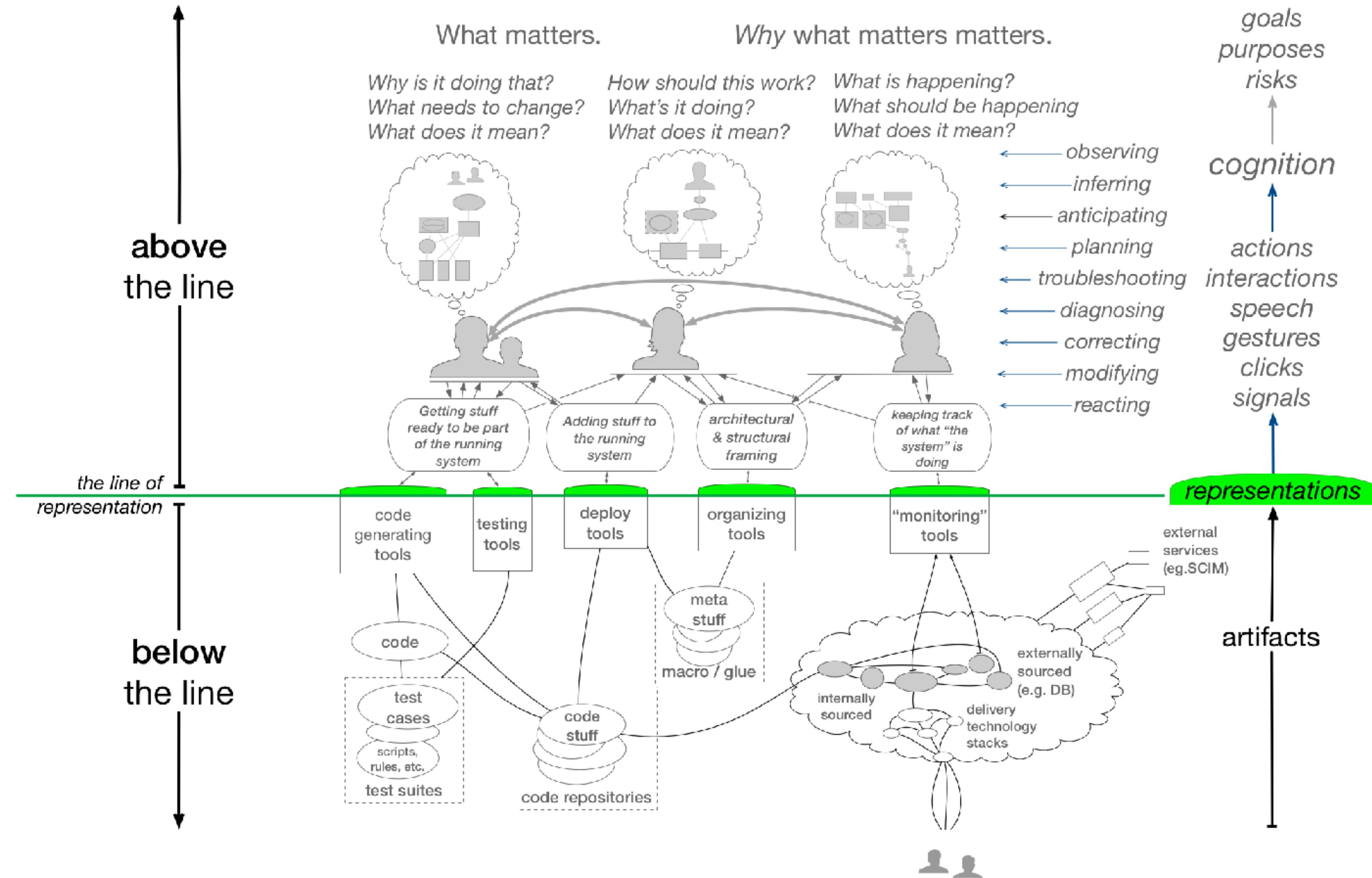
Hello, &/Retrobulletstation!
w/tergar17
```



the system

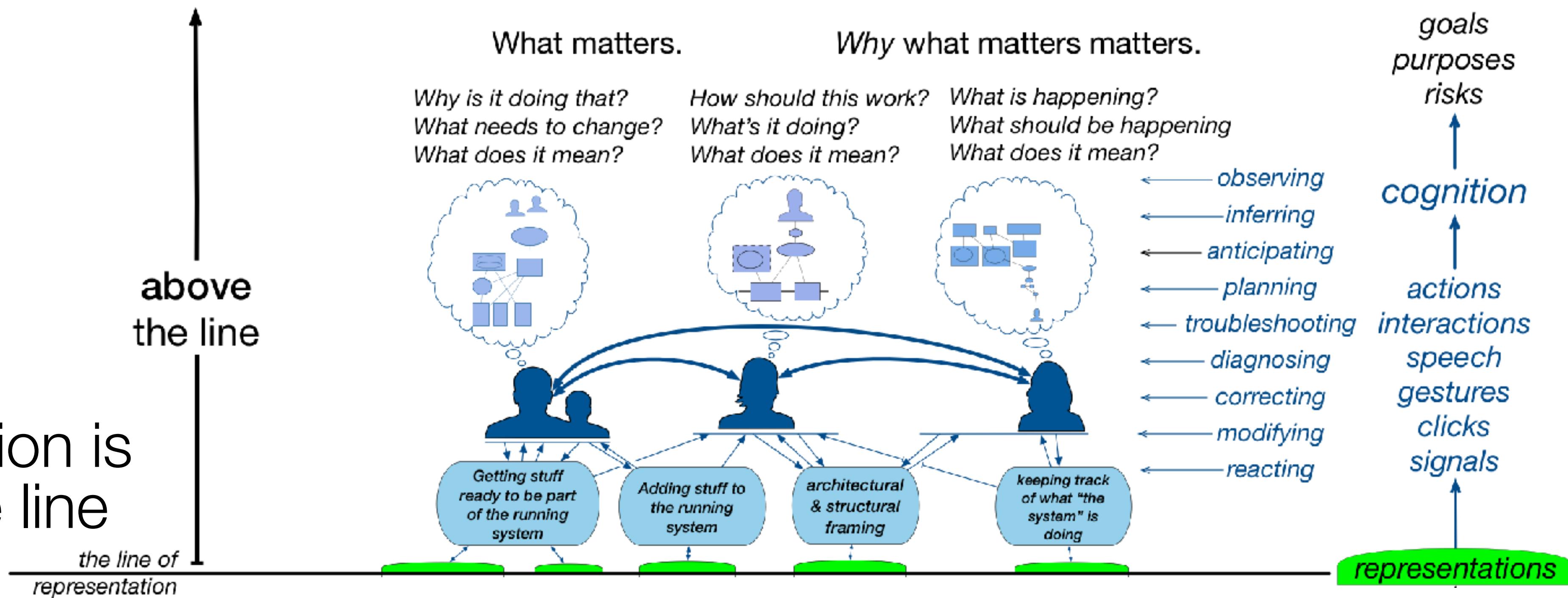


the system



the system

all the action is
above the line



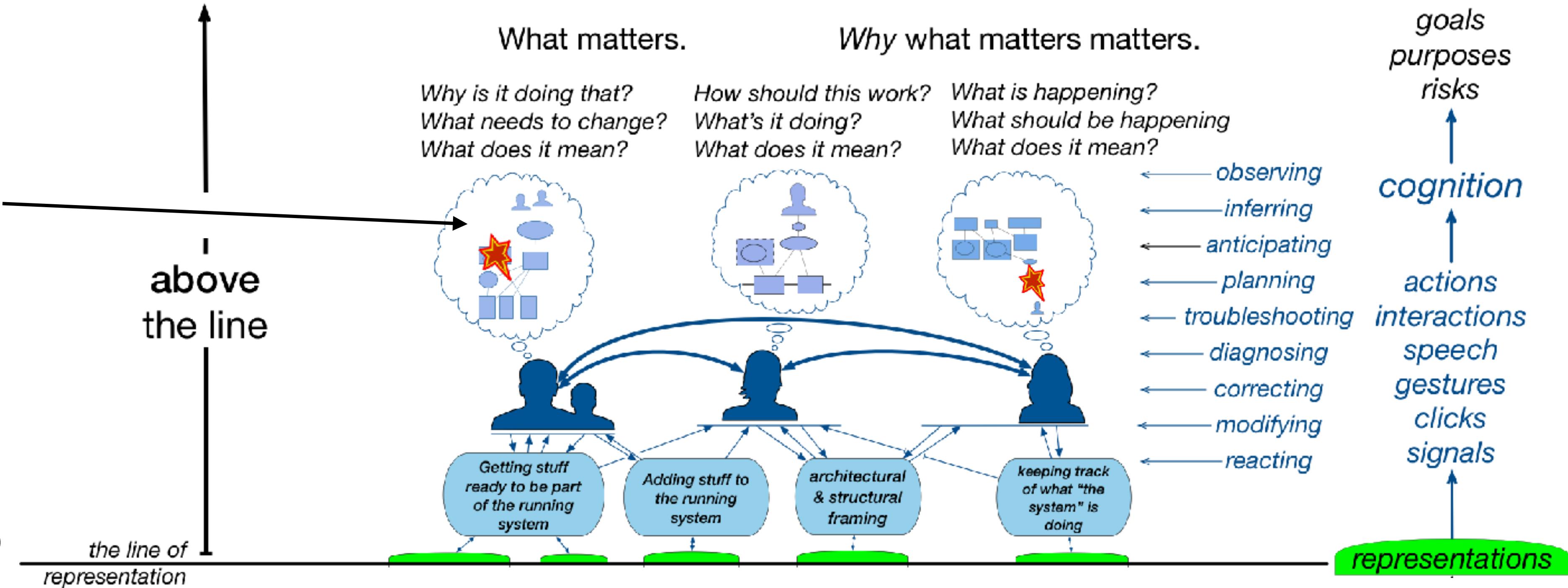
below the line is *never* visible



the system

incidents
occur
here

not here



Research results

