STATE OF HASKELL 2018

AARON CONTORER

TABLE OF CONTENTS

HASKELL USER SURVEY RESULTS

The powerful Haskell community includes researchers who advance the language, and users who apply the language to solve real-world problems. Recently 1100 people took the time to respond to FP Complete's Haskell User Survey. Here are the key results.

EXECUTIVE SUMMARY

Haskell is thriving in a wide range of real-world uses, and growing rapidly. Nearly all survey respondents wish to continue or increase their level of Haskell use. Satisfaction is very high (13 to 1, or better) with Haskell's security, quality, reliability, maintainability, and advanced capabilities.

Where 30 months ago there was huge demand for better package management tools, this former roadblock has been cleared. Satisfaction is high (at least 6 times as many satisfied users, vs dissatisfied) with Haskell's tools, cost-effectiveness, rapid development, performance, compatibility, and deployability -- and with the supportive Haskell community.

Users now want to adopt more Haskell in their work, so understandably they desire more training resources, Haskell expert staff, professional services, and IDE support. They emphatically believe Haskell's reputation has not yet caught up with its reality.

"HASKELL **MAKES PROGRAMMING BEAUTIFUL AND PROFITABLE**. TEAR-FREE, COMMERCIAL, PRODUCT DEVELOPMENT: HASKELL MAKES THE DEVELOPMENT OF NEW FEATURES AS WELL AS MAINTENANCE MORE PREDICTABLE. HASKELL IS MATH FOR THE REST OF US."

- SVEN HEYLL

THE BIGGEST SURPRISES

These thousand-plus respondents form a picture of a Haskell community that's much more diverse and project-focused than Haskell's historic "scientists only" reputation.

HASKELL HAS TAKEN ROOT IN DIVERSE INDUSTRIES.

Teams use Haskell in many fields and solve many kinds of problems. Survey respondents report building serious work projects in financial analysis, cloud middleware, network security, transportation logistics, healthcare, life sciences, robotics, data repositories, language design, high-performance computing, Web services, OCR, text analysis and natural language processing, system management, e-commerce, high-frequency trading, cryptocurrency, astronomy, online advertising, insurance, hardware engineering, software

HASKELL USER SURVEY RESULTS

engineering tools, Internet of Things (IOT), education, XML and JSON and other data parsing and data format conversion, and of course computer science research.

HASKELL IS USED TO BUILD MAJOR PRODUCTS.

While many users select Haskell just for key problems in mixed-language teams, or just the server side of things, many others report building end-to-end solutions fully in Haskell, including back end, front end, microservices, data management, everything.

ENGINEERS ARE GETTIN' IT DONE.

In 2015 users complained that Haskell tooling was frustrating and it wasn't clear how to get productive, but this has improved hugely. Most applied users (80%) use the tool Stack which is new since that time, and others use the tool Cabal which has improved too. With these open-source solutions in hand, the language itself has proven to be much easier than people had worried.

"I LIKE HASKELL BECAUSE IT **OPENS EYES TO LESS KNOWN, ALMOST "ANCIENT" SOFTWARE ENGINEERING IDEAS**; PARTS OF WHICH BEING RECENTLY INCORPORATED TO THE "POPULAR" LANGUAGES IN THE MARKET. I DON'T USE IT FOR WORK, BUT PURE LANGUAGE PRACTICE ALLOWED ME TO FIND AND DO A BETTER JOB."

- BARTŁOMIEJ JERZY STOBIECKI

HASKELL DELIVERS THE QUALITY, FEATURES, AND PERFORMANCE FOR PRODUCTION WORK.

