

Quicklisp

On Beyond Beta

Zach Beane / zach@quicklisp.org / [@quicklisp](https://twitter.com/quicklisp)

My Perspective

Quicklisp Overview

Easy to Install

```
curl qucklisp-install.net |  
sudo bash
```

Just Kidding!

Works (Almost)
Everywhere

Tested and Updated
Monthly

Everything Versioned
Together

Practical Interface

Practical Interface

- quickload - used daily

Practical Interface

- quickload - used daily
- system-apropos - used almost daily

Practical Interface

- quickload - used daily
- system-apropos - used almost daily
- update-dist - used monthly

Practical Interface

- quickload - used daily
- system-apropos - used almost daily
- update-dist - used monthly
- update-client - used a few times per year

Quicklisp isn't perfect
but it's tough all over

Node startup scores \$8M, launches Private Modules

Only \$7/person/month
VentureBeat, April 14, 2015

Go: How to deal with dependencies and their versions?

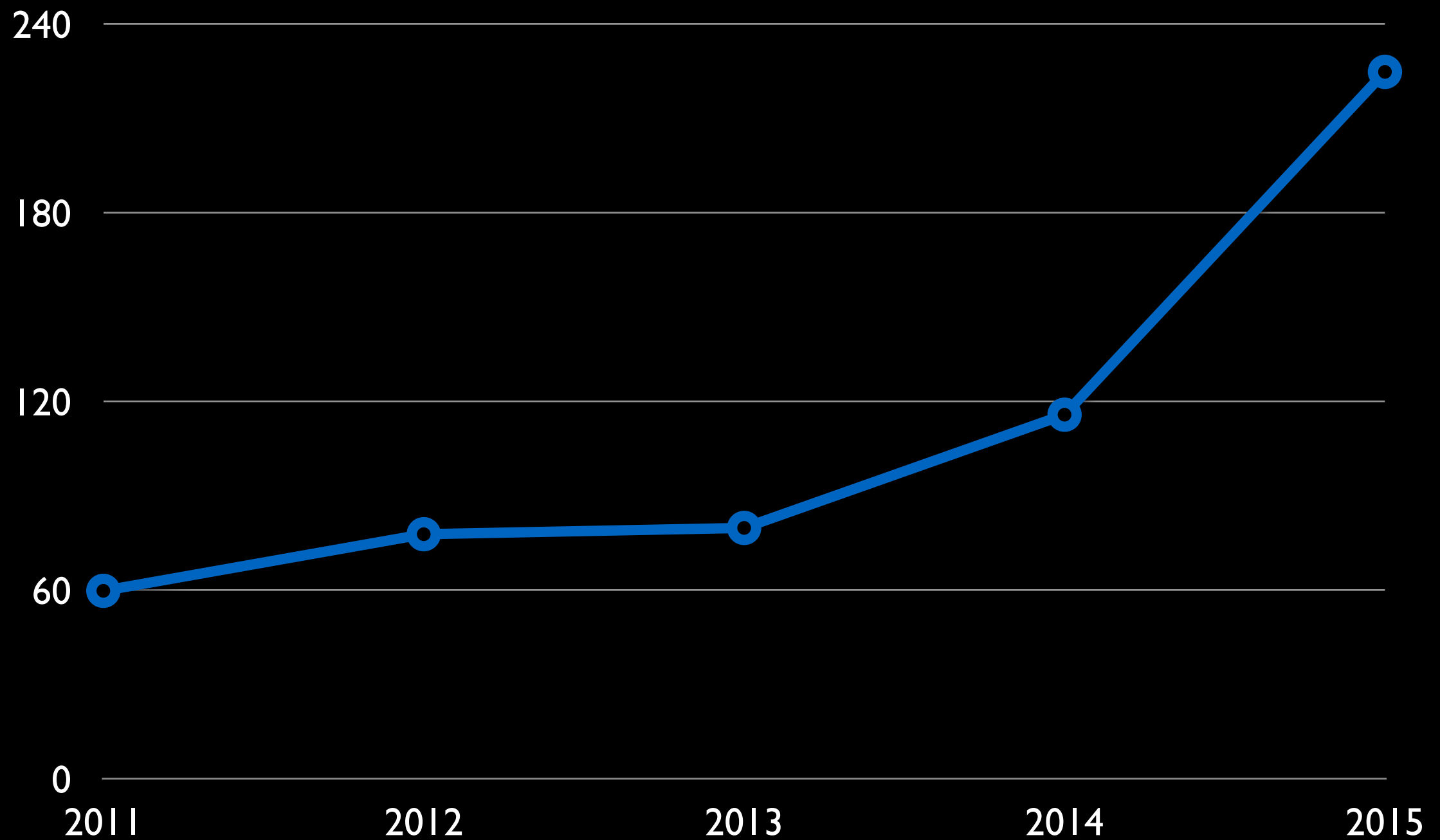
We've never recommended a particular answer.