tl;dr: a lot is happening on the other side of ‘deploy’.

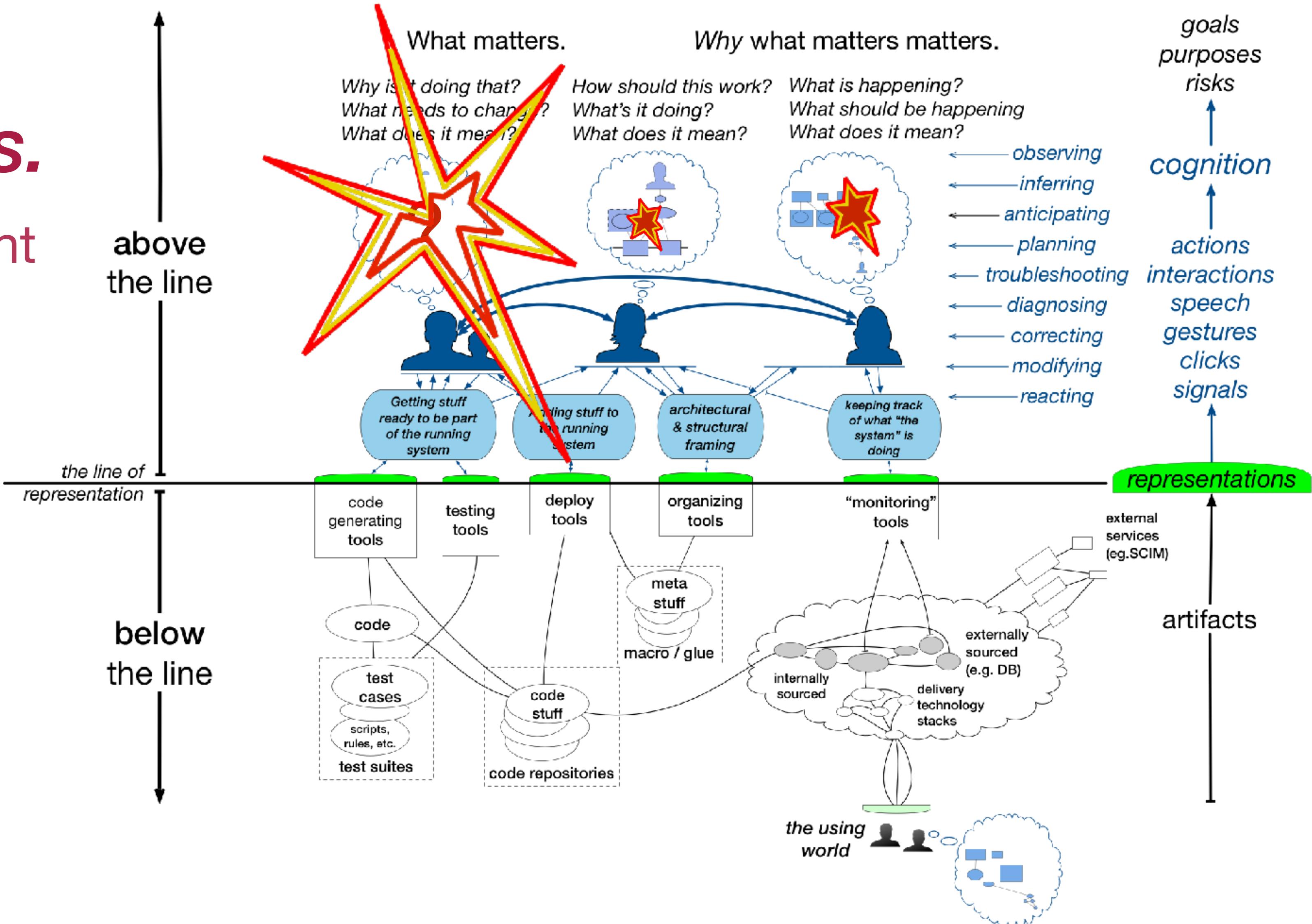
- Clarifying **system** changes everything
- Experience with incidents doesn’t fit the paradigm
 - ▶ Everyone is awash in this-might-be-an-incidents, incidents, is-this-big-one?s and OMGs.
 - ▶ Managing incidents is now a *thing*.
 - ▶ Learning from incidents is essential but quite hard.