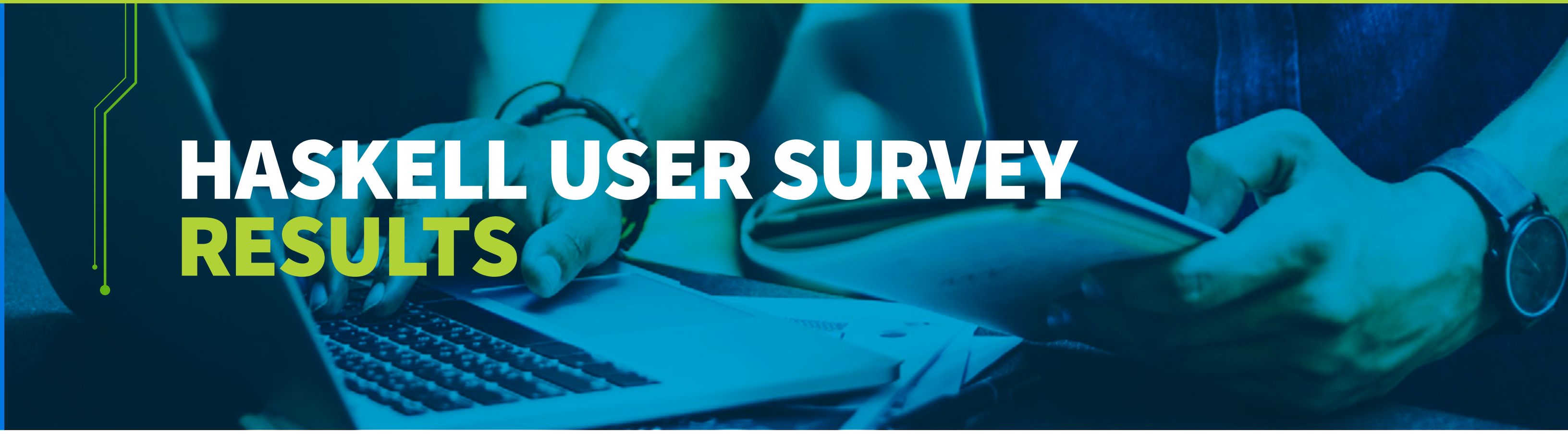
The survey focused intently on these issues, and users say in detail that they can build serious IT solutions in Haskell, and that it has the features they want. See the detailed "top strengths" below.

HASKELL'S REPUTATION, COMMUNICATION, AND EDUCATION NEED TO CATCH UP TO REALITY.

Severely trailing users' high satisfaction with the language, they are frustrated by a lack of materials to easily explain their enthusiasm to colleagues and management. Haskell's historic reputation as elite or hard to learn is a real problem for users, who believe better outreach material will increase adoption.

INTERMEDIATE USERS AND TEAMS NEED MORE ATTENTION FROM THE COMMUNITY.

While ratings for the Haskell community's inclusiveness remain stellar (19 times as many users feel included/ welcomed vs. the opposite), applied and non-advanced users express a strong desire to be embraced and helped along, with more encouragement and examples and mentoring, in less advanced terms they can easily follow.

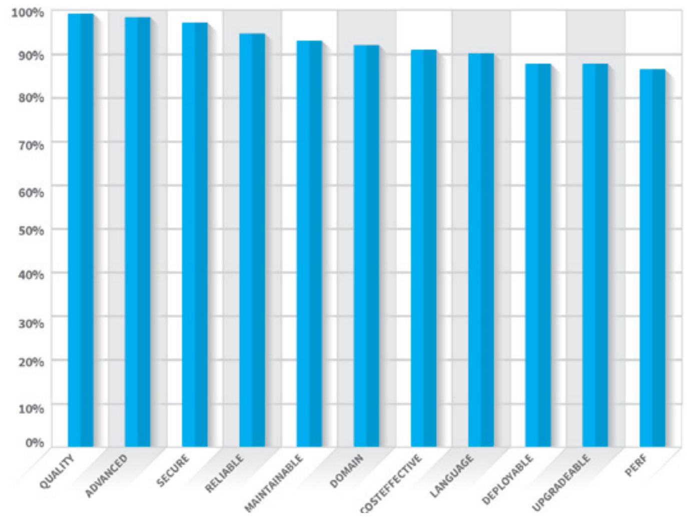


HASKELL USER SURVEY RESULTS

TOP STRENGTHS OF HASKELL

Which aspects of Haskell best satisfy users? These statements earned the highest ratio of positive (4 or 5, agree or strongly agree) to negative (2 or 1, disagree or strongly disagree). To help understand the ratings I've also included a few of the thousands of comments users gave.

TOP-RATED HASKELL BENEFITS
% OPINIONATED FAVORABLE OR STRONGLY FAVORABLE



118:1* (avg=4.37) Software developed using Haskell is ...high in quality (low in bugs or defects).

