

Would you start learning Smalltalk?

[closed]

Asked 16 years, 2 months ago Modified 9 years ago Viewed 14k times



85



Closed. This question is [opinion-based](#). It is not currently accepting answers.



Want to improve this question? Update the question so it can be answered with facts and citations by [editing this post](#).

Closed 8 years ago.

[Improve this question](#)

My questions is simple!

1. Would you start learning Smalltalk if you had the time? Why? Why not?
2. Do you already know Smalltalk? Why would you recommend Smalltalk? Why not?

Personally I'm a Ruby on Rails programmer and I really like it. However, I'm thinking about Smalltalk because I read various blogs and some people are calling Ruby something like "Smalltalk Light". The second reason why I'm interested in Smalltalk is [Seaside](#).

Maybe someone has made the same transition before?

EDIT: Actually, what got me most excited about Smalltalk/Seaside is the following Episode of WebDevRadio: [Episode 52: Randal Schwartz on Seaside \(among other things\)](#).

smalltalk

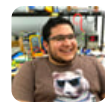
seaside

Share

Improve this question

Follow

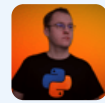
edited Dec 18, 2009 at 0:14



Nathan Campos

29.5k ● 61 ● 200 ● 307

asked Sep 29, 2008 at 14:57



Christoph Schiessl

6,858 ● 4 ● 35 ● 45

20 Answers

Sorted by:

Highest score (default)



64



If you like Ruby you'll probably like Smalltalk. IIRC Seaside has been ported to the [Gemstone VM](#), which is part of their Gemstone/S OODBMS. This has much better thread support than Ruby, so it is a better back-end for a high-volume system. This might be a good reason to take a close look at it.

Reasons to learn Smalltalk:

- It's a really, really nice programming environment. Once you've got your head around it (it tends to be a

bit of a culture shock for people used to C++ or Java) you'll find it to be a really good environment to work in. Even a really crappy smalltalk like the Old Digitalk ones I used is a remarkably pleasant system to use. Many of the old XP and O-O guru types like [Kent Beck](#) and [Martin Fowler](#) cut their teeth on Smalltalk back in the day [and can occasionally be heard yearning for the good old days in public](#) (Thanks to Frank Shearer for the citation, +1) - Agile development originated on this platform.

- It's one of the most productive development platforms in history.
- Several mature implementations exist and there's a surprisingly large code base out there. At one point it got quite trendy in financial market circles where developer productivity and time-to-market is quite a big deal. Up until the mid 1990s it was more or less the only game in town (With the possible exception of LISP) if you wanted a commercially supported high-level language that was suitable for application development.
- Deployment is easy - just drop the image file in the appropriate directory.
- Not really a reason, but the [Gang of Four Book](#) uses Smalltalk for quite a few of their examples.

Reasons not to learn Smalltalk:

- It's something of a niche market. You may have trouble finding work. However if you are producing

some sort of [.com application where you own the servers](#) this might not be an issue.

- It's viewed as a legacy system by many. There is relatively little new development on the platform (although Seaside seems to be driving a bit of a renaissance).
- It tends not to play nicely with traditional source control systems (at least as of the early-mid 90's when I used it). This may or may not still be the case.
- It is somewhat insular and likes to play by itself. Python or Ruby are built for integration from the ground up and tend to be more promiscuous and thus easier to integrate with 3rd party software. However, [various other more mainstream](#) systems suffer from this type of insularity to a greater or lesser degree and that doesn't seem to impede their usage much.

Share Improve this answer

edited Mar 5, 2010 at 8:59

Follow

answered Sep 29, 2008 at 15:51



[ConcernedOfTunbridgeWells](#)

66.5k ● 15 ● 148 ● 198

-
- 2 Ruby has also been ported to Gemstone: it's called Maglev: infoq.com/news/2008/04/maglev-gemstone-builds-ruby
– [brian d foy](#) Sep 29, 2008 at 22:17
-

1 What do you mean by "it tends not to play nicely with traditional source control systems"? – [detly](#) Mar 17, 2010 at 14:41

6 The way a traditional Smalltalk stored source code meant that it was embedded within the image and associated sources and changes files. You didn't get a nice clean set of program files like you would get with (say) C. Smalltalk systems tended to require specialised smalltalk-aware source control systems, which tended to be (a) proprietary, (b) expensive and (c) incompatible with everything else. – [ConcernedOfTunbridgeWells](#) Mar 17, 2010 at 15:32

