El arte de desarrollar: ¿En qué pensamos cuando pensamos en software?





Patrocinadores principales



Patrocinador Platino



Patrocinadores Oro

Libere KAIROSDS

Patrocinadores Bronce





Adrià Fontcuberta Software Engineer at Stripe

) 🔿 🌀						
	💠 🥵					
Host: Username:	Password:	Por	t: Quickc	onnect		
Status: File transfer successful Status: Starting upload of /Users/code Command: PASV Response: 227 Entering Passive Mode (75 Command: STOR fr.po Response: 150 Connection accepted		fr.po				
Local site: /Users/codesquid/svn/FileZil	la3/	Remote site:	/FileZilla3			
🔻 📔 svn	6	v ? /				
🔻 🧀 FileZilla3		📄 🗼 🧰 File	zilla2			
🕨 🧰 .svn			ezilla5			
autom4te.cache						
🔻 🧰 compile						
🔻 🧰 FileZilla.app						
Contents	*					
	•			•		
Filename	Filesize Filety	Filename		Filesize Filetype	e Last n	
🗀		a docs		Folder	06/15	
🧰 .svn	Fold	🧀 locales		Folder	06/15	
autom4te.cache	Fold	🚞 srq 🥢	Download	Folder	06/15	
🚞 compile	Fold	Contests		Folder	06/15	
🧰 config	Fold	DS	Add files to queue	e 12,295 File	06/15	
🧰 data	Fold		ew / Edit	2,272 File	06/15	



⊖– <mark> maste</mark> r	remotes/origin/master Post 2.3 cycle (batch #8)
Merge b	branch 'bw/kwset-use-unsigned'
•	kwset: use unsigned char to store values wit
	Merge branch 'ak/t5516-typofix'
`	t5516: correct misspelled pushInsteadOf
	Merge branch 'ms/submodule-update-
0	submodule: improve documentation
	Merge branch 'ja/clean-confirm-i1
>	Add hint interactive cleaning
	Merge branch 'mk/diff-shorts
<u>▶</u>	diffshortstatdirstat re
	Merge branch 'mg/doc
>	git-remote.txt: descr
	Merge branch 'no
•	grep: correct help
	Merge branch
>	Documenta
	Merge br
*	gettex
	Syr
<u> </u>	
•	
•	
•	
•	
•	
•	
-	
•	
•	
+	
•	
•	
•	
•	

th high-bit set

config-doc' of update subcommand 8n'

stat-dirstat-fix'

move duplicate output

-remote-tags-or-not

ibe behavior without -- tags and -- no-tags

l/grep-exclude-standard-help-fix'

string for --exclude-standard

'mr/doc-clean-f-f

tion/git-clean.txt: document that -f may need to be given twice

anch 'ye/http-accept-language'

t.c: move get_preferred_languages() from http.c

nc with 2.3.2

.3.2 remotes/origin/maint Git 2.3.2

Merge branch 'rj/no-xopen-source-for-cygwin' into maint

Merge branch 'rs/simple-cleanups' into maint

Merge branch 'mm/am-c-doc' into maint

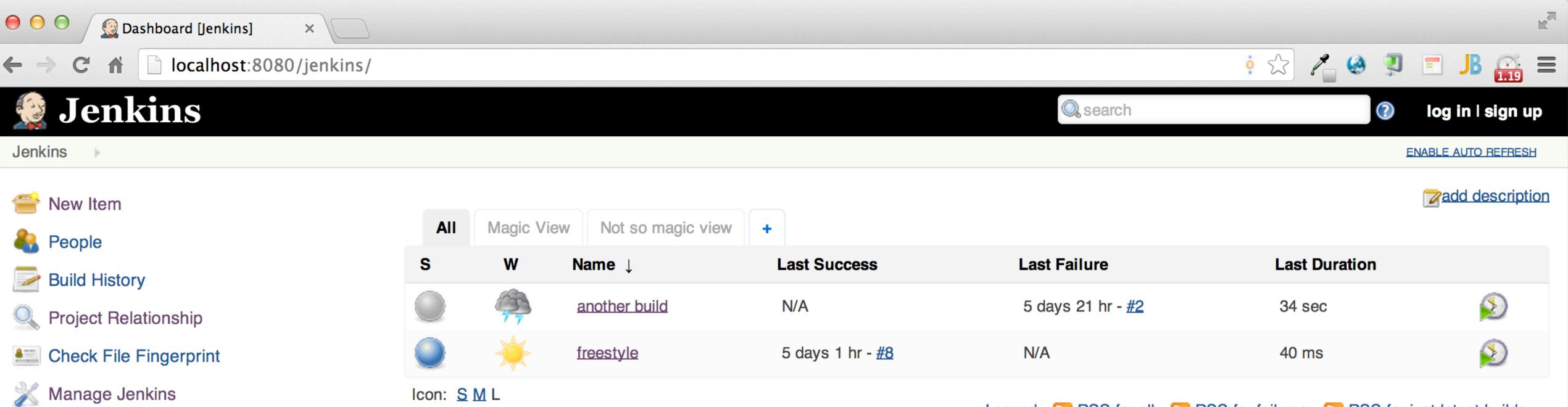
Merge branch 'ew/svn-maint-fixes' into maint