 - ▶ Ever more dependencies that aren’t yours to deploy.
 - ▶ There is wisdom in incidents.
- Complexity *is* change!

experience with incidents

The system is awash in incidents.

- ▶ this-might-be-an-incident
- ▶ incident
- ▶ is-this-big-one?
- ▶ rare OMGs

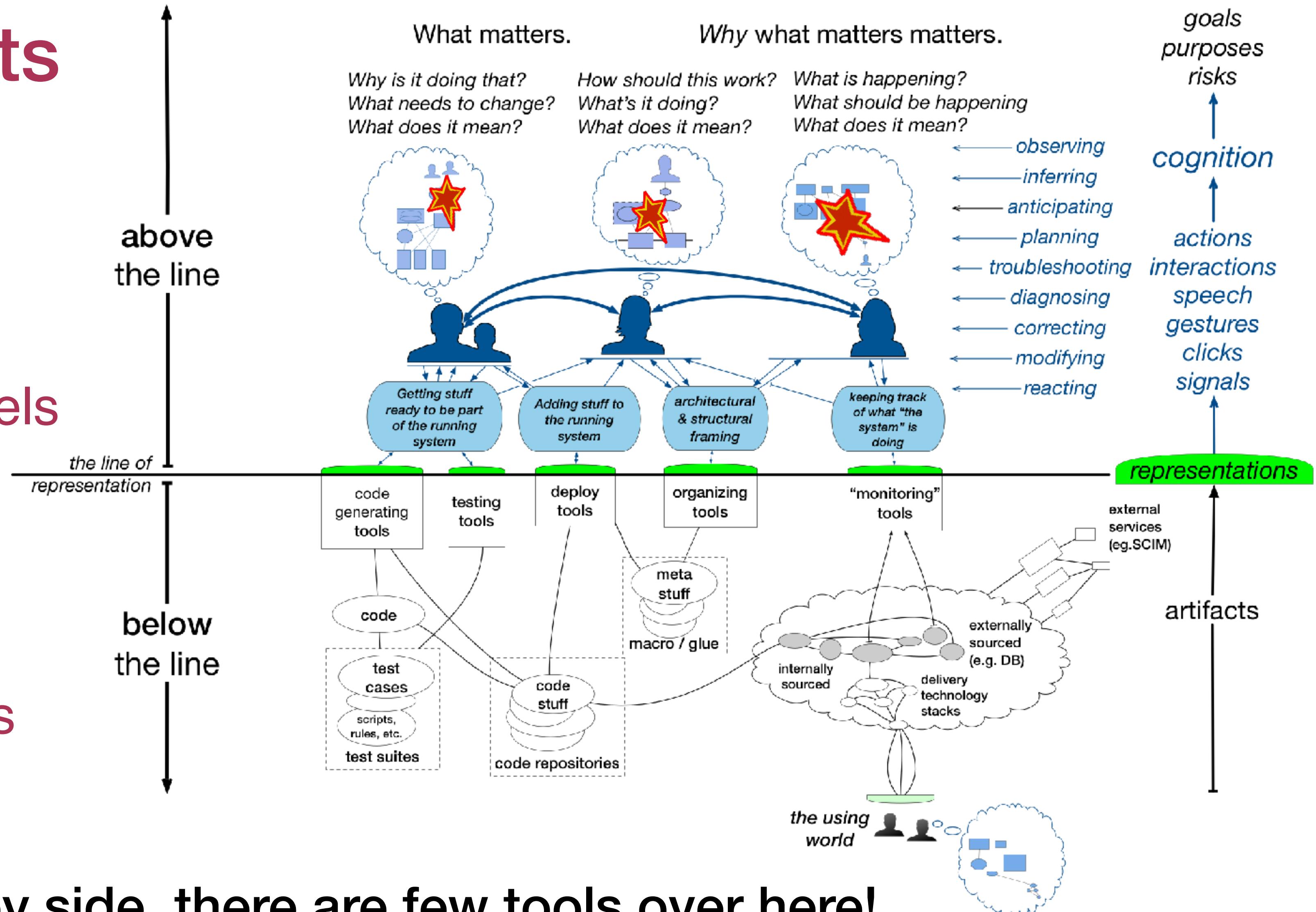
Ordinary firms are experiencing 1 to 5 acknowledged events per day.