2 And hopefully Monticello will address (a) and (b) of these issues. Squeak's already using it heavily, and SqueakSource has many many projects using it. It still has warts though. Once the git-Monticello projects are actually in use, (c) will hopefully also fall by the wayside. – [Frank Shearar](#) Oct 16, 2010 at 7:18

3 I generally like the answer but Smalltalk plays okay with third party software. The fact you don't see many many APIs is not because you technically can't integrate them, is because it's a niche (so less people needs external software). A secondary reason to that effect (that also isn't technical) is that smalltalkers will prefer to deal with problems in smalltalk rather than in other less-fun technologies. – [Sebastian Sastre](#) Dec 16, 2011 at 18:15



Well, since you mentioned me by name, I feel I should chime in.

48



As I said in that podcast interview, and as I have repeatedly demonstrated in my blog at <http://MethodsAndMessages.vox.com/>, this is "*the year of*



smalltalk". And having now done Smalltalk advocacy for the past ten months, I can see that it really is happening. More customers are turning to Smalltalk and Seaside, and the Smalltalk vendors are all working hard to capture this new influx of attention. More larger Smalltalk conferences are being planned. More job postings are being posted. More blog postings are being made.

If you turn to Smalltalk today, you are not alone. There are many others who are out there as well.

Edit

Well, a number of years later, I'm now recommending [Dart](#) instead. It's a great language originated by Google but now owned by an ECMA committee. It runs serverside in node.js style, but also clientside in modern browsers by transpiling to JavaScript. Lots of good books, blogs, help channels, IDE support, [public live pastebin](#). I think it's definitely got legs... enough so that I'm writing courseware to teach it onsite or online, and I'm pretty sure there's a book or two in the works from me. And Gilad Bracha, an old-time Smalltalker is a major contributor to the design, so there's a lot of Smalltalk in Dart.

Share Improve this answer

edited Aug 16, 2015 at 23:56

Follow

answered Sep 29, 2008 at 22:23



Randal Schwartz

-
- 3 That's why Stack Overflow is cool! Thank you for your answer! – [Christoph Schiessl](#) Sep 30, 2008 at 9:49
-
- 1 Out of curiosity, can you elaborate on the influx of attention - any ideas where the mindshare is coming from?
– [ConcernedOfTunbridgeWells](#) Nov 12, 2008 at 22:15
-
- 1 Ruby certainly helped, I think Javascript and Objective-C too. For people that don't get the pure OO thing, those languages provide a familiar starting point from which one can extrapolate how Smalltalk shines. – [Damien Pollet](#) Apr 9, 2009 at 21:31
-
- 1 Then the Seaside guys did some good PR, Avi Bryant with DabbleDB and the Maglev coup at Rails conf, and Randal is quite vocal too :) Also Google V8's Lars Bak comes from Self... there was many tiny cues like that, all hinting at Smalltalk and Lisp, strange languages from the future-in-the-past :) – [Damien Pollet](#) Apr 9, 2009 at 21:36
-

**32**

Smalltalk is a good language to learn, and the great thing is that it only takes a day to do it. It's a lot more than just an academic language. People are building huge, scalable, replicable applications handling billions of dollars. They just don't talk about it much. See, for instance, [GemStone and Orient Overseas Container Lines: A Shipping Industry Case Study](#).

Seaside is a good reason to learn Smalltalk, but I don't think you'll find it orders of magnitude better than Rails.

The thing that convinced me was GemStone. I really like Gemstone's GLASS (GemStone, Linux, Apache, Smalltalk, Seaside). The killer part of that is GemStone, which handles all the object persistence for you mostly without you thinking about it. Seeing some of their demos and hearing about what people are doing with GemStone reset my idea of what "big application" meant.

The part that bugs me the most about Rails is the object-relational mapping. That's nothing against Ruby because it sucks just as hard in GLORP (which handles ActiveRecord for Smalltalk), or Perl, or anything else. Mapping objects to database tables is just painful. With GemStone, thinking about the database disappears, so the work with the database disappears too. It's like a huge stone (or a troop of monkeys) is taken off my back.

