

El arte de desarrollar:

¿En qué pensamos cuando pensamos en software?



BilboStack;

Patrocinadores principales



Patrocinador Platino



Patrocinadores Oro



Patrocinadores Bronce





Adrià Fontcuberta
Software Engineer at **Stripe**

Host: Username: Password: Port: **Quickconnect** ▼

Status: File transfer successful
 Status: Starting upload of /Users/codesquid/svn/FileZilla3/locales/fr.po
 Command: PASV
 Response: 227 Entering Passive Mode (754,110,11,59,79,195)
 Command: STOR fr.po
 Response: 150 Connection accepted

Local site: /Users/codesquid/svn/FileZilla3/

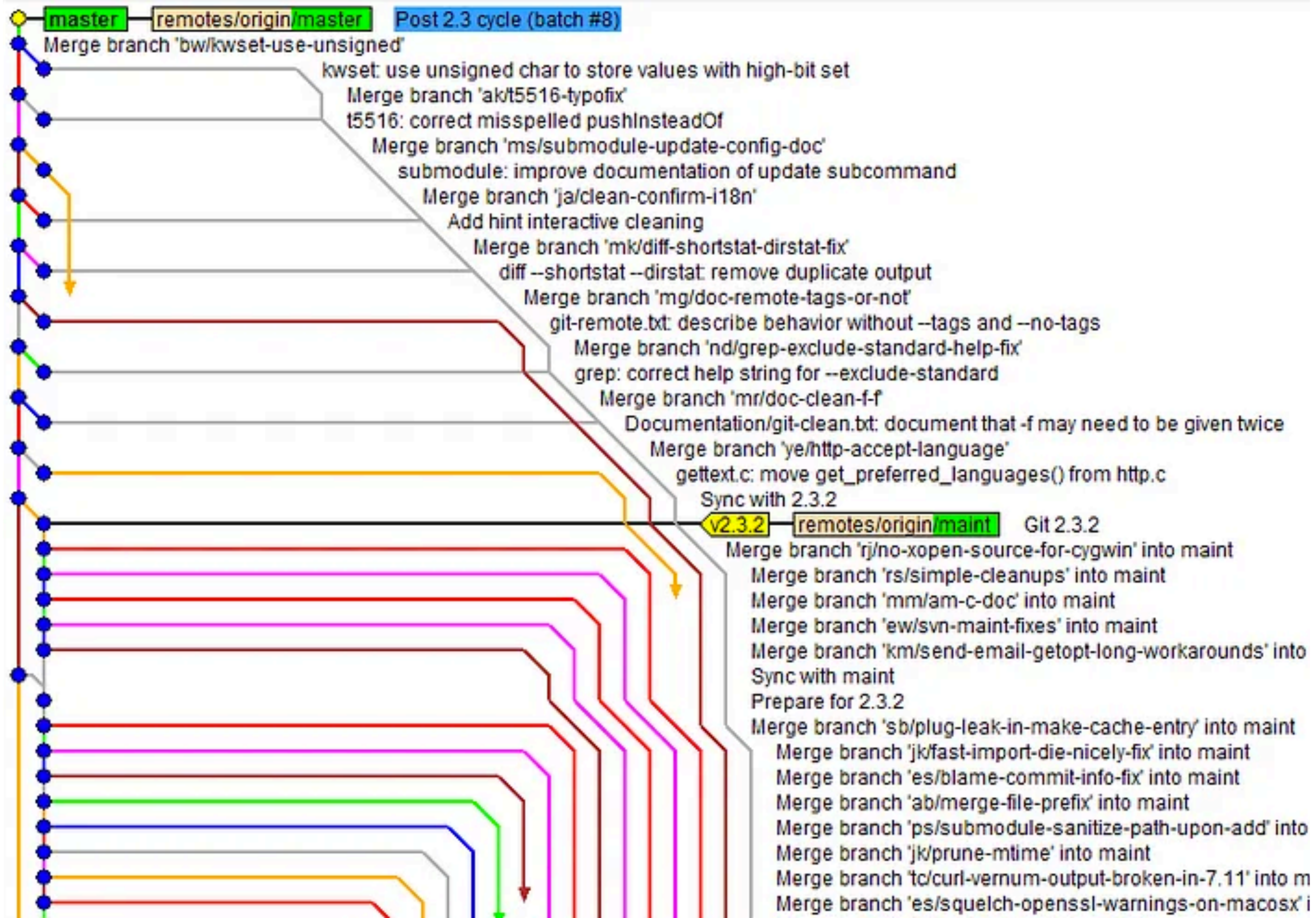
Remote site: /FileZilla3

- ▼ svn
 - ▼ FileZilla3
 - ▶ .svn
 - ▶ autom4te.cache
 - ▼ compile
 - ▼ FileZilla.app
 - ▼ Contents

- ▼ ? /
 - ▶ FileZilla3

Filename	Filesize	Filetype	Filename	Filesize	Filetype	Last modified
..			docs		Folder	06/15/2008
.svn		Folder	locales		Folder	06/15/2008
autom4te.cache		Folder	src		Folder	06/15/2008
compile		Folder	tests		Folder	06/15/2008
config		Folder	.DS_	12,295	File	06/15/2008
data		Folder	AUT	2,272	File	06/15/2008