experience with incidents

Managing incidents is now a *thing*.

- As many as 40 people involved in a response
- Multiple dedicated channels in IRC
- Formal role assignments
- Rules of behavior
- Formal escalation policies
- Automation experiments

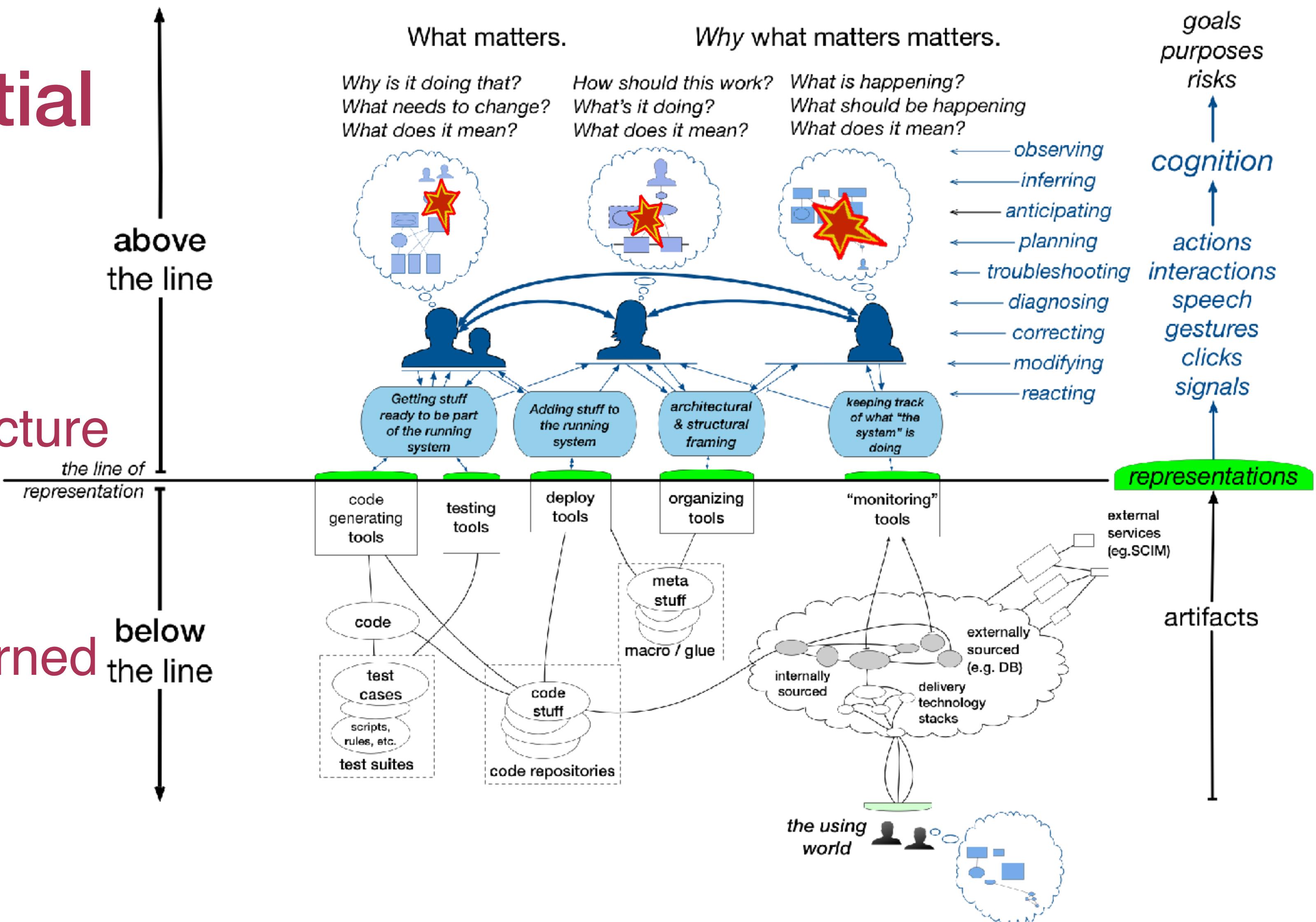


Unlike the deploy side, there are few tools over here!

experience with incidents

Learning from incidents is essential but quite hard.

- Retrospectives poorly prepared.
- Focus limited to micro-fracture repair.