- Really impressive. Bugs were trivial.
- It is probably the most production-capable language I've used.

**Unless stated otherwise, these ratios depict the number of satisfied and highly satisfied respondents as compared to the dissatisfied and highly dissatisfied respondents. In this case, there are 118 satisfied and highly satisfied respondents for every single dissatisfied and highly dissatisfied respondent.*

> "THERE'S NOTHING BETTER THAN **A LANGUAGE THAT WORKS WITH YOU** TO ACHIEVE THE BEST RESULTS. HASKELL EXCELS AT THIS AND MAKES PROGRAMMING A JOY."
>
> *- FINTAN HALPENNY*

50:1 (avg=4.21) … advanced, innovative.

- I use Haskell mostly for compilers and other symbolic processing, which I feel it fits naturally at. Along with prototyping, and evaluators, and so on.
- We have everything implemented in Haskell build system, backend services and microservices, even web frontend
- We've implemented a medical data consumption system for a client, most of the data processing was made in Haskell, also, a real time catalog was implemented in the past with great success.
- Machine learning
- Phylogenetic analysis software
- Cryptocurrency

HASKELL USER SURVEY RESULTS

47:1 (avg=3.88) … secure.

- Picus is a successful cyber security product built with Haskell primarily.

22:1 (avg=3.95) … reliable, handles faults well.

- Haskell doesn't prevent you from writing terrible code, it just helps more than regular languages.
- I really would consider using Haskell if I work on a very critical system with quite a lot of developers involved.

> "I LIKE HASKELL AS ANOTHER PROGRAMMING LANGUAGE. HASKELL IS **USEFUL NOW AND HAS FUTURE**. I USE IT FOR MY HOBBY PROJECTS."
>
> *- ALIAKSANDR LIAVONAU*

16:1 (avg=4.04) … maintainable over time.

- The primary benefits for me are maintainability and type-driven design. Refactors are easy and safe.
- The main win for us is maintainability. We're able to refactor ruthlessly because we know things won't go pear-shaped.
- Refactorable, too. Agile projects easier to do with a language that has a low barrier to entry (and even encourages!) refactoring.

13:1 (avg=3.94) Using Haskell, I can readily deliver projects that support the kinds of algorithms, computations, analysis, and other processing I care about.

- I use haskell for most of my web scraping, data processing, and some analysis. I also use haskell for building static web applications
- Internal support tools for research and development e.g. schema diagram generator for MS SQL Server databases
- Our software product is written primarily in Haskell (~80%, ~300,000 loc)
- Rapid development of a prototype interpreter/ compiler suite.
- Our Haskell data modelling and data processing engine has been shipped for several years.
- cryptocurrency development tools; database library
- Authentication service, OCR, text analysis
- Front end, back end, reporting solution for insurance
- We have everything implemented in Haskell build system, backend services and microservices, even web frontend
- One robot is moving right now and painted on a graph using Qt.
- Cardano is a product now in production
- Implementation of railway signalling system.
- Genome analysis tools
- We developed a Haskell SDK for using AWS S3 object storage
- Our core optimization and simulation logic is all written in Haskell and currently deployed at a test set of stores and distribution centers

HASKELL USER SURVEY RESULTS

10:1 (avg=3.69) ... cost-effective.

- The power to weight ratio for Haskell is amazing and with a small team I think we were able to handle a lot with more confidence. That said getting experience takes a while and, I guess like with most things, it really helps if you can work with someone more experienced.

“HASKELL TAUGHT ME THAT PROGRAMMING USING MATHEMATICAL CONCEPTS IS **FUN AND POWERFUL**. I ALWAYS ADMIRED THE BEAUTY OF ABSTRACT ALGEBRA WHEN I WAS AN UNDERGRADUATE, NOW I CAN USE THEM TO PRODUCE BETTER, **ROBUST AND BEAUTIFUL CODE** WITH HASKELL. HASKELL LEVERAGED MY ABSTRACT THINKING TO ANOTHER LEVEL.”

- ALEXANDRE GARCIA DE OLIVEIRA

9:1 (avg=3.80) The Haskell language has the features we need for successful projects.