- Download
- + Add files to queue
- View / Edit



- New Item
- People
- Build History
- Project Relationship
- Check File Fingerprint
- Manage Jenkins
- Credentials

[add description](#)

All Magic View Not so magic view +

S	W	Name ↓	Last Success	Last Failure	Last Duration
		another build	N/A	5 days 21 hr - #2	34 sec
		freestyle	5 days 1 hr - #8	N/A	40 ms

Icon: [S](#) [M](#) [L](#)

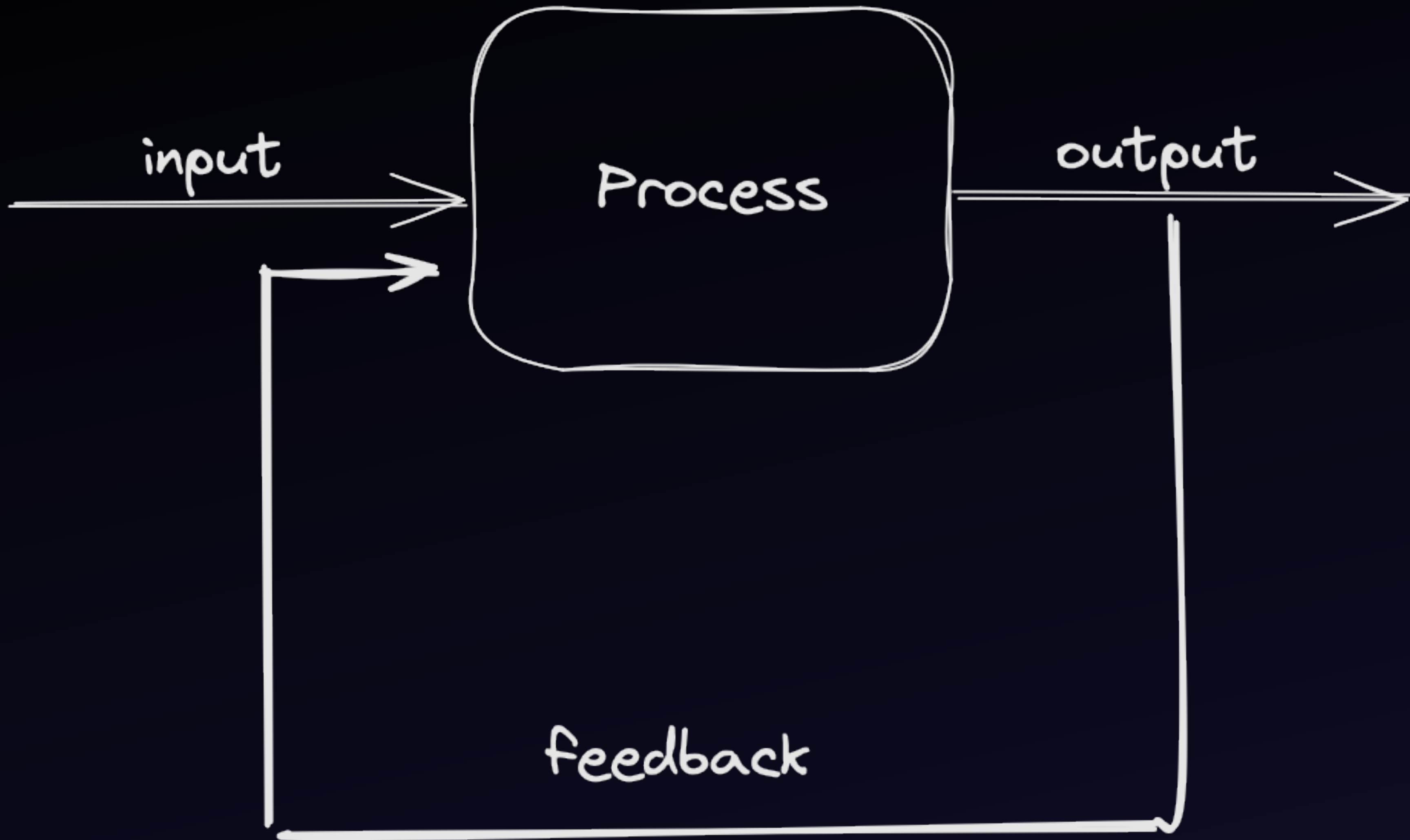
[Legend](#) [RSS for all](#) [RSS for failures](#) [RSS for just latest builds](#)