- Often a pro-forma chore
- No sharing of lessons learned
mainly because no lessons were learned

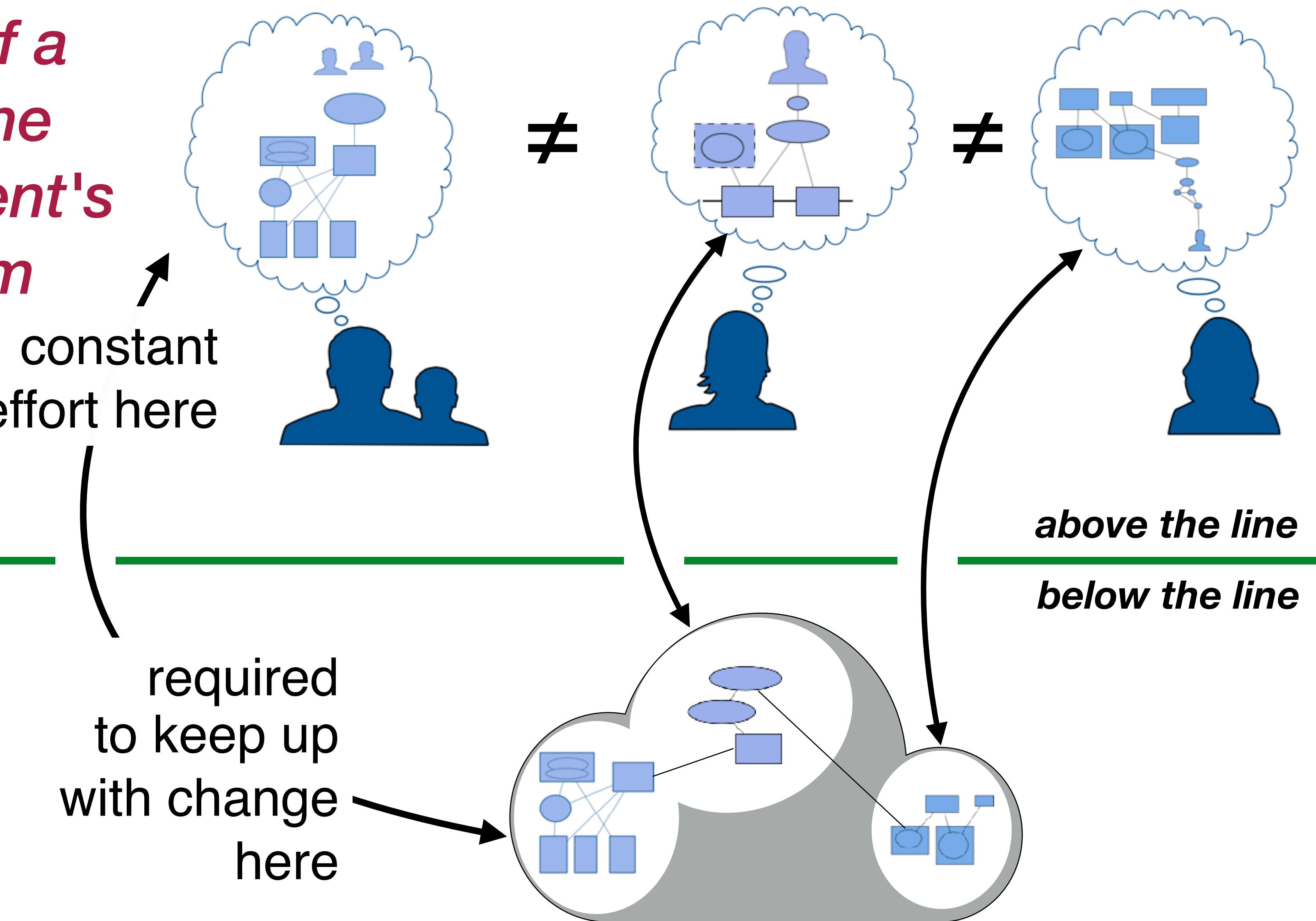


experience with incidents

the re-calibration problem

As the complexity of a system increases, the accuracy of any agent's model of that system decreases.

David Woods, 2017



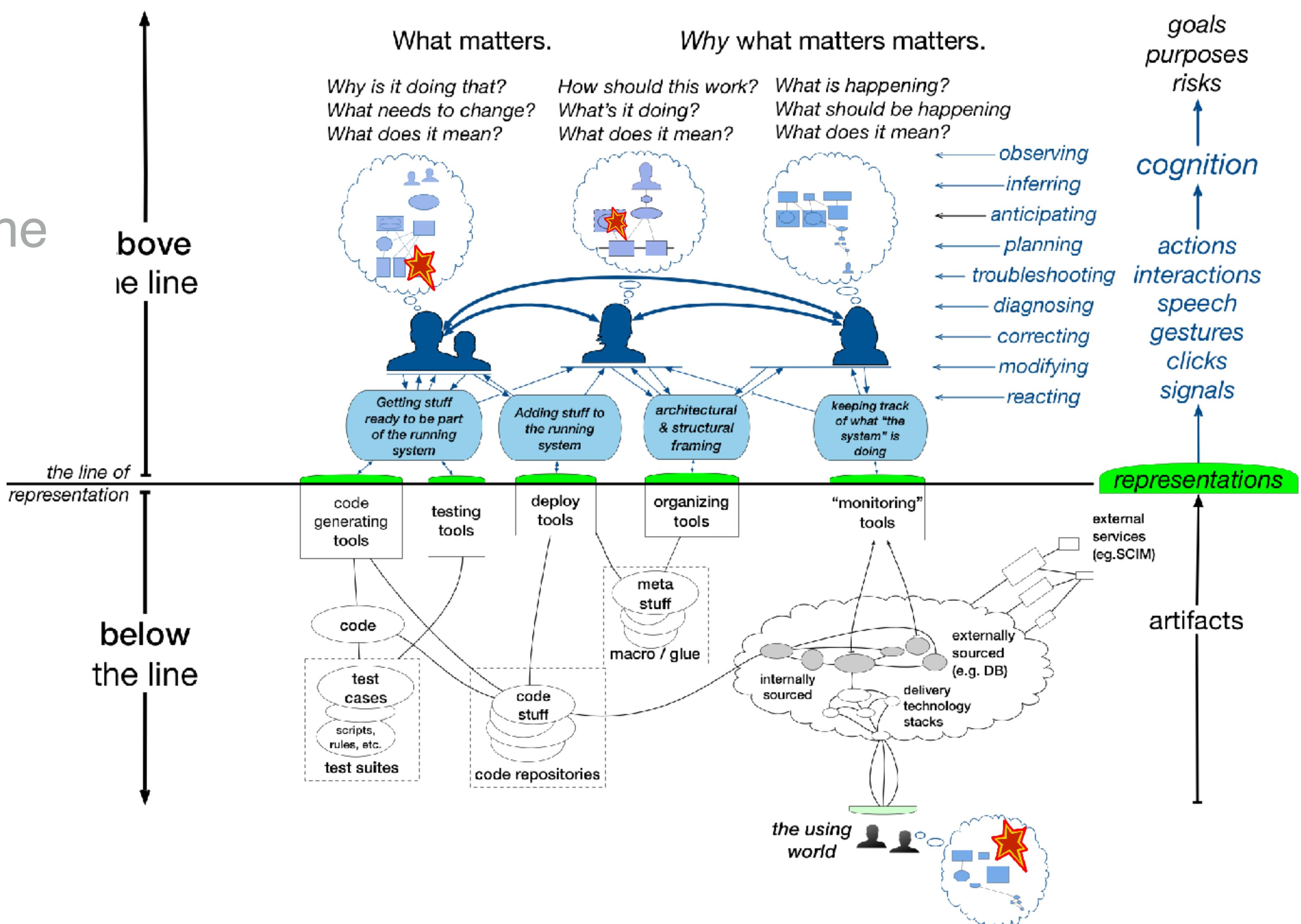
experience with incidents

There is wisdom in incidents.