- Haskell is by far the best language I have used and I’ve been writing software since 1974.
- The elegance and expressiveness please the developer.

7:1 (avg=3.63) ... deployable onto the machines where it will run.

- I develop across Windows, FreeBSD and nix and Haskell just works in those environments.
- building deployable artifacts on one OS and deploying it on another; not so simple

7:1 (avg=3.53) I can easily incorporate new features or versions without breaking my existing work.

- As I learned I improved the code. Quality did not degrade and I could rely on my functional tests all though the process.

6:1 (avg=3.55) ... high performing (fast).

- Our whole API gateway is written in Haskell, which handles millions of requests constantly. We could not be more satisfied with it.
- Haskell has some great tools for writing fast code where necessary. However, reasoning about performance in Haskell is harder
- [We built] a solution for high-frequency trading
- Serving ads quickly in multiple geographic regions with a lot of concurrency logic and relatively low latency, doing some analytics with complicated structured event data
- All backend web services are implemented in Haskell
- I like compiled languages for ease of deployment and speed

HASKELL USER SURVEY RESULTS

6:1 (avg=3.62) The available Haskell tools make me more productive.

- Haskell by itself would allow for rapid development, but when starting greenfield projects, there can be a lack of knowledge concerning best practices to use, advantages/ disadvantages of libraries etc. Once the right choices regarding this issues are made, Haskell is enormously beneficial.
- Haskell needs a lot more vertical integrated tools.
- The Haskell build tools are fantastic (Stack esp.)
- Stack and cabal have made huge improvements to software
- Stack made a big difference. I can go back to old projects, or code in old blog posts and get them to work quickly.

> "LEARNING HASKELL HAS LITERALLY **BROADENED MY HORIZONS** AND BROUGHT NEW WAYS TO SOLVING PROBLEMS IN MY WORK."
>
> *- RAYMOND TAY*

THE HASKELL COMMUNITY CAN BE VERY SUPPORTIVE

The Haskell community itself earns high ratings, though newcomers and intermediate users sometimes feel intimidated or desire more materials suited to their level:

19:1 (avg=3.78) I feel included and welcome in the Haskell community

- Seems like everyone embraces the teaching and learning spirit
- The coolest most forward thinking nerds ever
- I do, but I'm a cis white male

17:1 (avg=3.73) The Haskell community treats people the way we should be treated.

- The haskell community generally sticks to technical issues, and I am happy with it.
- There's a tendency to perpetuate the myth that Haskell can only be used by experts or mathematicians, when in fact the invariants provided by advanced languages accrue to the non-experts and beginners.

15:1 (avg=3.82) I have a positive overall experience when seeking help from the Haskell community.

- I've learned so much from the community, everyone is so intelligent and helpful.
- Even those that ought to be the busiest and most distracted by their own focus managed to find the time to answer questions often with amazing clarity and detail.
- Some tend to be very practical and understand the needs of someone coming from non-functional (or functional but not pure) backgrounds and answer appropriately. Others tend to be more academic

which can be off-putting, especially for such a demographic

8:1 (avg=3.70) There are clear opportunities to participate and give back to the Haskell community if I want to.

- Opportunities, yes. Clear ones, not sure. Contributing to GHC itself seems hard, but honestly haven’t looked into it. I certainly file issues on other Haskell libraries and attempt to help like that.
- It’s difficult to determine what “central” projects the community should focus on and how to help with those.

7:1 (avg=3.59) I easily get sufficient help and advice from the Haskell community.

- One strength of the community is how friendly and smart they are.
- Online tutorials, technical blogs are quite common and of really good quality. There are a lot of nice resources for beginners but less “practical” intermediate content. A good Haskell book targeting intermediate Haskellers will be welcomed.
- Haskell community is smart and passionate, and really helpful. A huge plus!!