Build Queue

No builds in the queue.

Build Executor Status

- 1 Idle
- 2 Idle



What if everything in software development was about **creating** feedback loops?

Unit test: “Does this small unit of code do what I think it does?”

Code review: “Are we writing readable, maintainable code?”

Daily standup: “Are we set and able to work on the right thing for the next few hours?”

Retrospective: “Did we work well, at a sustainable pace, for the last days?”

Monitoring/Analytics: “Did the new feature get the expected usage and performance?”

User research: “Do we understand the problem we're trying to solve?”

Am I doing the
right thing?

Unit tests

User research

Monitoring/Analytics

Am I doing the
thing **right**?

Retrospective

Code review

¿Daily standup?

It's a **loop**.

You need to **act** on the
result of your actions.

Doing something and not reflecting back is not a feedback loop.

It is **running in circles.**

Feedback loops are
not free.

Why do we use
feedback loops, then?

It's not like we have a choice.

It's the **nature of software**,
the nature of digital products.

The nature of
complex spaces.

Clear

Playing Tetris

Complicated

Fixing a car

Complex

Winning a race

Chaotic

Playing with kids

Have you ever tried
estimating a task?

00:03:22 (2.0)



Advance to Tool Age: Requires 2 Stone Age buildings (Food: 500)

Gofermajster: 210/210

Hatti
Town Center

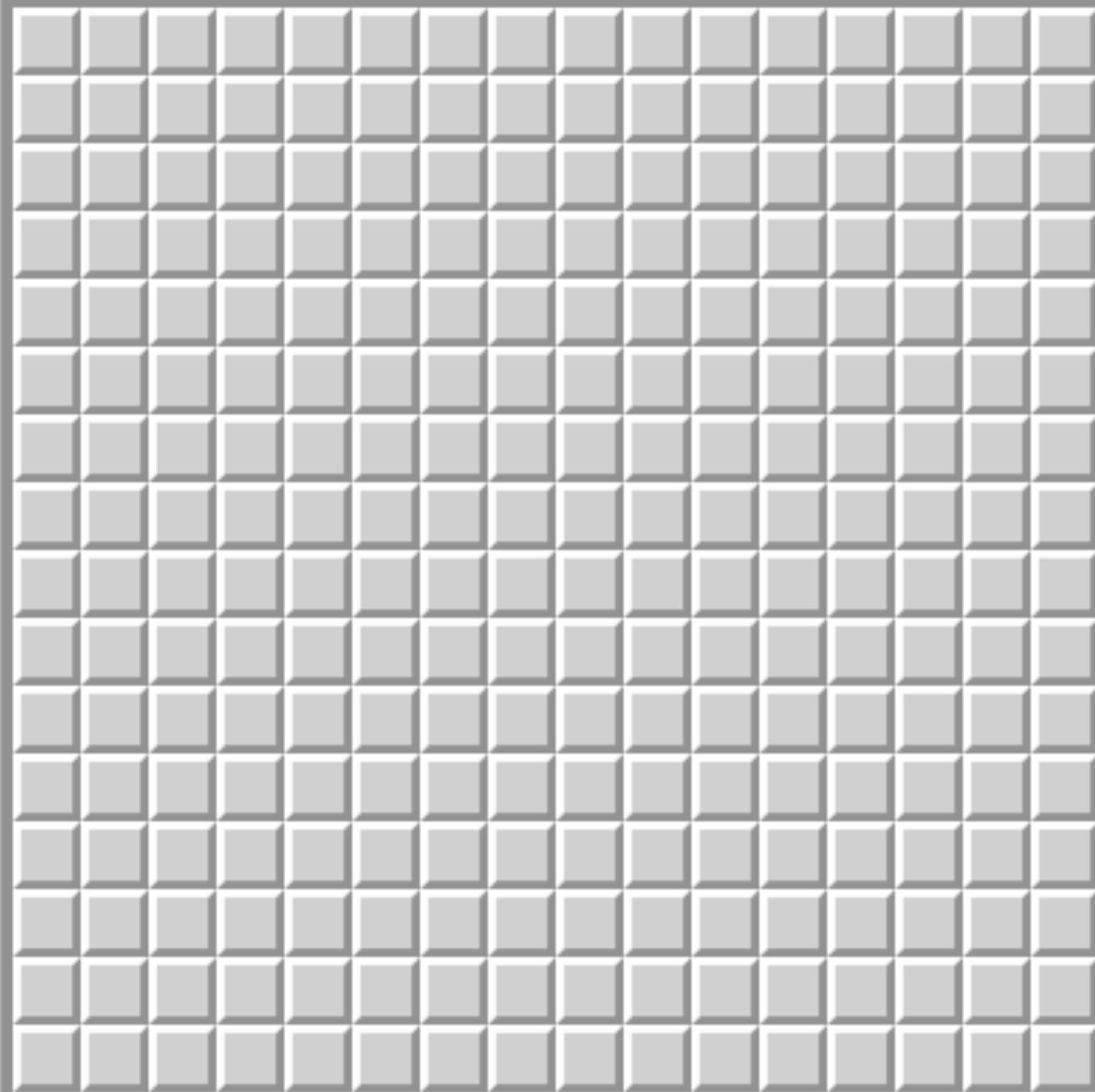


600/600




S

?