In Google's internal source tree, we vendor (copy) all our dependencies into our source tree.

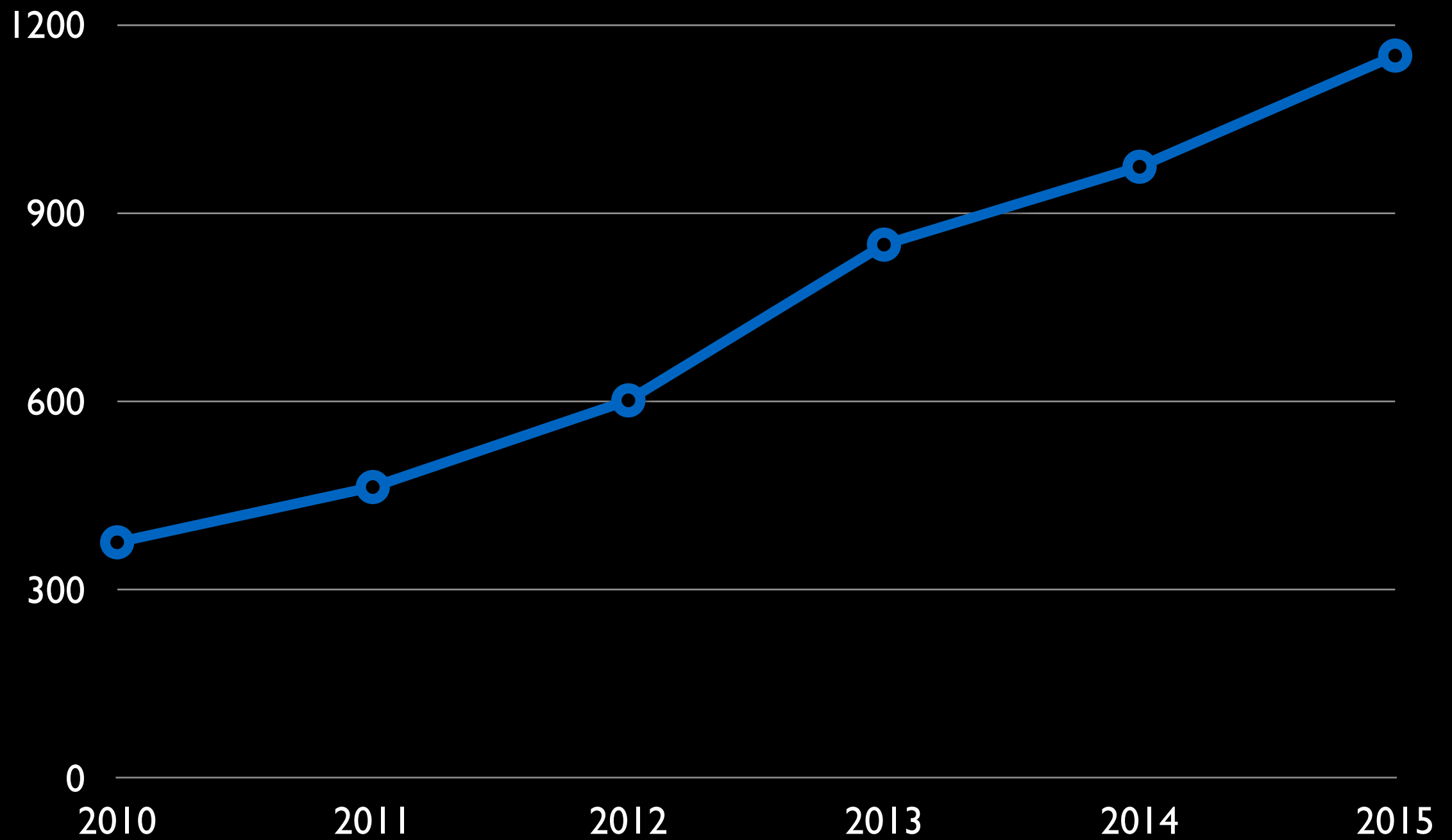
Brad Fitzpatrick, March 2, 2015

Trends in Quicklisp and Common Lisp

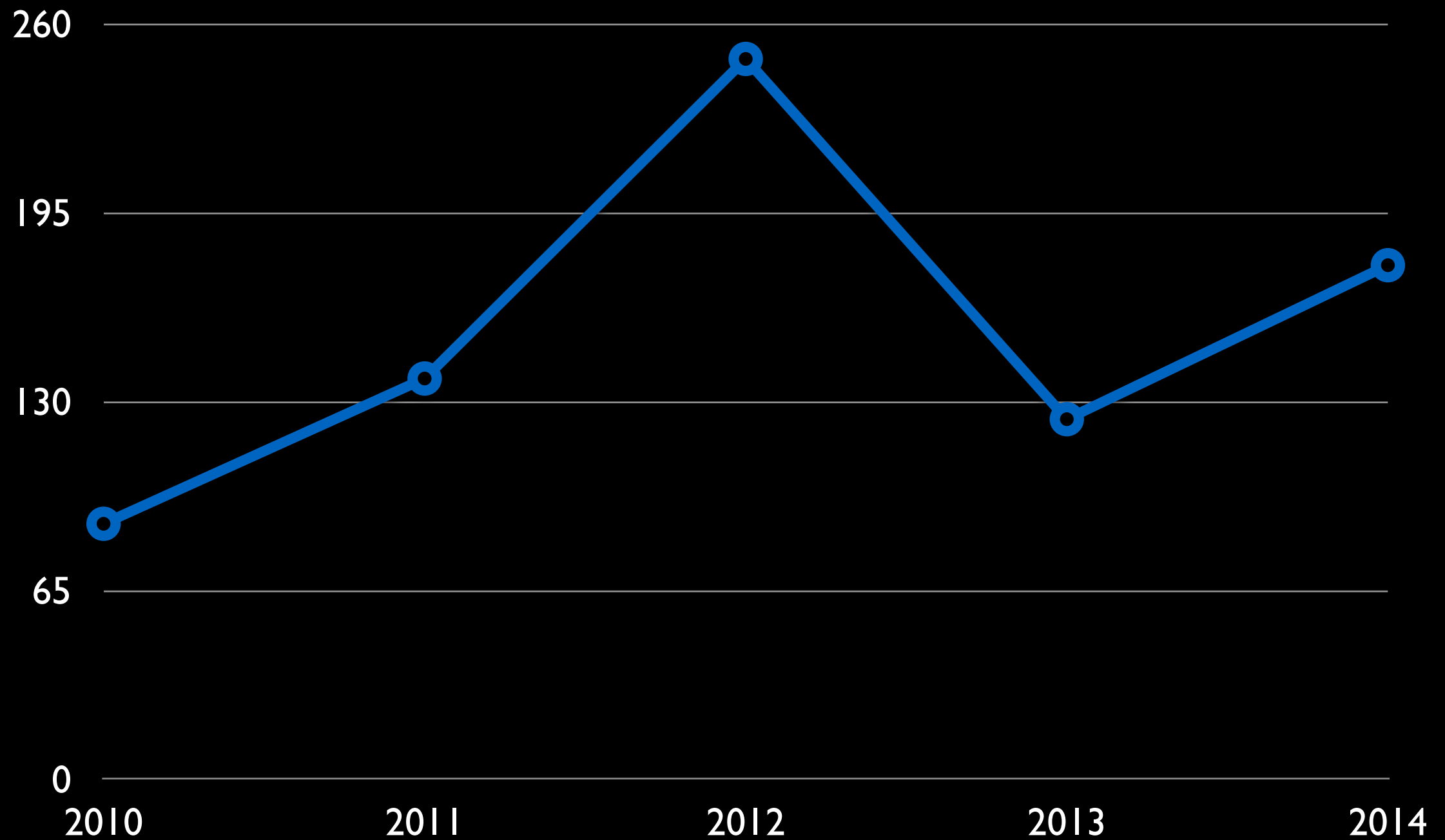