Share Improve this answer

edited Mar 17, 2010 at 14:33

Follow

answered Sep 29, 2008 at 22:27



brian d foy

132k ● 31 ● 211 ● 604

2 I wish I could vote twice... I totally agree.

– Sébastien RoccaSerra Sep 30, 2008 at 14:51

2 +1 for showing ODB's superiority over ORM. going from one to the other is definitely something you can *feel* – ryan0 Sep 10, 2012 at 22:12



> couldn't find a Smalltalk development environment that didn't cost both arms and a leg

17

Google - free smalltalk



Cincom Smalltalk, Squeak, GNU Smalltalk



Share Improve this answer

answered Sep 29, 2008 at 15:11



Follow



Isaac Gouy

-
- 1 Squeak is really nice! Pure OOP, completely graphic, lovely user interface, open source (some disagree on this), more or less portable... – [Joe Pineda](#) Sep 29, 2008 at 20:38
-
- 2 +1 for Squeak, and it has a great community with a nice mailing list for beginners. – [Sébastien RoccaSerra](#) Sep 30, 2008 at 14:47
-
- 1 +1 for Squeak, the object explorer (or whatever it's called) is invaluable for learning how the object model works. – [Mauricio Scheffer](#) Dec 7, 2008 at 4:09
-
- 5 Pharo is free and has all the modern tools.. – [Dafydd Rees](#) Aug 4, 2010 at 22:04
-

Note: GNU Smalltalk is headless, which means you can only use it via command line, not via an interactive development environment. Which was kind of the essence when the original Smalltalk was created. – [Sander de Jong](#) Dec 27, 2022 at 11:25



Learning Smalltalk will give you a grounding in object oriented software development from the perspective of

17



the man who invented OO (Alan Kay). The idea of a overlapping windowing environment came from Smalltalk.

A stumbling block to learning Smalltalk is that it is a message passing system with a strange syntax for flow control like:

```
i < 60  
  ifTrue: [ self walk ]
```

It has a very mature class library that has a consistency I've not seen too many places. The class library in all environments (even commercial Smalltalks) has available source which allows you to learn from the masters of the language. When programming Smalltalk, I always ask the question how is it done in the environment.

Smalltalk is generally implemented in an image which is a live environment for all the objects in your system.

The interactive debugger really separates Smalltalk from Ruby.

Seaside is the web development framework and has given Smalltalk a new spotlight. It is a continuation based environment that allows for intra-hit debugging and a smooth Rich Client type development experience (top application flow can be designed in a single method). It's integration with script.aculo.us has been done in such a way that it is easily called from within Smalltalk.

[Share](#) [Improve this answer](#)



Kevin Driedger

54k ● 15 ● 51 ● 55

2 The syntax is not strange - its actually VERY elegant. It must be read as: hey, whatever you outcome of "i<60" are, here is a piece of code (code-object passed as argument), to be evaluated if you think you are true. The funny thing is, that even works, if "i<60" returns a fuzzy, statistical thingy.

– [blabla999](#) Jan 16, 2009 at 12:30

1 ...and that fuzzy, statistical thing can then decide (as a method) how to deal with that. There is not a single builtIn operator or control structure which cannot be redefined this way. – [blabla999](#) Jan 16, 2009 at 12:31



11



Nigel, one quote I have is this:

Although it's now a long time since I did anything with it, I nominate Smalltalk, I still haven't come across anything quite like it for being able to transfer thoughts into computer code. It's not just the language: It's the wonderful browser environment, the libraries, and the culture of writing clear, well-designed code as quickly as anything else can crank out spaghetti. When the participants at JavaOne were extolling how Java was so much more productive than anything else, I needed a brown paper bag. Oh well, back to sorting out my classpaths... -- Martin Fowler (Software Development Magazine, Jan 2001)

I found it [here](#).

Share Improve this answer

answered Sep 30, 2008 at 13:17

Follow



Frank Shearar

17.1k ● 8 ● 71 ● 95



11



Would disagree with the poster who reckons you wouldn't use Smalltalk for large apps – that's precisely where it shines. But I have created fairly groovy (note lowercase) prototype apps in under a week too.

I learned OO in ST starting in 92, incredibly glad I did so. It gave me a real background in OO. Thinking in classes. No types. ST has a real emphasis on messaging. If you want to know something send an object a message and get an answer. IMHO, the ethos and the IDE really encourage you to do the right thing with your coupling and cohesion.

In my Java day job, I'm stuck with files, generics, IDE's like eclipse that are orders of magnitude less productive than any ST IDE. I was using ST the only time I finished a development ahead of schedule. In fact it was so productive, and we got so much reuse I had to be moved off to another project, as I had nothing to do! (Ok, maybe I could have spent time learning to estimate...)

