Automation for fun and profits

- Why do we have so many languages?
  - Memory is so precious - Embedded systems (C, C++,)
  - I can afford more RAM - Webservices, Data pipelines (Java, Scala)
  - I want to get stuff done - scripting (Perl, Python, Ruby)
  - Fun - (https://esolangs.org/wiki/Hello_world_program_in_esoteric_languages)
- Why do we have so many problems?
  - Tired of C and an impatient programmer to parse logs - Perl
  - A bored programmer on a Christmas weekend - Python
  - Tired with both of the above - Ruby
  - Ruby is so slow but we love the syntax - Crystal, Elixir, etc

- Quotes
    - The greatest single programming language ever designed. - **Alan Kay**
    - The most powerful programming language is Lisp. If you don't know Lisp you don't appreciate what a powerful language is. Once you learn Lisp you will see what is missing in most other languages. - **Richard Stallman**
    - Lisp has jokingly been called "the most intelligent way to misuse a computer". I think that description is a great compliment because it transmits the full flavor of liberation - **Edsger W. Dijkstra**
    - Lisp is worth learning for the profound enlightenment experience you will have when you finally get it; that experience will make you a better programmer for the rest of your days, even if you never actually use Lisp itself a lot.- **Eric Raymond**

- Clojure
    - Lisp on JVM
    - Immutable by default
    - Laziness
    - Dynamic nature and interactivity

- Redbus bot
    - Bus tickets : A pain to book in the long weekends
    - Ctrl + F5 every 5 minutes
    - Spots a JSON request
    - Lets scrap!
    - Python + requests + telegram library

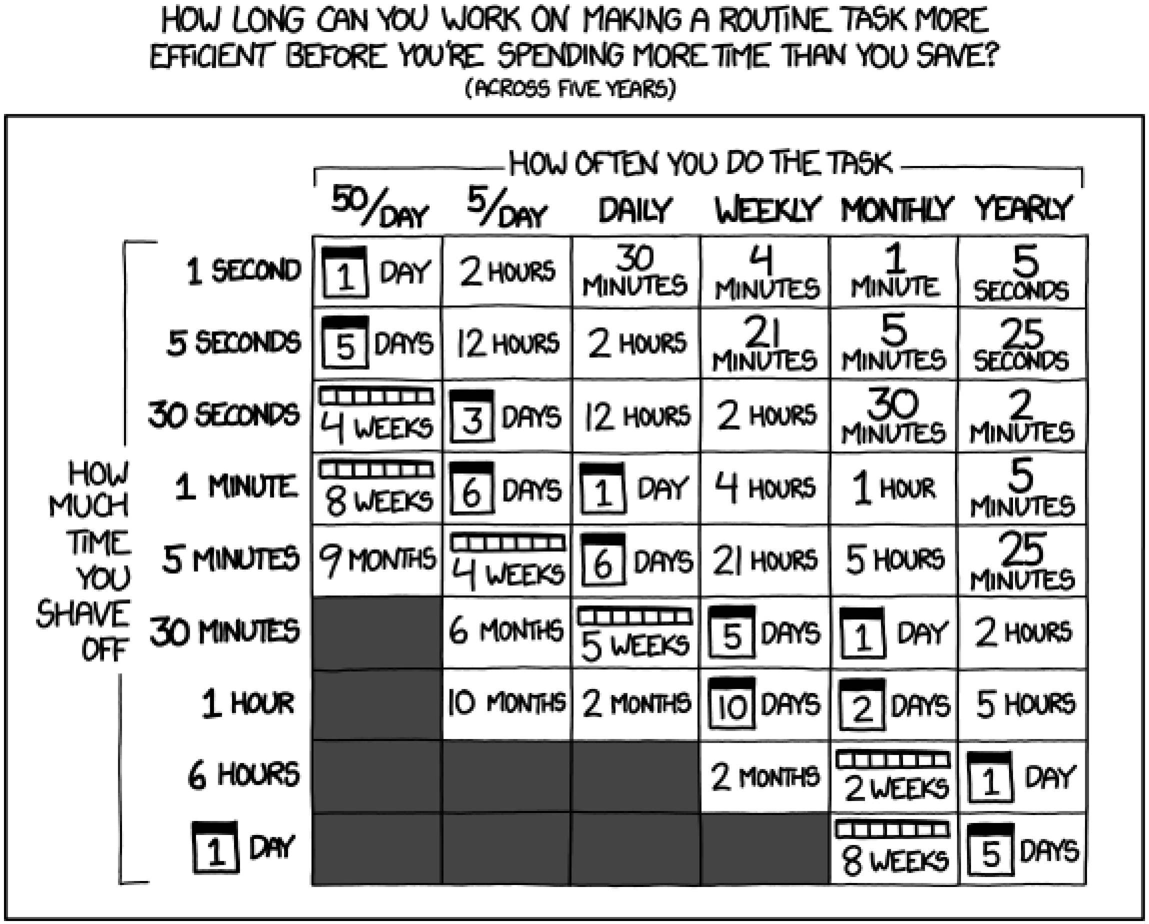
- Rewrite in Clojure
    - Learning experience
    - Less code
    - More performance (5x speed)
    - Simple cross platform deployment with a single jar

- Cool things about Clojure
    - map, filters, reduce and lambdas
    - REPL
    - Hot code reloading without service being down
    - The entire language is available all the time

- Pain points about Clojure
    - Knock knock java!
    - Error messages
    - Editor and tooling
    - Not useful for short scripts

- Automation
    - Less boredom
    - Refines skills and gives more end to end experience and responsibility
    - Useful to others and FOSS
    - More fun and wizadry

- Obligatory xkcd



HOW LONG CAN YOU WORK ON MAKING A ROUTINE TASK MORE EFFICIENT BEFORE YOU'RE SPENDING MORE TIME THAN YOU SAVE?
(ACROSS FIVE YEARS)

	HOW OFTEN YOU DO THE TASK					
HOW MUCH TIME YOU SHAVE OFF	50/DAY	5/DAY	DAILY	WEEKLY	MONTHLY	YEARLY
1 SECOND	1 DAY	2 HOURS	30 MINUTES	4 MINUTES	1 MINUTE	5 SECONDS
5 SECONDS	5 DAYS	12 HOURS	2 HOURS	21 MINUTES	5 MINUTES	25 SECONDS
30 SECONDS	4 WEEKS	3 DAYS	12 HOURS	2 HOURS	30 MINUTES	2 MINUTES
1 MINUTE	8 WEEKS	6 DAYS	1 DAY	4 HOURS	1 HOUR	5 MINUTES
5 MINUTES	9 MONTHS	4 WEEKS	6 DAYS	21 HOURS	5 HOURS	25 MINUTES
30 MINUTES		6 MONTHS	5 WEEKS	5 DAYS	1 DAY	2 HOURS
1 HOUR		10 MONTHS	2 MONTHS	10 DAYS	2 DAYS	5 HOURS
6 HOURS				2 MONTHS	2 WEEKS	1 DAY
1 DAY					8 WEEKS	5 DAYS

Questions

```
(map answer (filter questions? session))
```

Thanks

- https://yhatt.github.io/marp/ (Slides in Markdown)