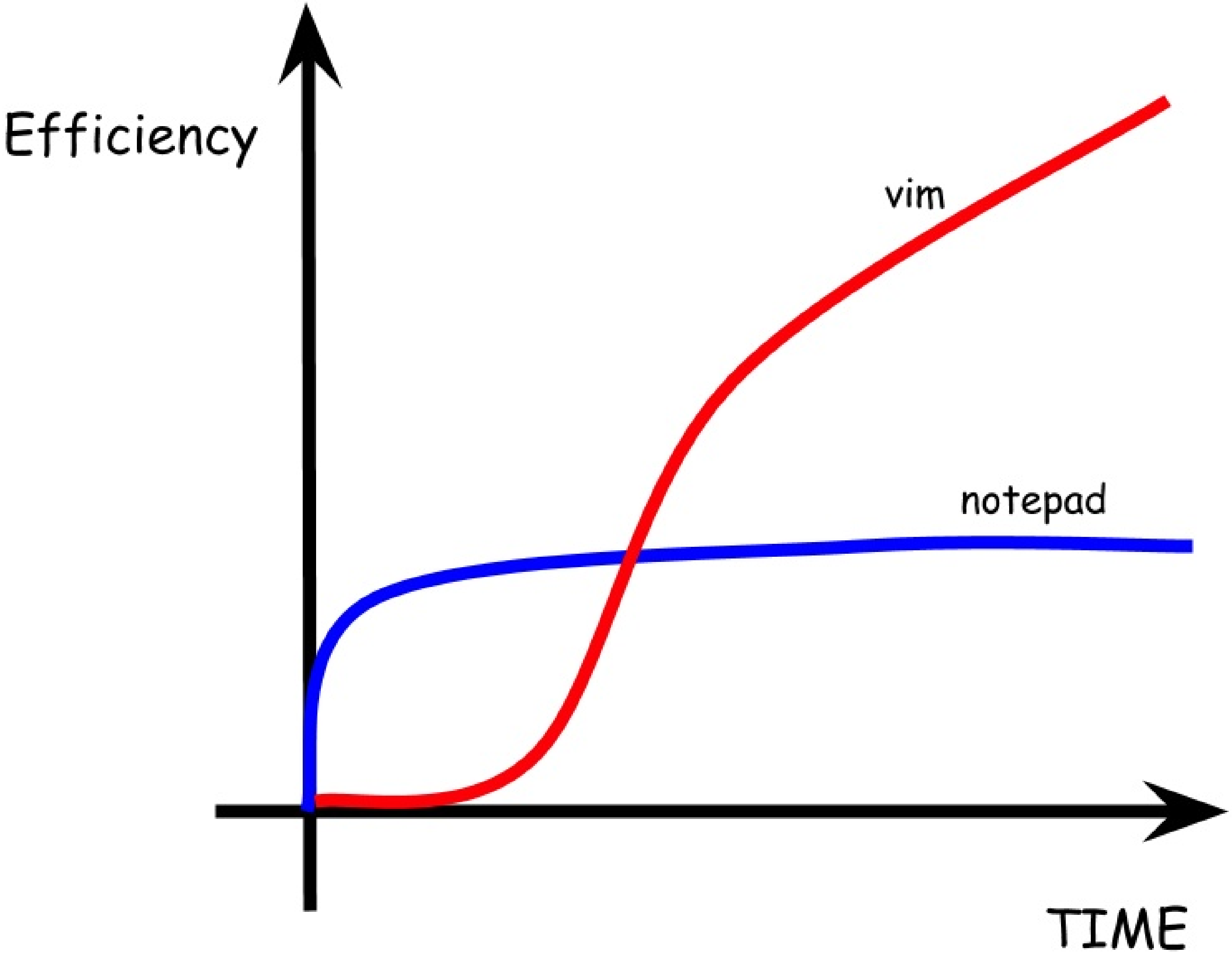
- It’s not that difficult

BaNaNa Talk 19-04-2018
Reinaldo Astudillo
R.A.Astudillo@tudelft.nl

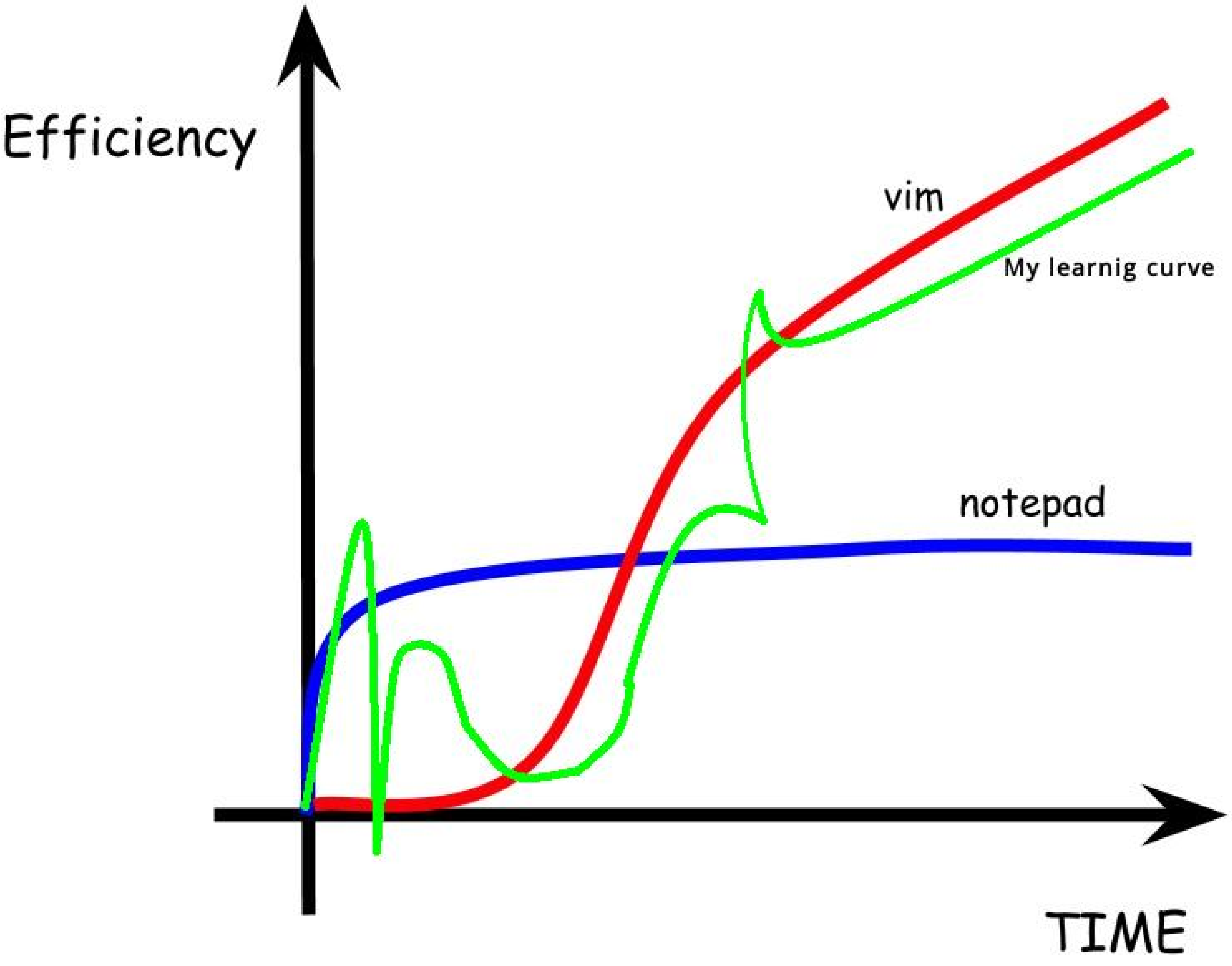
Why Vim?

- Ubiquitous.
- Configurable.
- Efficient^2.
- Multipurpose.