Installations per day



Project Count



New Projects per Year



Top Libraries, 2015

1. Alexandria
2. cl-ppcre
3. closer-mop
4. trivial-features
5. babel

Top Libraries, 2015

1. Alexandria - 2007
2. cl-ppcre - 2005
3. closer-mop - 2005
4. trivial-features - 2007
5. babel - 2007

Top Libraries, 2015

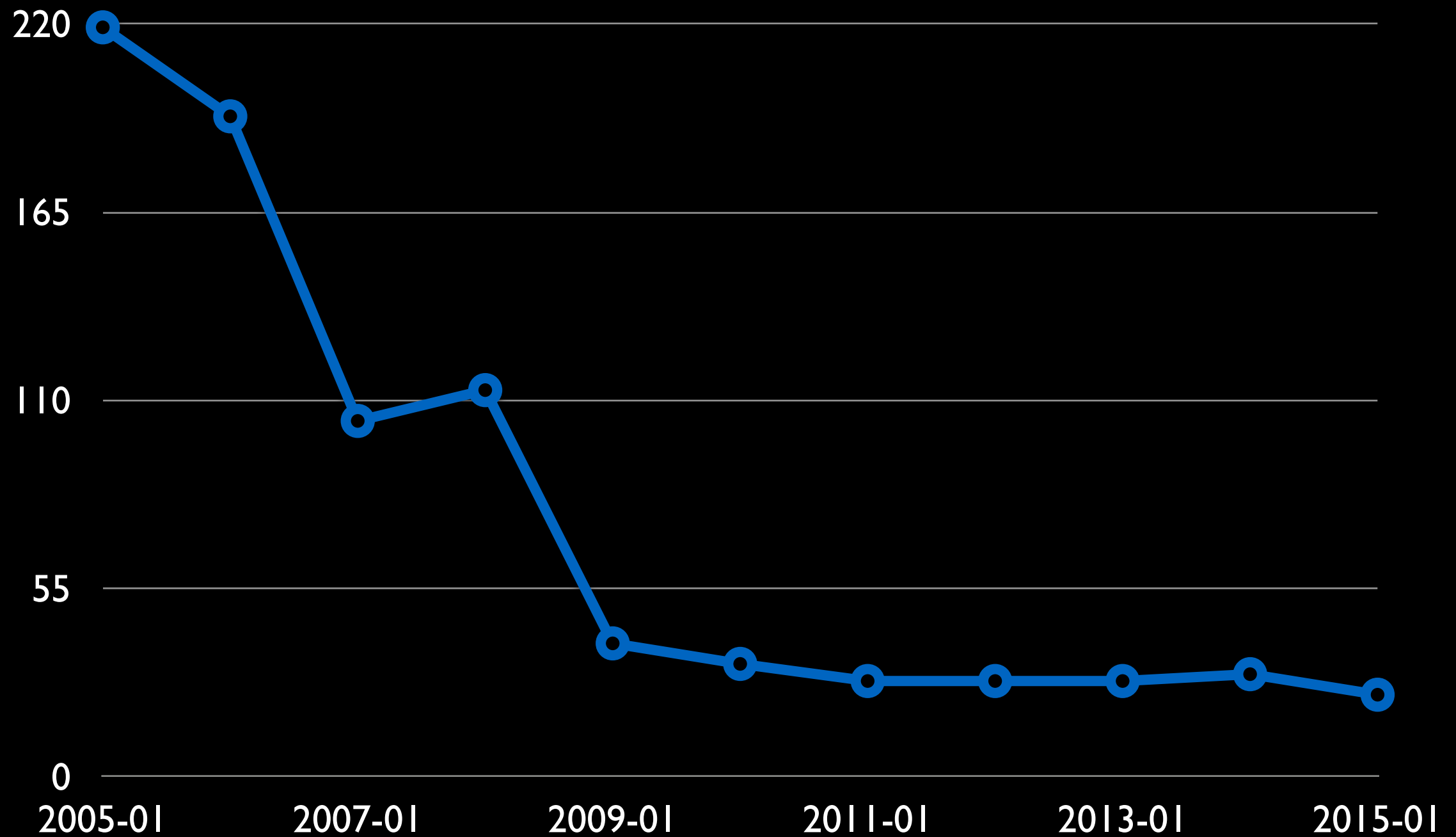
Top Libraries, 2015

- Of the top 25 downloads, 2 are newer than 2010 (optima, nibbles)