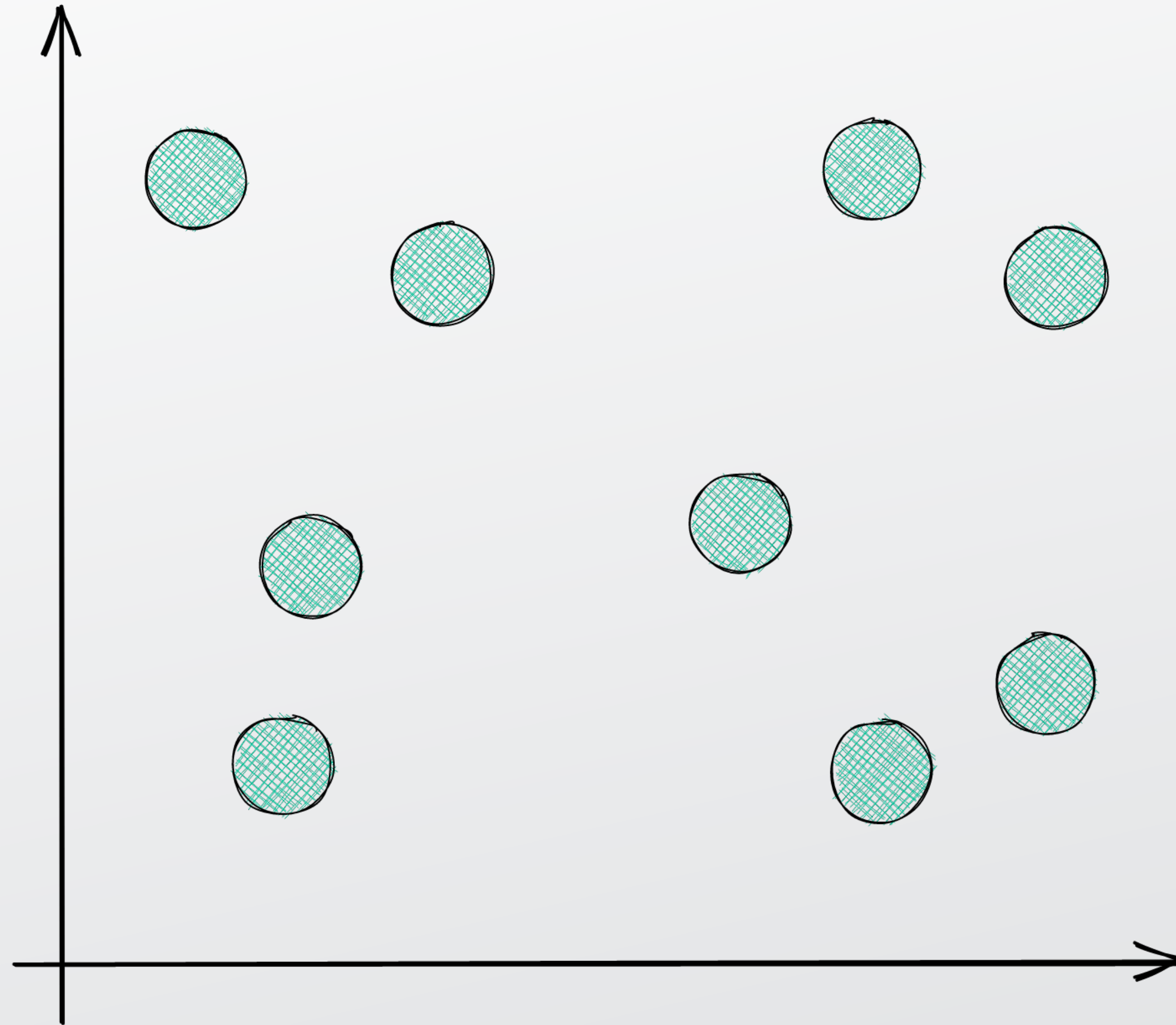
Reihen	Gruppe I. — R'O	Gruppe II. — RO	Gruppe III. — R'O ³	Gruppe IV. RH ⁴ RO ³	Gruppe V. RH ³ R'O ³	Gruppe VI. RH ³ RO ³	Gruppe VII. RH R'O ²	Gruppe VIII. — RO ⁴
1	II=1							
2	Li=7	Be=9,4	B=11	C=12	N=14	O=16	F=19	
3	Na=23	Mg=24	Al=27,3	Si=28	P=31	S=32	Cl=35,5	
4	K=39	Ca=40	—=44	Ti=48	V=51	Cr=52	Mn=55	Fe=56, Co=59, Ni=59, Cu=63.
5	(Cu=63)	Zn=65	—=68	—=72	As=75	Se=78	Br=80	
6	Rb=86	Sr=87	?Yt=88	Zr=90	Nb=94	Mo=96	—=100	Ru=104, Rh=104, Pd=106, Ag=108.
7	(Ag=108)	Cd=112	In=113	Sn=118	Sb=122	Te=125	J=127	
8	Cs=133	Ba=137	?Di=138	?Ce=140	—	—	—	— — — —
9	(—)	—	—	—	—	—	—	
10	—	—	?Er=178	?La=180	Ta=182	W=184	—	Os=195, Ir=197, Pt=198, Au=199.
11	(Au=199)	Hg=200	Tl=204	Pb=207	Bi=208	—	—	
12	—	—	—	Th=231	—	U=240	—	— — — —

This is not about moving fast.

This is about learning where
not to go, the sooner the
better.