Learning curve and efficiency in Vim



Learning curve and efficiency in Vim



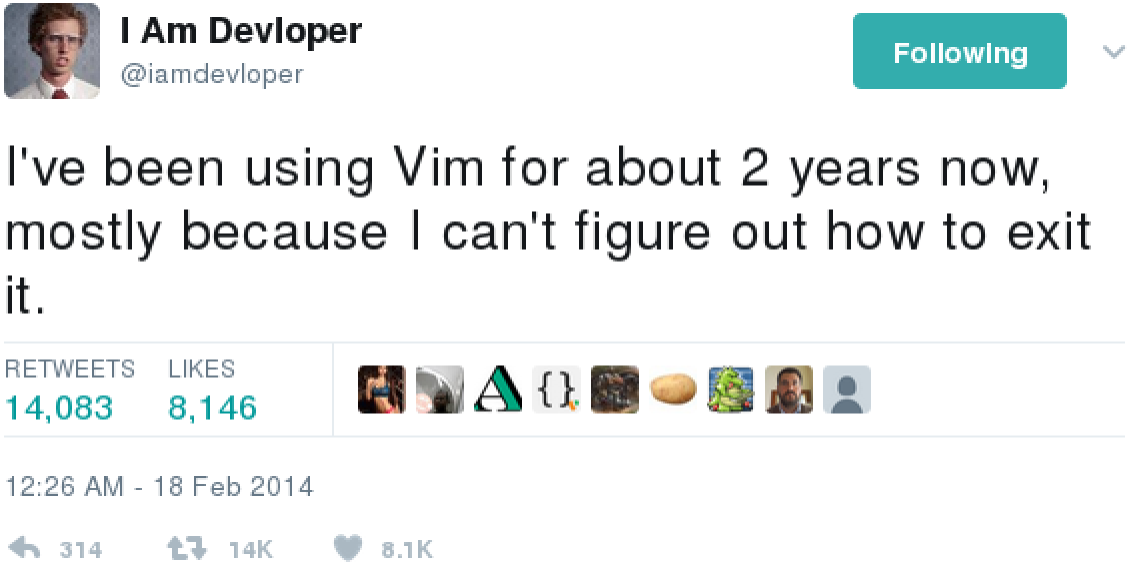
Vim

Schedule

- Vim modes
- Vim language
- Vim configuration
- Let’s try some plugins
- Takeaway
- For more information

Vim modes

- Normal mode (movement around, editing)
- Insert mode (inserting text)
- Command mode (save, open, close a file, exiting)



I Am Devloper
@iamdevloper

I've been using Vim for about 2 years now, mostly because I can't figure out how to exit it.