Top Libraries, 2015

- Of the top 25 downloads, 2 are newer than 2010 (optima, nibbles)
- Of the top 100, 31 are newer than 2010

Planet Lisp posts per month



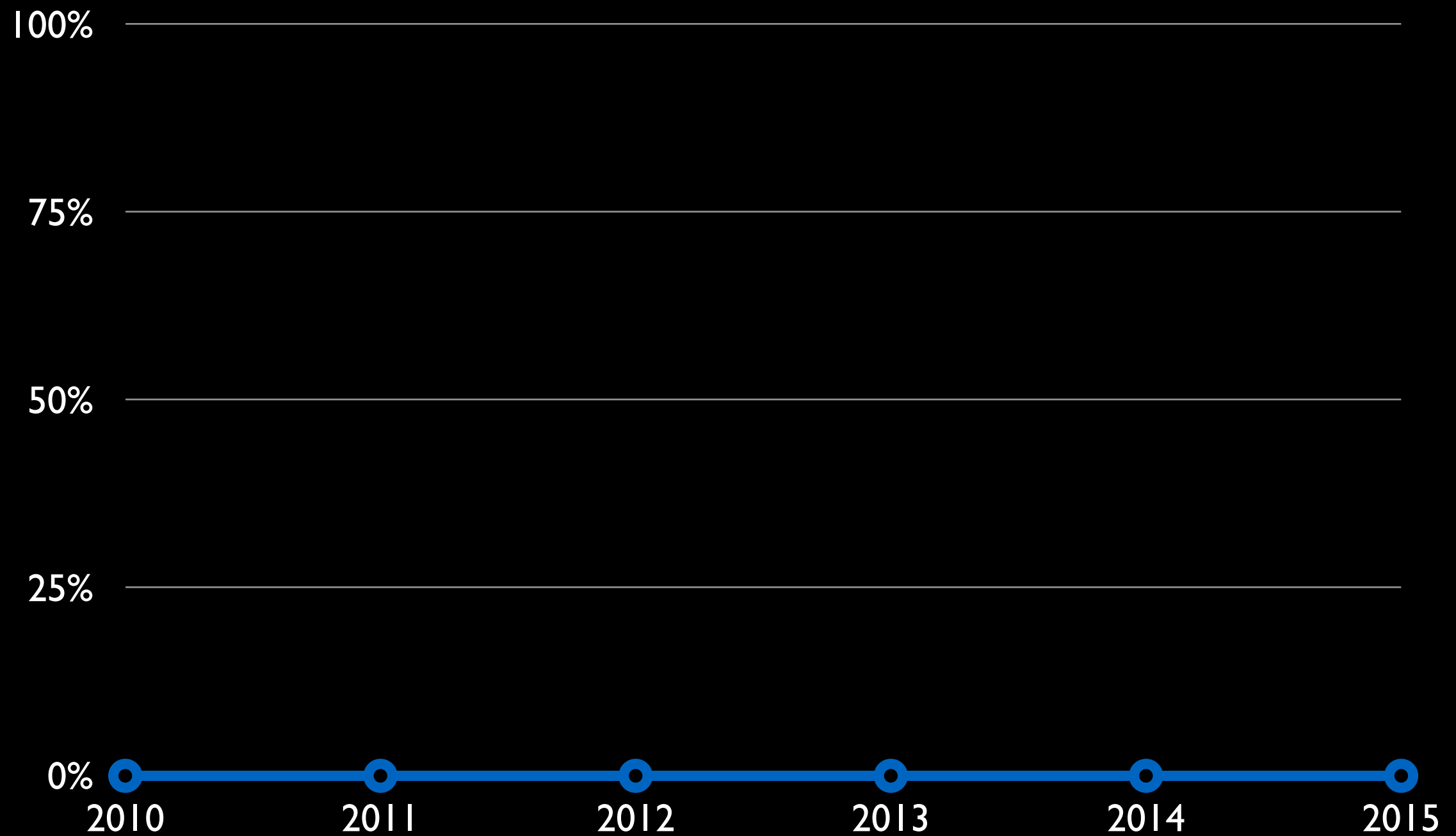
Lemonodor-fame is but a
hack away!