Merge branch 'km/send-email-getopt-long-workarounds' into Sync with maint

Prepare for 2.3.2

Merge branch 'sb/plug-leak-in-make-cache-entry' into maint Merge branch 'jk/fast-import-die-nicely-fix' into maint Merge branch 'es/blame-commit-info-fix' into maint Merge branch 'ab/merge-file-prefix' into maint Merge branch 'ps/submodule-sanitize-path-upon-add' into Merge branch 'jk/prune-mtime' into maint Merge branch 'tc/curl-vernum-output-broken-in-7.11' into m

Merge branch 'es/squelch-openssl-warnings-on-macosx' i

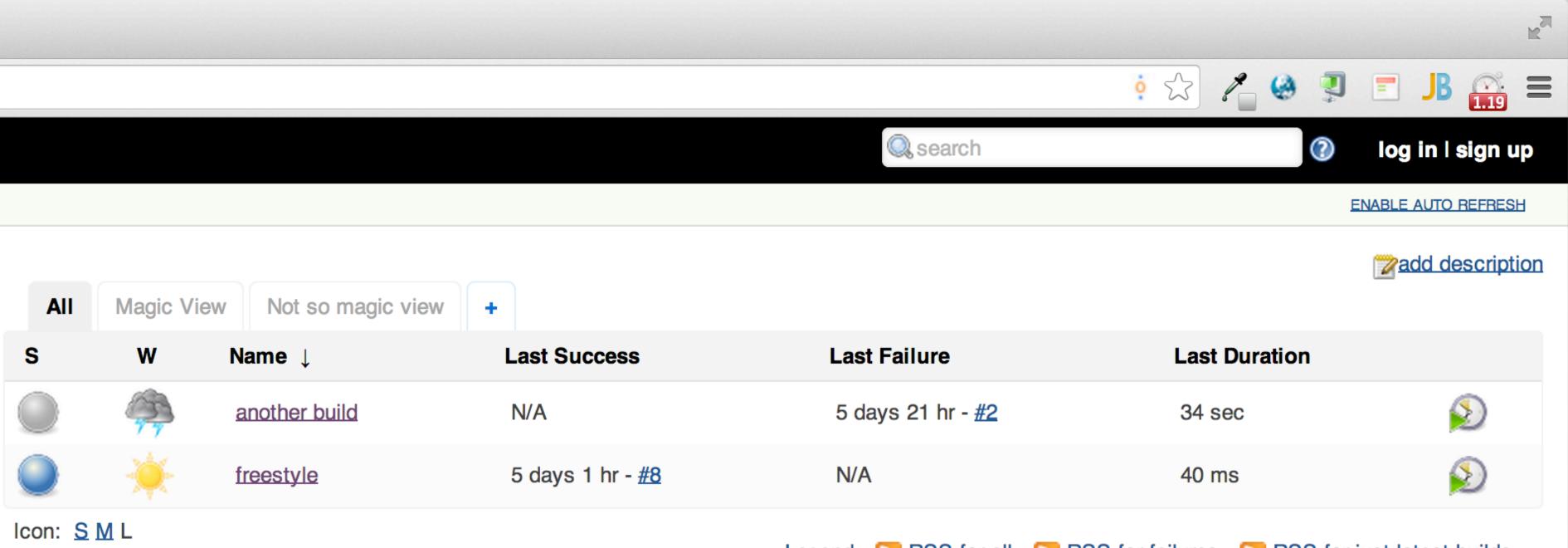






Credentials



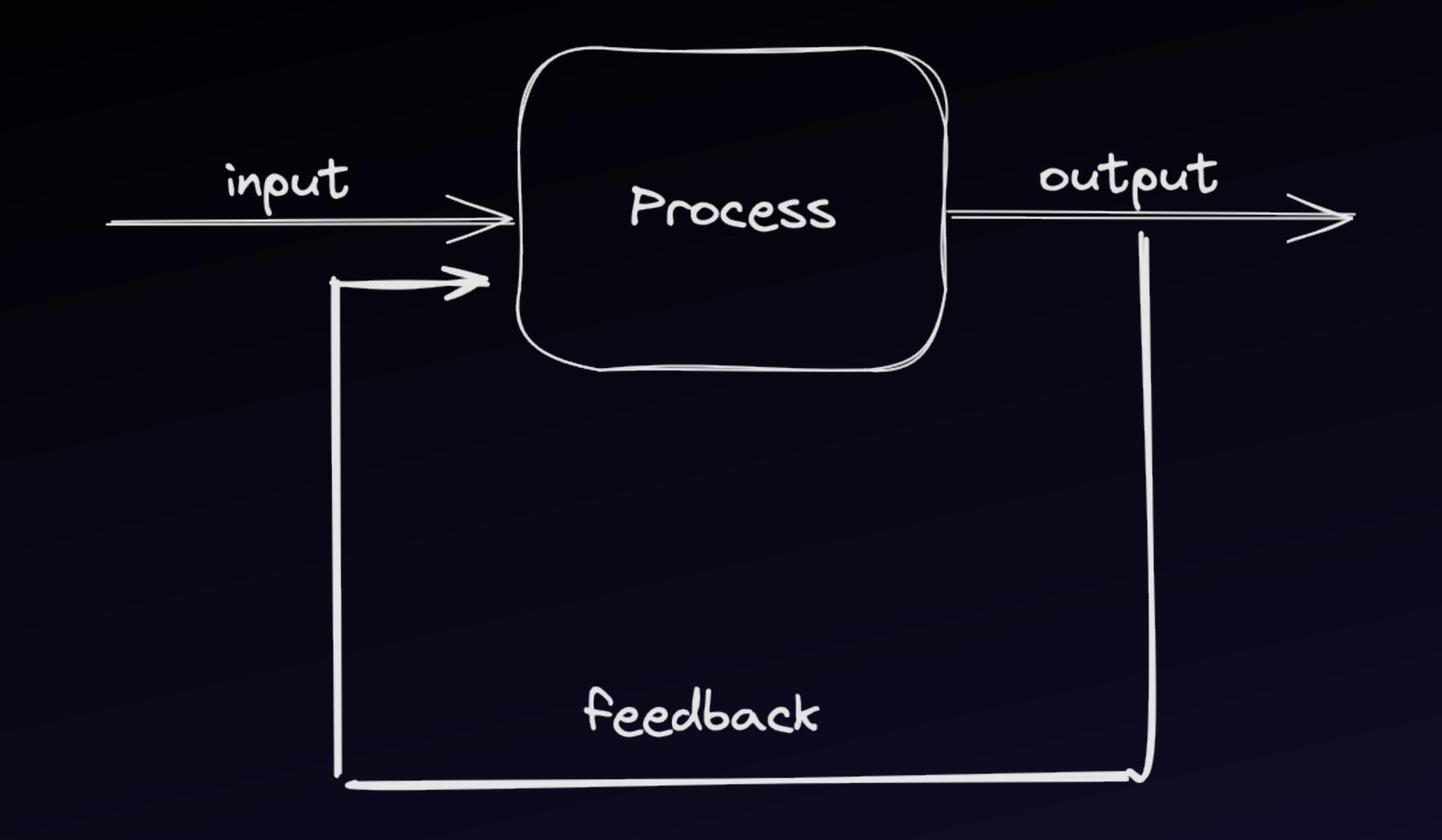


Build	Queue

No builds in the queue.