★ are messages sent from the system about how it *really* works.

★ point to stale, shallow, or inaccurate mental models.

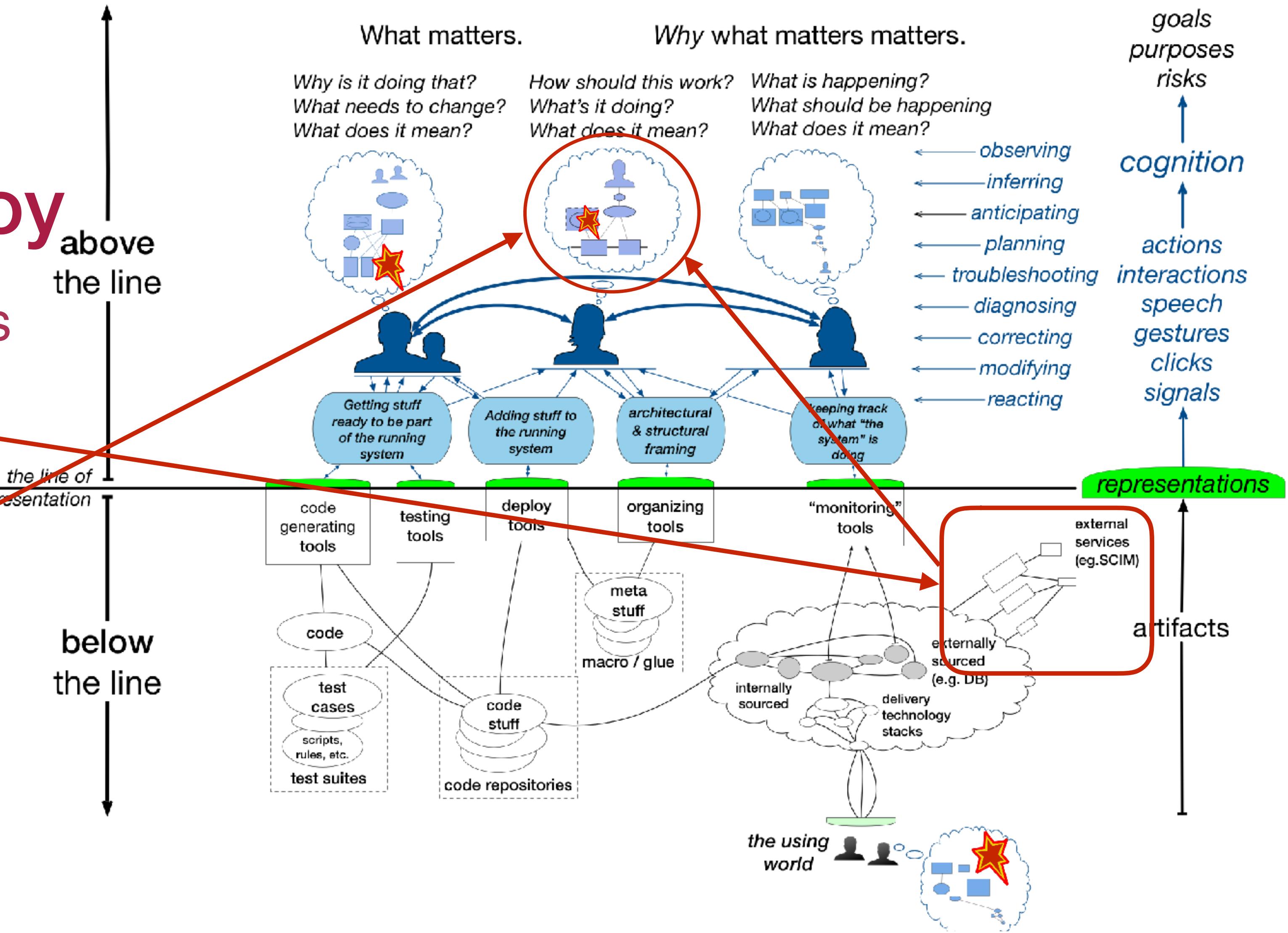
★ are prompts to engage in calibration /re-calibration.



experience with incidents

Ever more dependencies that aren't yours to deploy

- Fewer deploy-related events
- More SaaS-type problems
- Difficult to build MMs of external functionality and it's changes
- Step change in complexity unmatched by monitoring / tools