[your name here]-fame
is but a hack away!

Trends in Getting Help

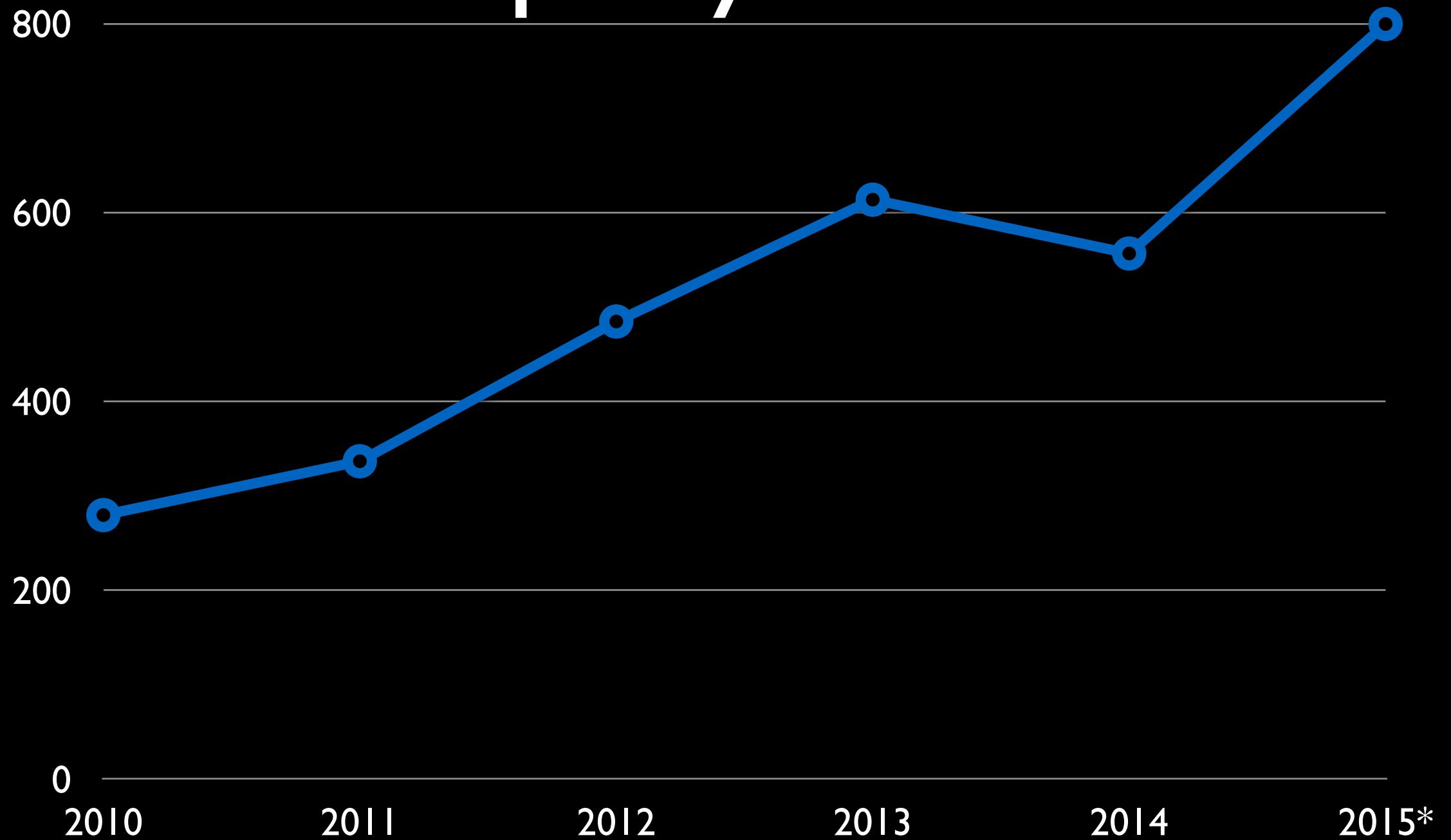
Usenet

Usenet User-Friendliness



Stack Overflow

StackOverflow Questions per year



Other Sources

- Reddit /r/lisp and /r/learnlisp
- Twitter
- Implementation mailing lists
- lispforum.com
- “Where to get help with Common Lisp”