Build Executor Status	-
1 Idle	
2 Idle	

Legend S RSS for all S RSS for failures S RSS for just latest builds



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?"

trying to solve?"

- **User research**: "Do we understand the problem we're

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

Complex Winning a race

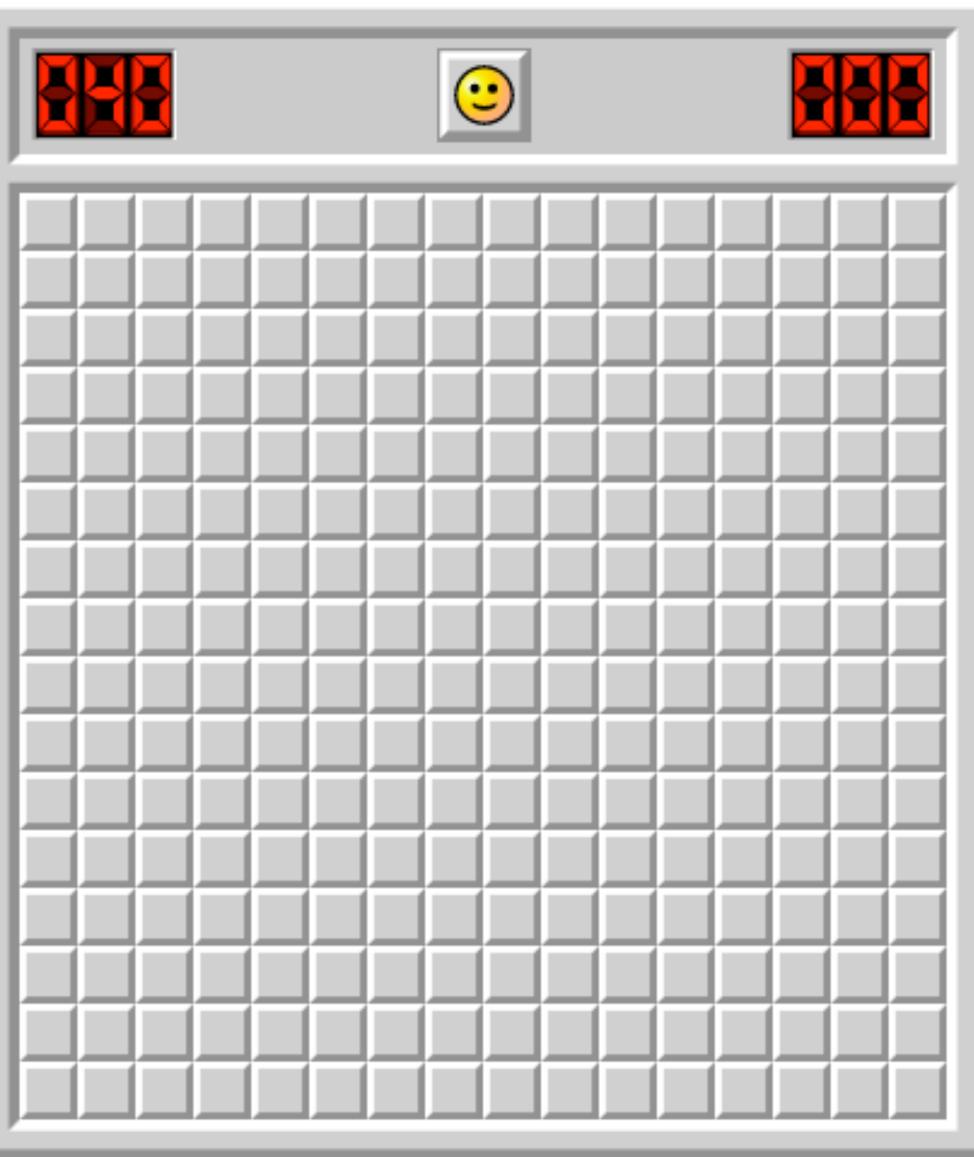
Complicated Fixing a car

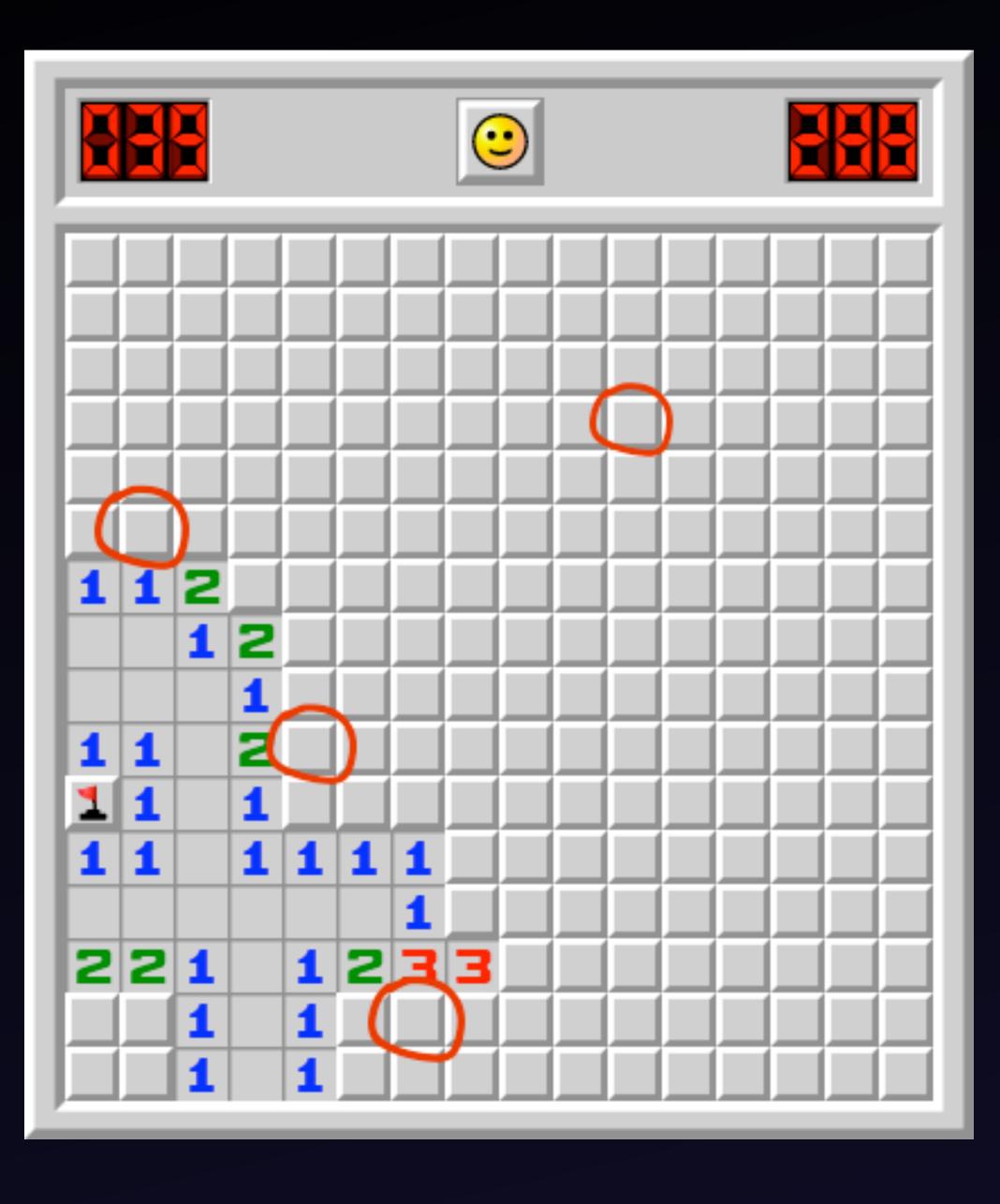
Chaotic Playing with kids

Have you ever tried estimating a task?