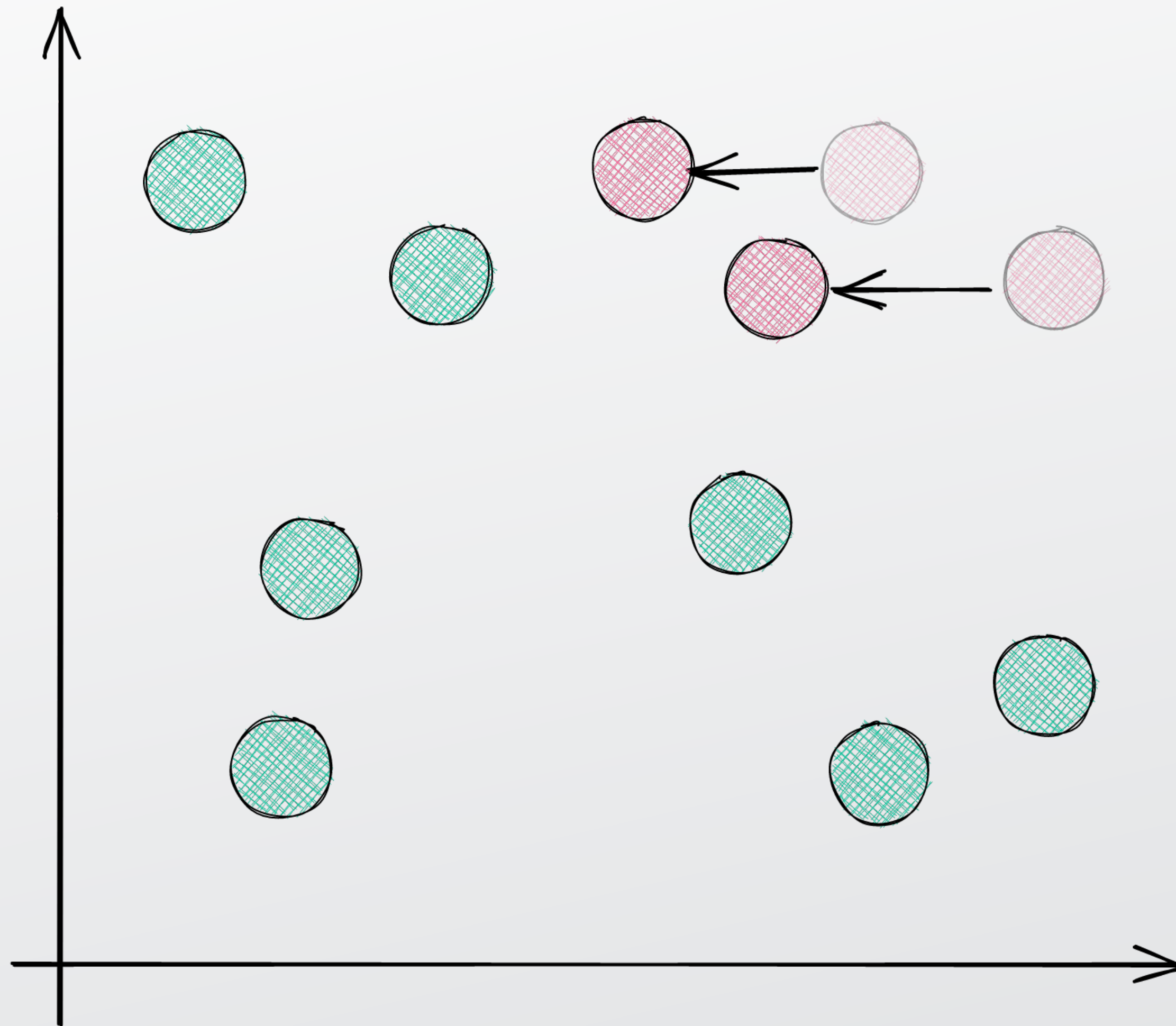
Creating software is a
game of **assumptions**.

How critical
is the
assumption?



How uncertain are we about it?

How critical
is the
assumption?



How uncertain are we about it?

What is the **level of confidence** we are comfortable with?

What is the **fastest** way to get there?

Agility vs. Predictability

Software is a set of
complex problems (mostly).

The *best* way to move forward is
by **experimenting, iterating,**
and **deferring** decisions.

We do so by leveraging
feedback loops.

1. **Analyze** the current situation.
2. **Take a step**, the smallest, safest possible that gives helpful info.
3. **Reflect** on what you just learned.
4. **Go back** to step 1.

Thank you, folks!

From @afontq with  to Bilbostackers