Building On Quicklisp

- cl-test-grid
- quickdocs.org
- qplot

Implementations

Dan Weinreb's 2010 Survey

- Allegro CL
- Armed Bear CL
- CMUCL
- Corman CL
- Clozure CL
- ECL
- GNU CL
- GNU CLISP
- LispWorks
- Scieneer CL
- SBCL

New Releases Since 2010

- Allegro CL
- Armed Bear CL
- CMUCL
- ~~Corman CL~~
- Clozure CL
- ECL
- GNU CL
- ~~GNU CLISP~~
- LispWorks
- ~~Scieneer CL~~
- SBCL

New or Relicensed Since 2010

- Clasp
- Corman CL (newly MIT licensed)
- Mezzano CL
- MKCL (fork of ECL)
- MOCL
- Eclipse CL (newly MIT licensed)

People Like Quicklisp
...and use it, too.

Beyond Beta

What is Quicklisp
supposed to do?

Make it easy to confidently
build on the work of
others.

(Also, work on as many
Common Lisps on as many
platforms as possible.)

(Short slogans are hard.)

What does Quicklisp get
right already?

Make it easy...

Make it easy...

- Easy to install

Make it easy...

- Easy to install
- Easy to use

Make it easy...

- Easy to install
- Easy to use
- Easy to update

Make it easy...

- Easy to install
- Easy to use
- Easy to update
- Easy to learn everything it does

Make it easy...

- Easy to install
- Easy to use
- Easy to update
- Easy to learn everything it does
- Easy to find software and get an idea of its utility

...to confidently...

...to confidently...

- Low risk to try libraries and updates

...to confidently...

- Low risk to try libraries and updates
 - Updates are tested

...to confidently...

- Low risk to try libraries and updates
 - Updates are tested
 - Undo is possible

...to confidently...

- Low risk to try libraries and updates
 - Updates are tested
 - Undo is possible
- Possible to stick to a known-working version of everything

...to confidently...

- Low risk to try libraries and updates
 - Updates are tested
 - Undo is possible
- Possible to stick to a known-working version of everything
- Trace software to its origin

...to confidently...

- Low risk to try libraries and updates
 - Updates are tested
 - Undo is possible
- Possible to stick to a known-working version of everything
- Trace software to its origin
- Know what to expect

...build on the work of
others

...build on the work of others

- Build on reusable libraries

...build on the work of others

- Build on reusable libraries
- Build on Quicklisp itself via flexible protocols, hooks, and data formats

...build on the work of others

- Build on reusable libraries
- Build on Quicklisp itself via flexible protocols, hooks, and data formats
- Code is is there, waiting!

...build on the work of others

- Build on reusable libraries
- Build on Quicklisp itself via flexible protocols, hooks, and data formats
 - Code is is there, waiting!
- Avoid library licensing problems

Closing the Gap

Closing the Gap

- Write documentation

Closing the Gap

- Write documentation
- Add missing features

Closing the Gap

- Write documentation
- Add missing features
- Gather and share metadata

Closing the Gap

- Write documentation
- Add missing features
- Gather and share metadata
- Support discovery and feedback

Documentation,
Documentation,
Documentation

User Documentation

Useful Obscure Tools

ql:where-is-system

ql:where-is-system

ql:where-is-system

- /usr/share/common-lisp/cl-foobar/
=> *Potentially broken Debian package*

ql:where-is-system

- /usr/share/common-lisp/cl-foobar/
=> *Potentially broken Debian package*
- /home/user/.sbcl/mumble/cl-foobar/
=> *Potentially ancient asdf-install artifact*

ql:where-is-system

- /usr/share/common-lisp/cl-foobar/
=> *Potentially broken Debian package*
- /home/user/.sbcl/mumble/cl-foobar/
=> *Potentially ancient asdf-install artifact*
- Fortunately, not very common any more