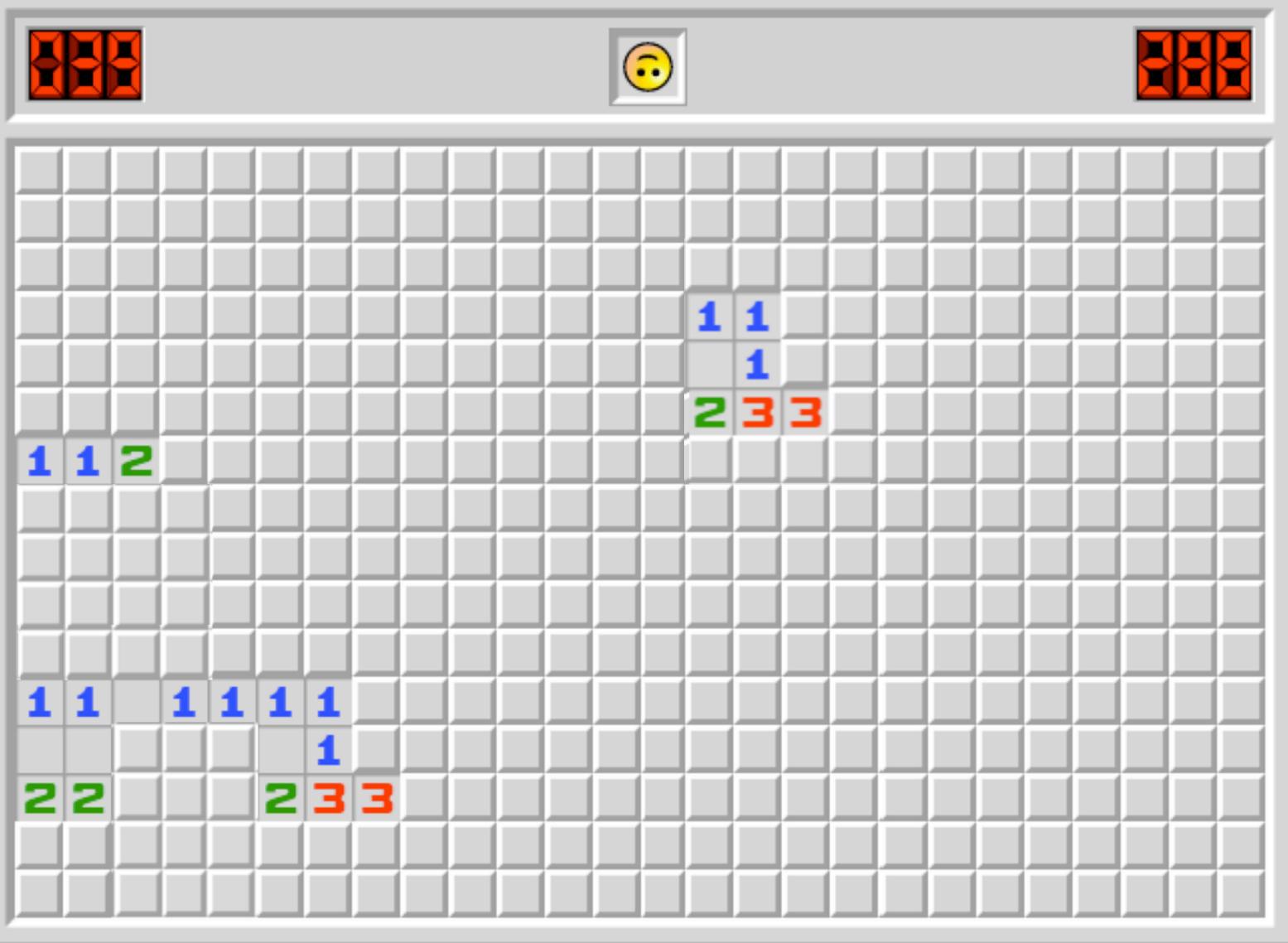












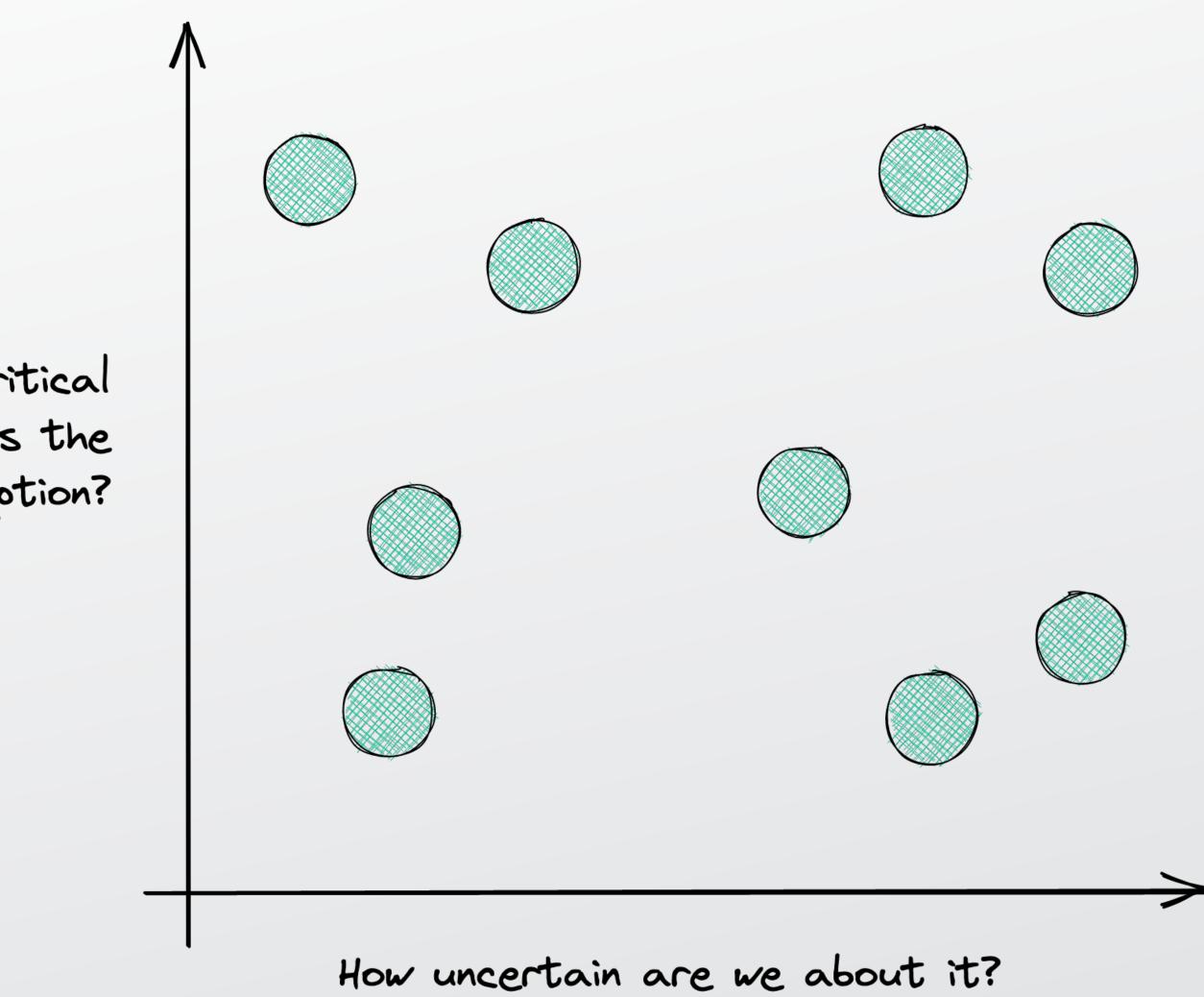
Reihen	Gruppo I. R*0	Gruppo II. RO	Gruppe III, R'0'	Gruppe IV. RH4 RQ3	Grappe V. RH ³ R ² 0 ³	Grappo VI. RHª RO'	Gruppe VII. RH R*0'	Gruppo VIII. RO4
1	II=1					1		
2	Li = 7	Bo=9,4	B==11	C==12	N = 14	Q== 16	F=19	
\$	Na=23	Mg = 24	A1=27,8	Si=28	P=31	S=32	Cl== 35,5	
4	K=39	Ca== 40	-==	Ti== 48	V==51	Cr == 52	Mn=55	Fo=56, Co=59, Ni=59, Cu=63.
5	(Cu=63)	Zn==65	-=68	_=72	As=75	So=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	?Co=140	-	-	-	
9	()		- 1	_	-		-	
10	-	-	?Er=178	?La=180	Ta=182	W=184	-	Os=195, Ir=197, Pt=198, Au=199.
11	(Au=199)	flg=200	T1== 204	Pb=207	Bi== 208	- 1	-	
12	-	-	-	Th=231	-	U==240	-	