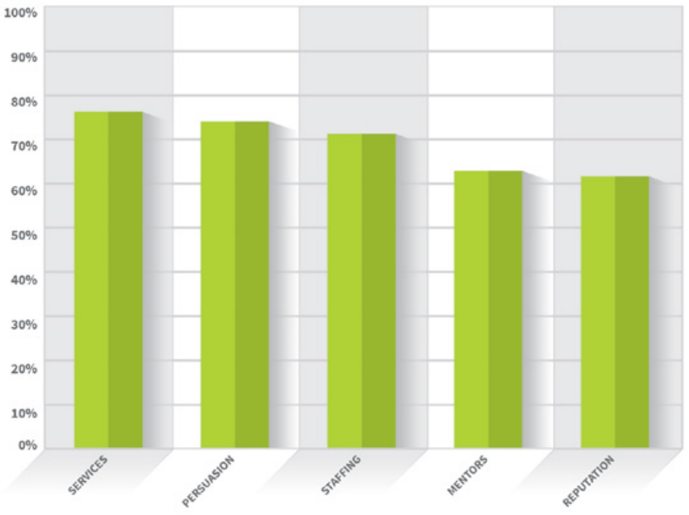
There is a minority of users who feel the community can be elitist or sexist. I believe our community needs to get an A+ in this topic, and I will make a special blog post about these comments received. Let’s see how we can do better.

THE NEED FOR SERVICES, PUBLICITY, AND TRAINING

These 3 topics all showed up on the 2015 survey, so we asked about them again in various ways. Now that technical things are going so well, these remain as the trouble spots for users. Interestingly, none are really technical issues.

TOP-RATED HASKELL BENEFITS

% OPINIONATED UNFAVORABLE OR STRONGLY UNFAVORABLE



Here’s the ratio of disagreeing (rating = 1 or 2) vs. agreeing (rating = 4 or 5), along with the the average rating, where 3 = neutral. I’ve included representative comments from the very many provided.

3:1 (avg=2.46) My organization and I have the professional services, support, mentoring, or training needed for efficient Haskell adoption.

- Missing the “lower end” staff training stuff
- We do it all on a really adhoc basis, without formal resources other than stuff like haskellbook and typeclassopedia
- Consulting, code reviews and overall support might be necessary in the future
- We don’t and I don’t know how we would go about it

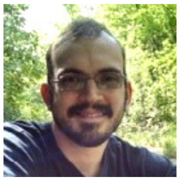
3:1 (avg=2.42) It is easy to convince my management, and any other approvers I care about, that using Haskell merits their approval and support.

- Sadly, Haskell is still perceived as having a steep learning curve
- My management doesn’t care which tools or languages I use, they do care about having the job done, and I like it this way.
- The difficulty lies in selling the necessary upskilling of engineers not familiar with Haskell (or FP in general).
- This wasn’t the case a year ago. In recent months, some of the approvers are swayed by the languages features and elegance.
- I am the management! We’ve even convinced clients!
- This is heavily reliant on my personal gravitas, such as it is.
- Haskell’s long age has given it a reputation that no longer holds true. Fortunately, there is a tremendous amount of new evidence to help demonstrate the benefits of Haskell.

2.5:1 (avg=2.53) I believe my organization can easily locate and hire, and/or train in-house, all of the skilled staff needed to adopt Haskell.

- Training smart engineers in Haskell has been a doddle. We had most of the team productive within a day or two.
- We haven’t tried.
- There is really not enough Haskellers in my area.
- We’ve been able to find people when we are hiring, but it usually takes a while. We do have a pretty specific house style of Haskell that’s easy to teach given some basic familiarity.
- We don’t have the staff to do it inhouse, and Haskell is not on that many resumes.
- I’m figuring this out again, if one accepts remote developers it is easy.
- Code often suffers from poor, and nonstandard documentation it will claim to do something, but short of reimplementing it is impossible to understand why or how.

HASKELL USER SURVEY RESULTS



> “HASKELL TAUGHT ME HOW TO MODEL PROBLEMS AS LOGICAL CONSTRUCTS AND ALLOWS ME TO SOLVE THOSE PROBLEMS WITH **CLEAR, SYSTEMATIC DEDUCTION**. COMING FROM OTHER LANGUAGES, I FEEL TWICE AS PRODUCTIVE IN HASKELL BECAUSE THE COMPILER IS LIKE A PERSONAL ASSISTANT THAT SPOTS YOUR MISTAKES!”
>
> *- ELLIOT CAMERON*

1.7:1 (avg=2.70) My colleagues and I have sufficient access to Haskell experts or mentors who can help us find the best ways to get things done.