Download squeak, find a good book and play. Only downside is that if your day gig is using Java or C#, you'll end up wishing you could use ST. You'd get home sooner.

Chris Brooks

Share Improve this answer

answered Oct 14, 2008 at 11:41

Follow



Chris Brooks

299 ● 1 ● 2



I recommend everybody to learn Lisp (Scheme) or Smalltalk.

10



Smalltalks have wonderful IDEs which you dont want to miss once you got over the culture shock. And yes, there are more than one free ones: Squeak, Dolphin, Smalltalk/X, and Visualworks (Non-Comercial).



Lisp may be even cleaner in its mathematic foundation, though.

regards

PS: actually I recommend learning both !

Share Improve this answer

edited Dec 23, 2008 at 0:59

Follow

answered Dec 23, 2008 at 0:53



blabla999

3,200 ● 23 ● 24



I do not know Ruby..

9

Smalltalk is a **pure** OO language. If you feel the need to really understand OO, and not just the simulated OO of



most popular 'OO' languages (like C++, Java, etc), then I would recommend that you play with smalltalk.



In smalltalk everything is an object, with attributes, behavior and meta. In the simulations you have data types that you use in your objects.

I would say play with it, you will only benefit.

Share Improve this answer

answered Sep 29, 2008 at 15:09

Follow



[slashmais](#)

7,137 ● 9 ● 56 ● 80

2 In SmallTalk, even "classes" are objects, instances of the class Class!!! This drove me mad until I grokked it!
– [Joe Pineda](#) Sep 29, 2008 at 20:43

2 Ruby is also a pure OO language in the sense that even Classes are objects. So are blocks, methods and the reference of a null pointer. – [Alan](#) Oct 1, 2008 at 3:20

5 Ruby takes (Matz will freely state this) many ideas from Smalltalk. In many ways it can be viewed as a 'Pragmatic' (apologies to Hunt and Thomas) Smalltalk-like language.
– [ConcernedOfTunbridgeWells](#) Oct 1, 2008 at 14:44



I'm totally in your shoes. Im using RoR and looking into Smalltalk land. Here's some pros & cons I find important:

8

Pros:



- Mature & stable environment
- Fast development cycle





- Makes you think more and write less

Cons:

- Requires different thinking
- Still didn't quite grasp it

It's quite funny how I got to know about Smalltalk. It was this one thing that kept popping up in Google results when searching for Lisp and Erlang stuff. One day I checked it out and was amazed with nice windows environment. Few moments later I've found Aida/Web framework. I was hooked and started learning Smalltalk through web development with this framework.

Still not quite there, but it's so damn interesting I just can't sit still... :-) I'm having fun again.

Share Improve this answer

answered Jan 16, 2009 at 12:05

Follow



Damir Horvat

1,037 ● 1 ● 10 ● 12



6

Would not start learning it if I had the time. Why not? Because it would be more productive and lucrative financially to learn C# or Java.



On the other hand if your a hobbyist, and would like to go on an archeological dig, then I'd suggest spending some time looking at the What, When, Why and how of smalltalk by researching Alan Kay. Fascinating story and an incredible person (after all, he earned the Turning



Award). Then maybe play with squeak a little to get a feeling for the language. After this you might have a newly found respect/understanding of blocks, closures, and Object Oriented principles.

I know and use Smalltalk, have for about 15 years, still **maintaining** it, and would not recommend Smalltalk to a friend. Why not? Employment is a good thing to have and keep getting. Although you can learn a lot from Smalltalk you can't easily turn that into gainfully being employed in this day and age.

Also, you appeared to be excited over Seaside and I would assume the Seaside/GemStone partnership. I've used GemStone for quite some time and the two together are very appealing. I hope they can get the market share and momentum required to be successful.

Share Improve this answer

edited Dec 7, 2008 at 3:56

Follow

answered Dec 7, 2008 at 3:32



[daduffer](#)

347 ● 2 ● 4



Don't! If you really start learning it, you might not want to program in something else anymore ever.

6

This may be not true, if you are a lisp programmer.



Share Improve this answer



Follow

answered Jan 22, 2010 at 0:12



[Richard Durr](#)

3,231 ● 3 ● 22 ● 26



5



Absolutely, learn Smalltalk! This is 2015 and Smalltalk is on the rise again, thanks to [Pharo](#). Pharo is FREE. Pharo is evolving quickly into a powerful enterprise tool. At Version 4.0, and soon to be 5.0, it has matured a great deal in just four years!