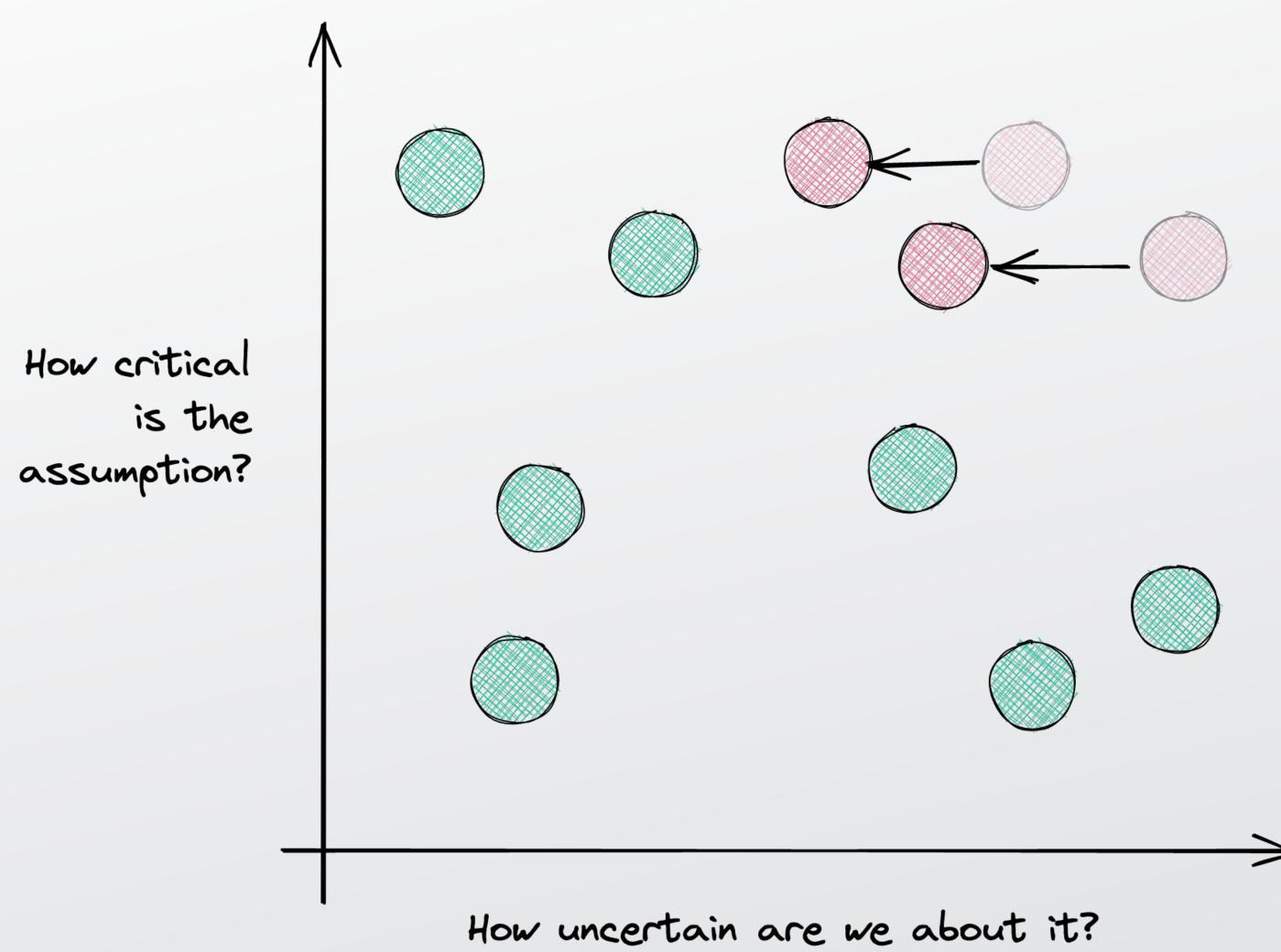
This is not about moving fast.

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

Creating software is a game of assumptions.



How critical is the assumption?



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 @afontg with 💙 to Bilbostackers