- No one at my workplace knows Haskell better than me. And I am myself a beginner
- We don’t have access to experts or mentors but we are all engineers who understand it.
- Great community online, know almost no one offline
- We just don’t have access to these professionals
- I’ve had to mostly forge my own path
- I have gone to many conferences (academic and industrial) to make personal connections with experts
- No FP experts in my organisation
- Haskell by itself would allow for rapid development, but when starting greenfield projects, there can be a lack of knowledge concerning best practices to use, advantages/disadvantages of libraries etc. Once the right choices regarding this issues are made, Haskell is enormously beneficial.

1.6:1 (avg=2.74) I believe I could easily onboard new engineers on my team to my Haskell projects.

- People are actually curious to learn new things at the core, but everyone feels pressure to get things done so few would want to take the risk vs staying with the thing they already know
- Haskell propaganda has an odor of Academia, and not enough corporate bravado, to convince people that there are serious business interests backing the ecosystem
- Sometimes the engineers refuse to learn new tools, and I cannot do anything
- There is no agreed upon collection of snippets that would illustrate to the novice how to accomplish common tasks
- Where do you find these people?
- I feel confident in my ability to teach Haskell to new people. However I also think that Haskell is a lot to take (FP, purity, laziness) in all at once.
- Onboarding is rough. the standard prelude should actually include the major packages that developers should use telling people to override to use data. text instead of string is wholly inadequate. Improve the out-of-box experience.
- I am not in a position to convince my superiors of switching to new language/paradigm. Too many unknowns for management to justify it before investors.
- Due to library documentation quality, availability

of examples to work with, i'm still at the point where haskell development is not what i would call "rapid".

1.6:1 (avg=2.77) Haskell's reputation helps efforts to get it adopted.

- It's reputation as a powerful tool for superbrains can be a liability
- On one hand reputed to be hard to learn. on the other known for being extremely solid. cuts both ways
- Haskell has a very good academic reputation, but people are generally not convinced that it can be used for real world usage.
- Most of the people think it is black magic and stay away from it
- Yes and no. No, because some people are terrified by unknown.
- It's theoretically intimidating and there's a well known dearth of quality "tutorials", such as you would get in any of the widely used languages.

(I would add anecdotally that the financial technology industry seems to be the exception -- I've met so many FinTech people whose impression of Haskell is outstanding, even if they haven't tried it yet.) It seems we have our work cut out for us -- actual Haskell users are so satisfied, but constantly encounter non-users who believe it would be hard to learn.

> "I LIKE HASKELL BECAUSE IT IS MATHEMATICAL AND MATH IS THE MOST CLEAR HUMAN LANGUAGE. I USE IT BECAUSE OF THE **POWER OF COMPOSITION AND PURITY**. HASKELL MADE ME BE INTERESTED IN CATEGORY THEORY. I SEARCH FOR STRUCTURE IN THE ABSTRACT...I SEARCH FOR DEFINITIONS. ROADMAPS NOT ONLY FOR COMMUNICATING WITH THE MACHINE. BUT ROADMAPS FOR PROPER INTERACTION WITH EVERYONE AND EVERYTHING."
>
> *- AHILLEAS BALAKTSIS*

Clearly users want better resources to train their colleagues on Haskell, and to teach their management about the benefits that users are already seeing. And clearly there is a need for senior Haskellers to provide commercial-quality services that help get teams up to speed. (This is consistent with our anecdotal experience at FP Complete serving numerous companies that use Haskell -- they value such services, and typically want to reuse known best practices -- if someone will share them.)

HASKELL USER SURVEY RESULTS

WHERE IS HASKELL USED MOST?

Commercial use of Haskell has increased dramatically and diversified hugely since our previous survey in 2015.

At FP Complete we regularly serve Haskell users in a wide range of industries, but the biggest are financial technology (FinTech), healthcare/life sciences, security-focused Internet services, and transport/aerospace. These and many more were represented in the survey results.

Haskell-at-work users who answered the survey were also led by a large contingent from FinTech, as well as academia (Haskell being a favorite with researchers); then quite a few from network security, e-commerce, and networking in general. We see transport/logistics, diverse fields of manufacturing, and data processing infrastructure such as data mining and document management and cloud storage.