~/quicklisp/local-projects

~/quicklisp/local-projects

- Automagically makes systems loadable

~/quicklisp/local-projects

- Automagically makes systems loadable
- Uses `ql:*local-project-directories*`

~/quicklisp/local-projects

- Automagically makes systems loadable
- Uses `ql:*local-project-directories*`
- Has `ql:register-local-projects` as a fallback

~/quicklisp/local-projects

- Automagically makes systems loadable
- Uses `ql:*local-project-directories*`
- Has `ql:register-local-projects` as a fallback
- Transparent, reliable, useful

~/quicklisp/local-projects

- Automagically makes systems loadable
- Uses `ql:*local-project-directories*`
- Has `ql:register-local-projects` as a fallback
- Transparent, reliable, useful
- Barely documented by a blog post from 2011, and the README files of a dozen github repos

Many, many more
30+ user functions available

Not Just Commands

- Quicklisp has a flexible initialization sequence
- The Quicklisp install directory can be easily moved around, by design
- Barely documented in FAQ and some blog posts

Developer Documentation

Developer Documentation

- Data model and protocols
- Data formats

Data Model

- Dist
 - Release
 - System

Data Model

- Dist - quicklisp 2015-04-07
- Release - vecto 1.2.1
 - System - vecto

Protocols and Hooks

Protocols and Hooks

- install, uninstall, installedp, ensure-installed

Protocols and Hooks

- install, uninstall, installedp, ensure-installed
- enable, disable

Protocols and Hooks

- install, uninstall, installedp, ensure-installed
- enable, disable
- provided-releases, provided-systems

Protocols and Hooks

- install, uninstall, installedp, ensure-installed
- enable, disable
- provided-releases, provided-systems
- inhibit-subscription

Protocols and Hooks

- install, uninstall, installedp, ensure-installed
- enable, disable
- provided-releases, provided-systems
- inhibit-subscription
- all-dists, enabled-dists

Protocols and Hooks

- install, uninstall, installedp, ensure-installed
- enable, disable
- provided-releases, provided-systems
- inhibit-subscription
- all-dists, enabled-dists
- *dist-enumeration-functions*

Metadata Format

Why Developer Documentation?

Missing Features

Missing Features

- Fetching software: more than just HTTP

Missing Features

- Fetching software: more than just HTTP
- Bundling libraries for use without Quicklisp

```
(ql:bundle-system '("hunchentoot" "veto")  
                  :to "src/webbundle/")
```

restart without quicklisp

```
(load "src/webbundle/bundle.lisp")  
(asdf:load-system "hunchentoot")
```

Available now!

Metadata

Metadata

- See more descriptive info than just the system name, when available; searchable

Metadata

- See more descriptive info than just the system name, when available; searchable
- License info

Metadata

- See more descriptive info than just the system name, when available; searchable
- License info
- Library provenance: where, when, and enough descriptive info to get the original

Metadata

- See more descriptive info than just the system name, when available; searchable
- License info
- Library provenance: where, when, and enough descriptive info to get the original
- (ql:hack-on-project “vecto”)
Cloning into 'vecto'...

Discovery and Feedback

Discovery and Feedback

Discovery and Feedback

- What does a library do?

Discovery and Feedback

- What does a library do?
- What library might be suitable for my needs?

Discovery and Feedback

- What does a library do?
- What library might be suitable for my needs?
- If there's only one, do people use it? Is it actively developed?

Discovery and Feedback

- What does a library do?
- What library might be suitable for my needs?
- If there's only one, do people use it? Is it actively developed?
- If there are multiple options, which might be preferable?

Discovery and Feedback

- What does a library do?
- What library might be suitable for my needs?
- If there's only one, do people use it? Is it actively developed?
- If there are multiple options, which might be preferable?
- What do others think of the library?

Roadmap out of Beta

- Documentation
 - User and Developer
- Features
 - Fetch Control
 - Library Bundles
- Metadata
 - Descriptions, Licenses, and Provenance
- Discovery and Feedback
 - Gather, Index, Rank, and Review

Timeline?

Depends on time & funding.

Want something implemented sooner?

Want to work on something yourself?

Talk to me afterwards!

Thank You!

Questions?

Zach Beane / zach@quicklisp.org / [@quicklisp](#)