Then there's [Amber](#), which is Smalltalk for the web. It's also FREE and evolving quickly.

Despite [Smalltalk's reputation](#), this is not your father's Smalltalk. Modern Smalltalk is exciting and promising.

It's true that Smalltalk jobs are not (yet) plentiful. But if enough of you aggregate to a new wave of Smalltalkers, then the industry will adapt to it and we'll see wider adoption of Smalltalk in business. The question is, do you have the vision?

Share Improve this answer

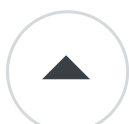
answered Nov 25, 2015 at 21:24

Follow



[Richard Eng](#)

2,024 ● 6 ● 25 ● 32



4

I was taught Smalltalk in one of the first graduate college level Object-Oriented Programming courses (circa 1988). The teacher thought it best to start was a "pure" OO



language, before moving on to a more trendy one (we did a bit of C++ at the end of the semester).



By that measure, it's still best to start with pure OO, although these days we have Java & C#, both of which are "nearly-pure" OO -- close enough that you can get by ignoring the non-OO features of them, and limiting yourself to the Pure-OO subset of the languages.

Share Improve this answer

answered Sep 29, 2008 at 15:05

Follow



James Curran

103k ● 37 ● 185 ● 262



If you want a better understanding of Extreme Programming (and even Scrum) I'd say **yes**.

4

Why impatient Java programmers need to learn Smalltalk:



<http://www.dafydd.net/archive/2010/why-smalltalk-isnt-just-another-language/>

Share Improve this answer

edited Feb 21, 2011 at 12:08

Follow

answered Aug 4, 2010 at 17:24



Dafydd Rees

6,983 ● 3 ● 43 ● 48



4



I've been a software engineer for quite a few years now. I've heard people bring up Smalltalk a few times, and certainly Smalltalk has been around since about 1980, but it's one of those languages that's never seemed to make it into the software mainstream. Sort of like Objective C, CLIPS, PL/I, etc--something you may have heard of, but something that most folks have never programmed in.

I probably wouldn't take the time to learn Smalltalk unless I needed to for a particular job. I looked at some Smalltalk tutorials and examples briefly a few years back, and it looks like it has some clear advantages for certain aspects of OO programming (like the message concept seems cool). But sadly, it is not mainstream, and doesn't seem to be gaining much momentum.

Share Improve this answer

edited Oct 1, 2012 at 0:29

Follow

answered Sep 29, 2008 at 15:27



[Ogre Psalm33](#)

21.9k ● 17 ● 77 ● 92

4 Objective-C ... dont think that the statement still stands as of 2012. :) – [sjas](#) Sep 27, 2012 at 21:27

1 @sjas How true! All Smalltalk needs now is for a major developer to create a new device niche market, and specify Smalltalk as the defacto language for writing apps for the device! – [Ogre Psalm33](#) Oct 1, 2012 at 0:28



3

This thread has become very actual for me. I'm planning for a Software migration to a web-application. It's a database based software. I'm especially checking the alternatives 1) Rails 2) Seaside



If I can get the figures for the Gemstone/S as Database, I'll consider that also. So for me it means I have to learn Smalltalk (better) than before. Because it could be that it will be my work for the next 15 years. You would (and should not) work with software you don't like for that long ;-). I've the impression Gemstone/S is one of the "killer" applications. But persistence of Objects still is a very difficult field....

Share Improve this answer

Follow

answered Dec 30, 2009 at 7:40



Friedrich

5,996 ● 1 ● 29 ● 46



3

1) Yes! It's always good to learn a language. If you are going to learn a language, make it a powerful, influential language that can be learnt easily and quickly.



Smalltalk remains a pre-eminent language and environment for learning OO concepts.

It is all objects, all the way down. This makes for a really consistent approach to working.

Integers are instances of Class Integer. Strings are a collection of character objects. Classes are singleton instance objects for the class they define.

Control structures work by sending get messages to instances of Class Boolean.

Even anonymous methods (blocks of code, aka blocks) are objects.

Everything is done by sending a message to an object. The syntax can be fitted on a postcard.

The clarity of the concepts and their implementation in Smalltalk mean that you can develop ways of thought which transfer directly into Java, Ruby and C#. I expect it's true for Python, too.

It's so good for making the concepts clear that a major UK University used Smalltalk to train 5,000 people a year in object-oriented computing.

Squeak 5, has just been released. It has gained major performance increases from its new Cog/Spur VM, which features with progressive garbage-collection.