Most surprisingly, we also see Haskell users in truly diverse industries not historically considered Haskell zealots, including gaming, electronics design, and physical sciences HPC (supercomputing/cluster computing). In the past two years Haskell has really broken out of its imaginary "box" and is being used in a wide range of fields. We even see quite a few independent IT consulting/contracting firms using Haskell to create solutions for other companies -- Haskell isn't just for in-house use anymore.

The biggest lesson for me? Haskell is affecting the world at large. While Haskell continues to be an outstanding research platform -- and may the Haskell community keep up this momentum for a long time to come -- it is now firmly established as a commercial tool for use in a wide range of industries. Given that the top strengths from the survey were "quality, advanced, secure, reliable, maintainable," with a side dish of "cost effective, productive, high-performance," this increased popularity makes sense.

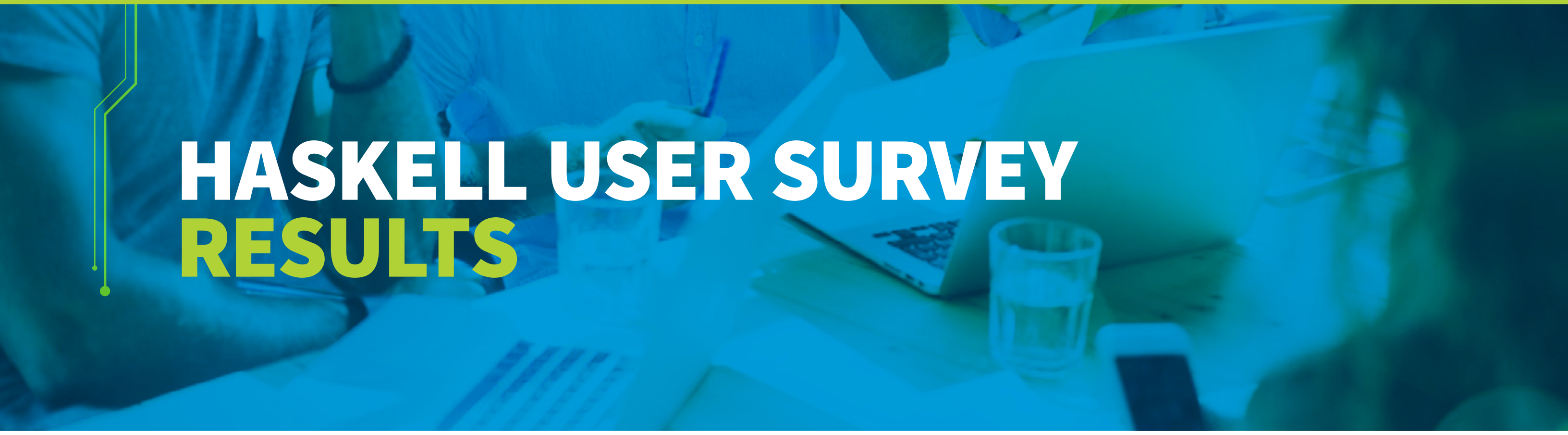
The second biggest lesson for me? Haskell is seeing particular success in financial technology. FinTech is one of the most demanding industries for powerful IT, so this validation is gratifying.

"AT OTAS WE'VE USED HASKELL WITH GREAT SUCCESS, IT'S A LANGUAGE THAT MAKES **BUILDING GREAT SOFTWARE EASIER, FASTER, AND MORE RELIABLE**."

- TOM DORIS PH.D., CEO, OTAS TECHNOLOGIES

SUGGESTIONS FOR COMMUNITY ACTION

We can and should celebrate the huge strides Haskell has made in applied adoption. After many years of academic-style development (which continues), a whole lot of applied Haskell users are now deep into



successful production with this language and toolset.

We are now a community of fairly advanced to very advanced and successful software developers and computer scientists. We can see how well our technology is working for friends and strangers alike. The challenge before us is to make this powerhouse technology more accessible to the more numerous masses of beginning and intermediate adopters.