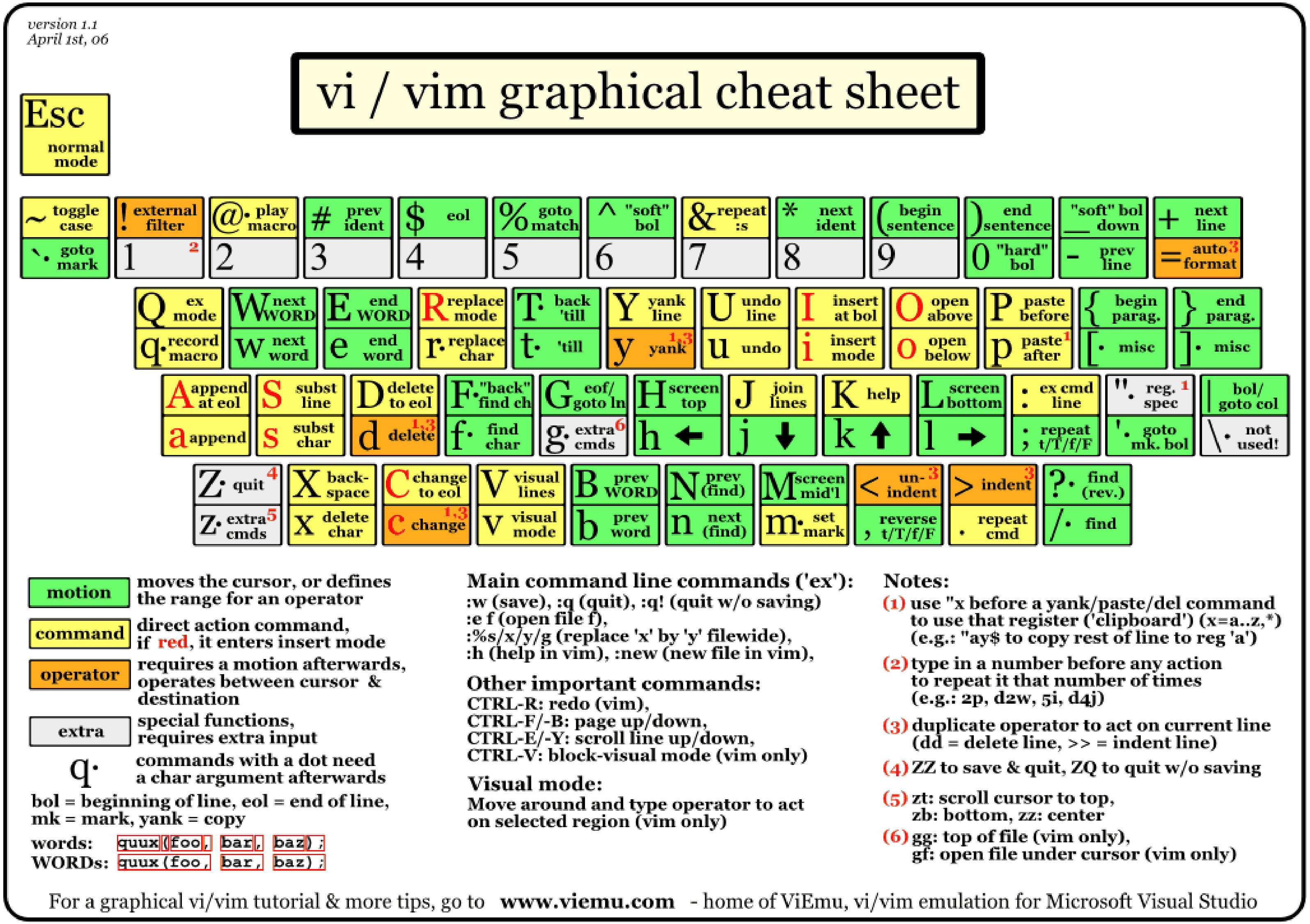
RETWEETS 14,083 LIKES 8,146

12:26 AM - 18 Feb 2014

Vim language

- **[number] + verb + text object or movement.**
- *Movement*: left (h), down (j), up (k), right (l), word (w), end of line ($), find, search.
- *Text object*: a word, a sentence, a paragraph, inside quoted text, inside html tags.
- *Verb:* (v)isual, (c)hange, (d)elete, (y)ank [copy], (g)o (U)ppercase (gU).
- *Changes* should be undoable a repeatable.

More commands



version 1.1
April 1st, 06

vi / vim graphical cheat sheet

Esc normal mode

~ toggle case
` · goto mark
! external filter
1 [2]
@· play macro
2
# prev ident
3
$ eol
4
% goto match
5
^ "soft" bol
6
& repeat :s
7
* next ident
8
(begin sentence
9
) end sentence
0 "hard" bol
_ "soft" bol down
- prev line
+ next line
= auto format [3]

Q ex mode
q· record macro
W next WORD
w next word
E end WORD
e end word
R replace mode
r· replace char
T· back 'till
t· 'till
Y yank line
y yank [1,3]
U undo line
u undo
I insert at bol
i insert mode
O open above
o open below
P paste before
p paste after [1]
{ begin parag.
[· misc
} end parag.
]· misc

A append at eol
a append
S subst line
s subst char
D delete to eol
d delete [1,3]
F· "back" find ch
f· find char
G eof/ goto ln
g· extra cmds [6]
H screen top
h ←
J join lines
j ↓
K help
k ↑
L screen bottom
l →
: ex cmd line
; repeat t/T/f/F
"· reg. spec [1]
'· goto mk. bol
| bol/ goto col
\· not used!

Z· quit [4]
z· extra cmds [5]
X back-space
x delete char
C change to eol
c change [1,3]
V visual lines
v visual mode
B prev WORD
b prev word
N prev (find)
n next (find)
M screen mid'l
m· set mark
< un-indent [3]
, reverse t/T/f/F
> indent [3]
. repeat cmd
?· find (rev.)
/· find

motion	moves the cursor, or defines the range for an operator
command	direct action command, if red, it enters insert mode
operator	requires a motion afterwards, operates between cursor & destination
extra	special functions, requires extra input
q·	commands with a dot need a char argument afterwards