Pharo 4 has a lovely clean-looking desktop theme. The next version, Pharo 5, will be released soon. It will move to using the Cog/Spur VM, it will have about 5,000 classes in the release, and additional packages of classes are readily available from the net via the Configuration Browser tool.

Squeak 5 is performant even on first-gen Raspberry Pis, and is almost 50% faster on the new \$5 Raspberry Pi zero. \$99 buys you a Raspberry Pi 2, screen and case - running a mature, fully feature-complete IDE.

Leading edge research is being done on co-ordinated, distributed OO systems in Smalltalk (e.g. Naiad and Spoon).

Some of the world's largest corporate databases are run on Smalltalk - including tracking of 60% of the world's shipping containers, and trading systems in the world's largest bank.

You can use Smalltalk as a sort of super-powered CoffeeScript, writing in [Amber Smalltalk](#) and transpiling to JavaScript, running in the browser.

Squeak, Pharo, and Amber are all Free, Open-source, open-licenced languages and environments.

Squeak and Pharo provide write-once, run anywhere facilities for MacOS, Windows and Linux. (Possibly RiscOS, too).

Dolphin Smalltalk is targetted firmly at native Windows look-and-feel, and lets you compile closed .exes of your finished work for distribution to end users. Further development of Dolphin by the vendor has stopped, but it is completely functional, and, like all Smalltalks, designed to be massively extensible. (Did I mention that Pharo now

has 5,000 classes, compared to Squeak's 3,000? Pharo is a fork of Squeak 3.9)

******There is a How-to guide for installing and starting Squeak, Amber, Pharo, Cuis and Dolphin at: ******

<http://beginningtosmalltalk.blogspot.co.uk/2015/11/how-to-get-smalltalk-up-and-running.html>

The Seaside web framework runs on Squeak and on Pharo. It's a wonderful mature tool, as is the more traditional AidaWeb framework.

VisualAge, VisualWorks and Gemstone all provide enterprise-grade robust systems. Gemstone provides an infinitely scalable object database with transactions and persistence.

2) Yes - I do already use it.

I learnt it via the Open University, and was immediately productive in Ruby (a copy of the Pickaxe book and the library reference by my side). It helped me enormously with Java, and with Xerox Moo-code.

I have just returned to it to write apps to control manage and distribute responsive, massively multi-platform mobile apps.

I expect that soon I'll be re-writing my JavaScript mobile apps using Amber, too.

Follow



Euan M

1,136 ● 11 ● 20



2



I don't really know what you're looking for.

If you are looking for a different language to write in, I'd think that would depend heavily on the libraries available.

I know neither Ruby nor Smalltalk, but it seems likely that the most efficient way to write Ruby on Rails-sorts of applications may not be Smalltalk.

If you are looking to learn the ideas behind Ruby, this might be a very good move. I don't have anything quantitative, but I always felt better about using tools (such as language systems) if I knew more than just the tools, if I knew the ideas behind them or how they worked.

If you want to learn different sorts of object-oriented languages, you might well want to learn Smalltalk (if it differs significantly from Ruby), something like Java or C++, and perhaps also the Common Lisp Object System.

If you just want to learn something different, Smalltalk may well be a good choice. I'd also suggest Common Lisp, and other people will doubtless have other suggestions (can you get a good Forth system nowadays?).

Share Improve this answer

answered Sep 29, 2008 at 15:31

Follow



David Thornley

1 It is clear you didn't take a look at Seaside. The most efficient way to write Ruby-on-Rails sort of applications is clearly with Seaside. – [Stephan Eggermont](#) Feb 10, 2009 at 12:18

1 Didn't take a look at Seaside, and haven't used Ruby or Smalltalk, so this is general advice. Thanks for pointing out Seaside. – [David Thornley](#) Feb 10, 2009 at 14:38

**-6**

Yes, I'm interested in it. Tried to start once already, but couldn't find a Smalltalk development environment that didn't cost both arms and a leg.



Share Improve this answer

Follow



answered Sep 29, 2008 at 15:08



[Brian Knoblauch](#)

21.3k ● 16 ● 61 ● 96

2 Squeak (from squeak.org) is a free open source implementation based on the original Smalltalk-80 image. – [Kevin Driedger](#) Sep 29, 2008 at 15:14

1 VisualWorks is also available as a freebit now. You can get it from cincomsmalltalk.com – [ConcernedOfTunbridgeWells](#) Oct 8, 2008 at 12:33