In addition to the well-understood areas for standard tool development (IDE support, compiler speed, performance profiling, etc.), and the newer DevOps areas that have gained such importance in recent years (continuous integration, continuous deployment, cloud support, cross-platform integration, etc.), we now face non-technical work items that require some community focus:

- Improved learning paths for developers switching to Haskell (or even just trying Haskell out) from other language backgrounds.
- Community support and pro services for teams that have made the jump into Haskell and now need to ensure projects success -- not about theoretical discussions, but about “how do I solve my practical problem right now, with speed and quality, and impress the boss and colleagues.”
- Continued publication of success stories to help Haskellers overcome the language’s outdated reputation as suitable for only the most senior experts
- Improvements to community moderation to ensure that newbies asking the same old question for the hundredth time, or the weirdest question for the first time, are treated in the welcoming style that makes us all proud to be Haskellers

SUGGESTIONS FOR USER ACTION

If you’ve been looking at Haskell for a while, the enormous success of the language during the past 2.5 years suggests that it may be time to move forward. The variety of users reporting success with Haskell in a wide range of applications and industries tells us that the tools have firmly matured to production quality.

If you’ve had successes with Haskell (perhaps you’re even one of the survey respondents), consider writing up your experience report in some detail so that other would-be Haskellers can learn from you. If you don’t have a good place to post it, I’ll help you find one. (Our own blog isn’t full yet!) Especially consider publishing on sites (or linking from sites) that focus or your industry, not only sharing with current Haskellers but also with your colleagues in FinTech, in genetic research, in molecular dynamics, in network security, in cloud DevOps, in data modeling, in transportation logistics, or whatever field you’re working in. People need to hear about Haskell, and it’s ready to be talked about.

HASKELL USER SURVEY RESULTS

> **“HASKELL FITS THE WAY I THINK.** I STUDIED CATEGORY THEORY AND TRIED TO IMPLEMENT FUNCTIONAL PROGRAMMING IN PYTHON THEN DECIDED TO WHY NOT DIRECTLY PROGRAM IN THE ORIGINALLY FUNCTIONAL PROGRAMMING LANGUAGE.”
>
> *- ANDREW STARODUBTSEV*

If you need more active help moving forward with Haskell, don’t hesitate to be vocal online about your needs. If you need *really* active hands-on help, you are of course always welcome to contact FP Complete since we do that for a living.

If you’re one of the very many educators using Haskell, consider accelerating your teaching of Haskell and Functional Programming in general to your undergraduates. Industry is clearly ready to move past the era of only-imperative programming, and we need qualified grads.

If you’re an advanced engineer looking to contribute to hardcore Haskell tools or packages, and don’t know where to turn, please show up on the various Haskell online forums and ask “how can I help with x.” If you don’t know whom else to ask, you are welcome to ask me and I’ll refer you to someone helpful. If you’re a lead-level organizer of projects, so much the better.

WHAT ELSE DID FP COMPLETE LEARN?

We learned again that Haskell users are generous with their time; it is no small thing for a thousand software experts to complete detailed a 50-question survey, and hundreds made extra time to provide thousands of informative comments. Thank you, Haskellers!

We learned that our thousands of hours contributing to open-source components have not been wasted, and noted with special pleasure that 80% of commercial Haskell users are using Stack, an open-source tool initiated and lead-maintained by engineers from FP Complete. It was heartwarming to read the many comments of thanks and support directed at FP Complete regarding numerous projects we’ve done for the community.

As the world’s leading provider of Haskell tools and services, FP Complete is committed to contributing more than its fair share to the community. These encouraging survey results just reinforce our commitment. We paid for the survey ourselves and did the design, distribution, and data analysis, and are happy to share the results with the community.

More blog posts are forthcoming, with analysis of more specific topics in the survey results, and more great quotes from the thousands of lines of comments your fellow Haskellers kindly wrote.

HASKELL USER SURVEY RESULTS

Based on the survey results we will continue and even enhance our commitments to Stack and to Haskell training, technical staffing, and professional services.

FOR FURTHER READING

The old 2015 Haskell User Survey -- see how far we've come [https://www.fpcomplete.com/blog/2015/05/thousand-user-haskell-survey]

Get started with Haskell [https://www.fpcomplete.com/introduction-to-haskell]

Our Haskell engineering assistance and consulting [https://www.fpcomplete.com/haskell-functional-programming-cloud-devops]