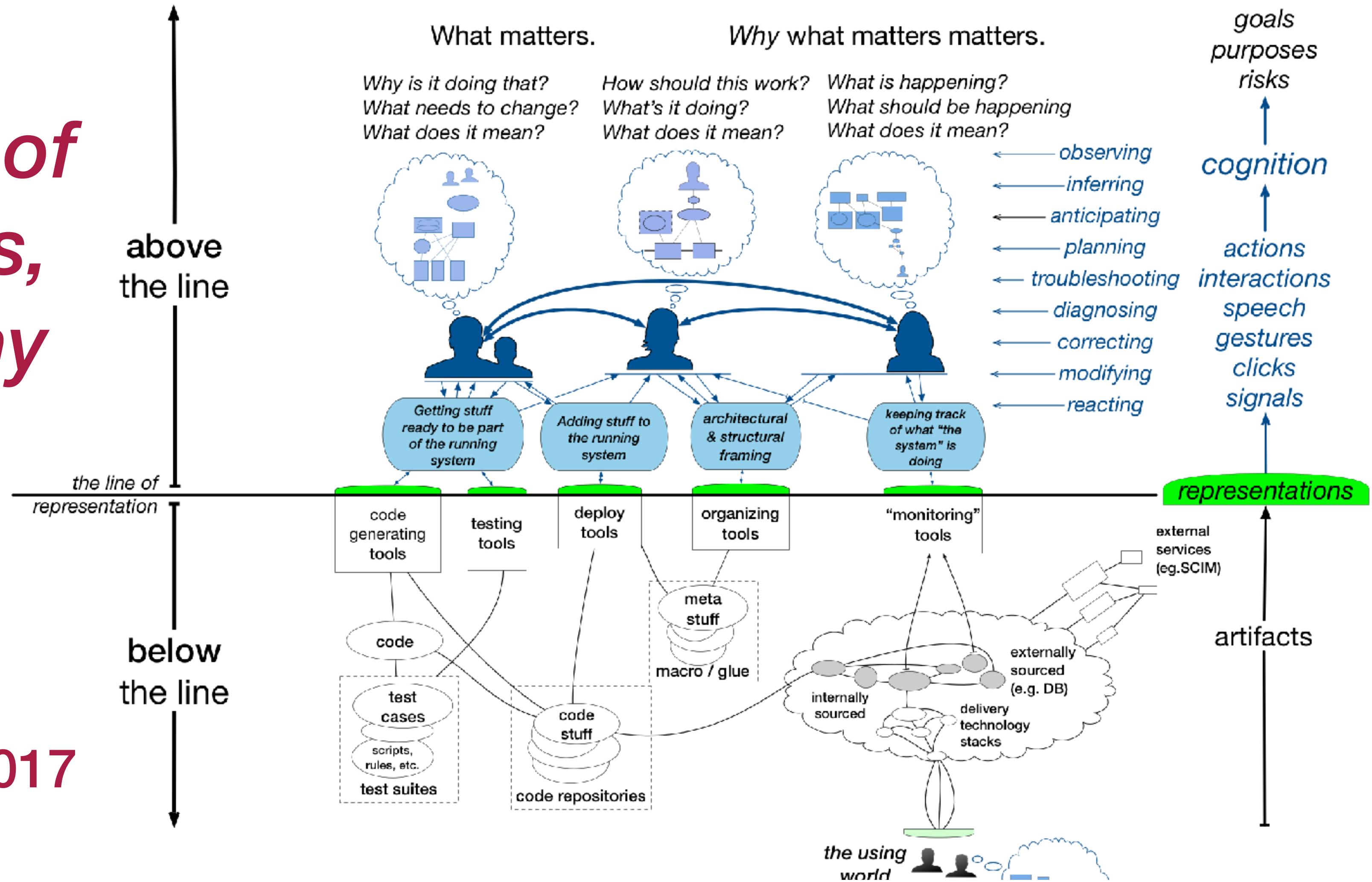


The look-back / roll-back technique is becoming less useful. reserved

the calibration / re-calibration problem

As the complexity of a system increases, the accuracy of any agent's model of that system decreases.

David Woods, 2017



It's becoming harder to keep MMs relevant and fresh.

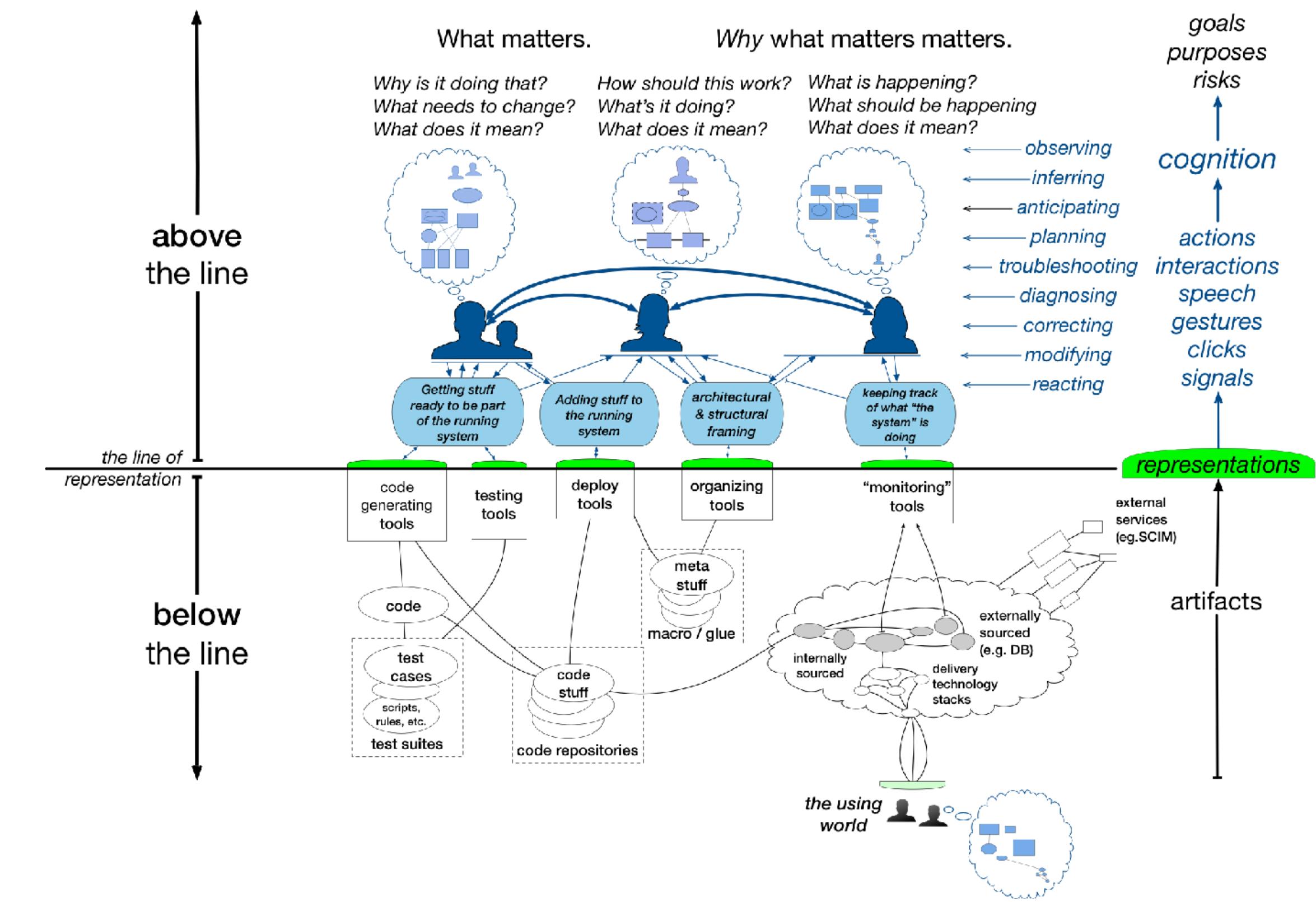
Research results

tl;dr: a lot is happening on the other side of ‘deploy’.