bol = beginning of line, eol = end of line,
mk = mark, yank = copy

words: quux(foo, bar, baz);
WORDs: quux(foo, bar, baz);

Main command line commands ('ex'):
:w (save), :q (quit), :q! (quit w/o saving)
:e f (open file f),
:%s/x/y/g (replace 'x' by 'y' filewide),
:h (help in vim), :new (new file in vim),

Other important commands:
CTRL-R: redo (vim),
CTRL-F/-B: page up/down,
CTRL-E/-Y: scroll line up/down,
CTRL-V: block-visual mode (vim only)

Visual mode:
Move around and type operator to act on selected region (vim only)

Notes:

(1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a..z,*) (e.g.: "ay$ to copy rest of line to reg 'a')

(2) type in a number before any action to repeat it that number of times (e.g.: 2p, d2w, 5i, d4j)

(3) duplicate operator to act on current line (dd = delete line, >> = indent line)

(4) ZZ to save & quit, ZQ to quit w/o saving

(5) zt: scroll cursor to top, zb: bottom, zz: center

(6) gg: top of file (vim only), gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to **www.viemu.com** - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

Vim configuration

- The ~/.vimrc file.
- Let’s check my vimrc

Let’s try some plugins

- commentary.vim
- scrooloose/nerdtree
- surround.vim
- garbas/vim-snipmate
- fugitive.vim

Takeaway

- Vim is a very flexible and powerful text-editor.
- There are tons of vim plugins (fugitive, surround, the nerd commenter). Try them and share your experience.
- You don't have to memorize a bunch of commands just the learn the vim language.
- Create your personalized *vimrc* file (share it and steal ideas from others *vimrc* files).
- De-active the arrow keys :-).

For more information

- Vimtutor
- Vim adventures
- Practical Vim, Edit Text at the Speed of Thought, 2nd ed. by Drew Neil, Pragmatic Bookshelf, 2012.
- Learning the Vi and Vim Editors: Text Processing at Maximum Speed and Power 7th ed. by A. Robbins, E. Hannah, L. Lamb, O'Reilly Media, 2008.
- Talk: Mastering the Vim Language by Chris Toomey, on thoughtbot's youtube channel, 2015.