- Clarifying **system** changes everything
- Experience with incidents doesn’t fit the paradigm
 - ▶ Everyone is awash in this-might-be-an-incidents, incidents, is-this-big-one?s and OMGs.
 - ▶ Managing incidents is now a *thing*.
 - ▶ Learning from incidents is essential but quite hard.
 - ▶ Ever more dependencies that aren’t yours to deploy.
 - ▶ There is wisdom in incidents.
- **Complexity is change!**

Complexity is change!

- Structure & function below and above the line are interwoven.
- Any analysis exclusively on one side will be deeply flawed.
- The changing *pattern* of incidents points to the places where recalibration will be valuable.
- Change and complexity are the same entity. Complexity is impossible without change. Change flows from complexity.



Complexity is change!

INT. J. SYSTEMS SCI., 1970, VOL. 1, NO. 2, 89-97

Every good regulator of a system must be a model of that system†

ROGER C. CONANT

Department of Information Engineering, University of Illinois,
Box 4348, Chicago, Illinois, 60680, U.S.A.

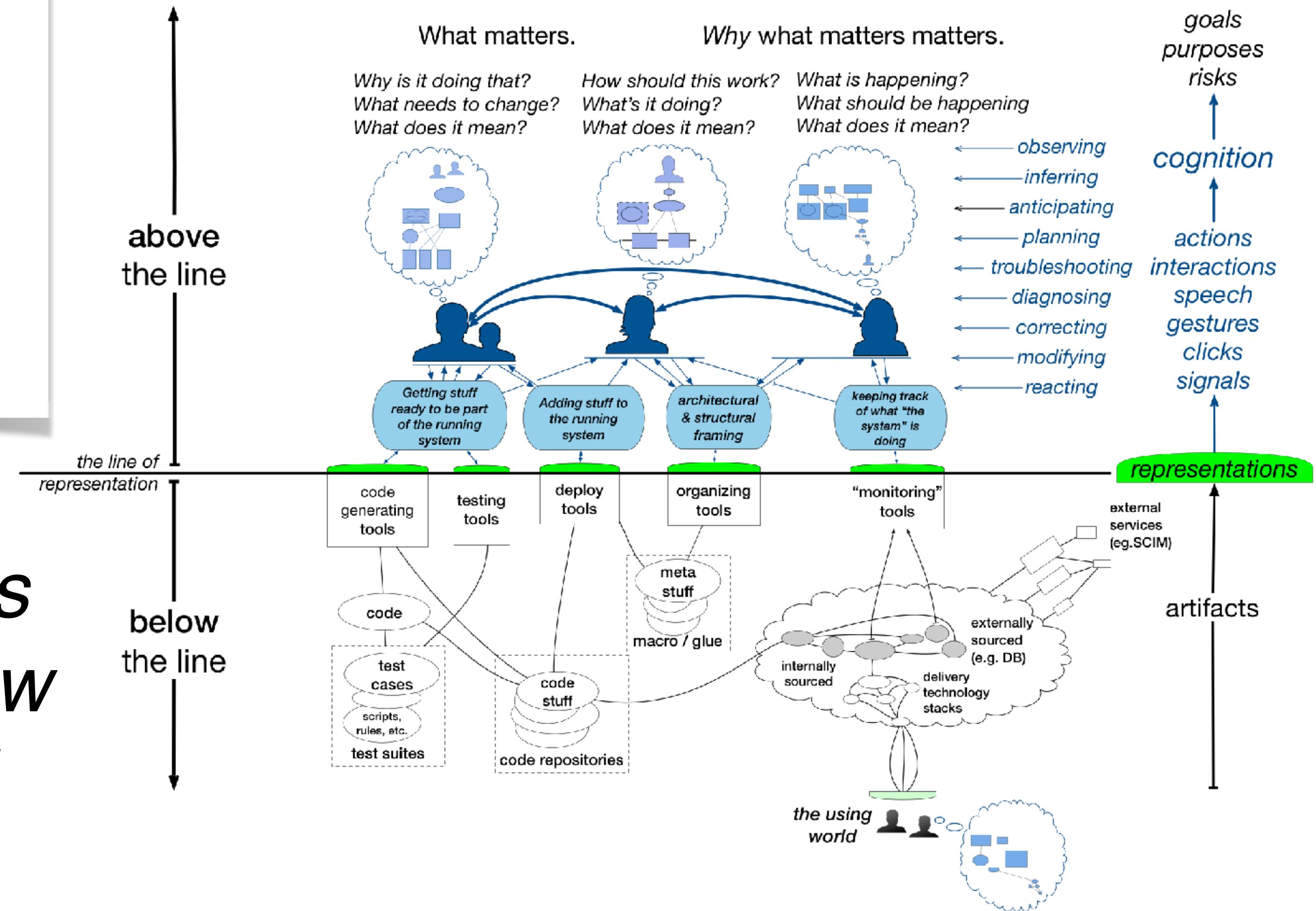
and W. ROSS ASHBY

Biological Computers Laboratory, University of Illinois,
Urbana, Illinois 61801, U.S.A.‡

[Received 3 June 1970]

hypothesis based on Conant & Ashby (1970):

All the distributed systems behaviors that occur below the line will also be found above the line.



Research results

tl;dr: *a lot is happening on the other side of ‘deploy’.*

- Clarifying **system** changes everything
- Experience with incidents doesn’t fit the paradigm
 - ▶ Everyone is awash in this-might-be-an-incidents, incidents, is-this-big-one?s and OMGs.
 - ▶ Managing incidents is now a *thing*.
 - ▶ Learning from incidents is essential but quite hard.
 - ▶ Ever more dependencies that aren’t yours to deploy.
 - ▶ There is wisdom in incidents.
- Complexity *is* change!

join me **Tuesday**
at **15:15** in **Breakout D**

1974

