The Future of GWT Report

Statistics of today and desires of tomorrow

2013

The Future of GWT looks bright

When we ran the Future of GWT survey for the first time in 2012, the amount of responses overwhelmed us. We titled the report “The Future of GWT” because we wanted to know if there was any truth in rumors that GWT was slowing down. With over 1300 responses it showed us and the world that GWT is very much alive and kicking. There were huge applications being built with GWT in very large corporations, and investments in GWT were growing.

With the Future of GWT 2013, however, we wanted to know more about how GWT should be developed in the future (Section 6), what Java and GWT versions GWT teams are using, to understand what needs to be supported (Section 4) and the extensions you need and other frameworks you’re looking at (Section 5). To get some background and demographical data we also asked about your team (Section 1), the apps you’re writing (Section 2) and how you’re going about writing them (Section 3). With this information, gathered in this survey, we feel that this is the most comprehensive study conducted on GWT thus far.

In 2013 we knew that GWT had picked up even more pace, with a dedicated conference in December (gwtcreate.com), exciting plans for GWT 3.0 and large companies investing in GWT as their technology for the future. In 2013 the Steering Committee also decided upon a tick-tock release cycle for GWT, with one major and one minor release per year.

This survey is the result of the work of Vaadin, Ray Cromwell (Google representative and acting GWT Steering Committee Chair), Daniel Kurka (Google, developer of mGWT and GWT-phonegap), Artur Signell (Vaadin representative), Bhaskar Janakiraman (Google), Colin Alworth (Sencha), Christian Goudreau (Arcbees), Konstantin Solomatov (Jetbrains), Mike Brock (RedHat), Stephen Haberman (Bizo), Joonas Lehtinen (Vaadin) and Thomas Broyer. You’ll see their comments and reactions throughout this report.

This survey was published at GWT.create 2013, with over 600 eager GWT developers attending the largest GWT event of the year. We’d like to thank all of the over 1400 respondents for their time and honesty, and look forward to your comments!

Fredrik Rönnlund,
VP of Marketing & GWT fanboy, Vaadin

Contents

Section 1:

Learn about GWT teams

1.1 Team size

The average team size for teams doing GWT development is 12,5 people whereas the median sets at 8 members per team. The fact that the team sizes have stayed the same from last years' validates the surveys credibility. The amount of front-end developers per team has its median at 2 developers, meaning that the UI layer roughly swallows 25% of the development team.

We also asked about how the teams have changed in the past 12 months to which 87% replied that they have either stayed the same or grown.

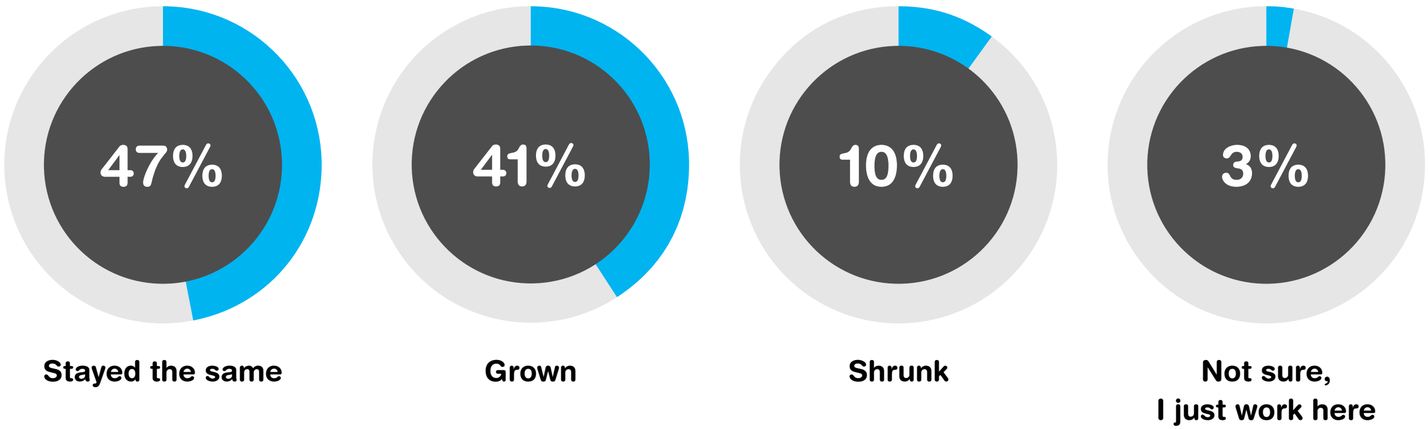
"While we do not have statistics, it might be a good assumption that people are likely to only know the total size of the team exactly and classify the rough division of time inside the project by the main roles of each person. Thus it is likely that many more than 2 persons from the team are coding with GWT."

Joonas

"This conforms with my experience of seeing GWT teams with 5 to 10 developers on average"

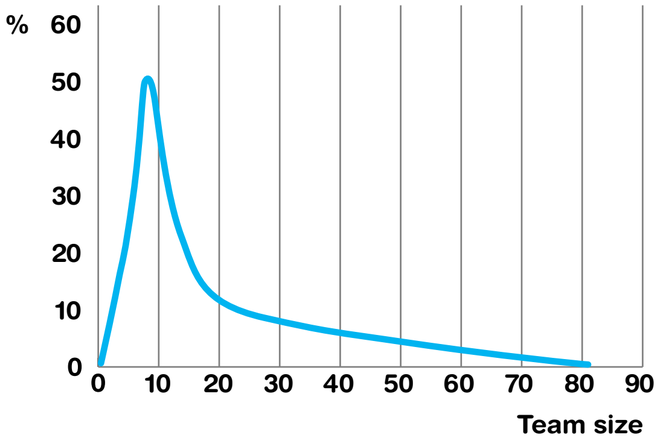
Christian

Team change in past 12 months



1.1 Team size

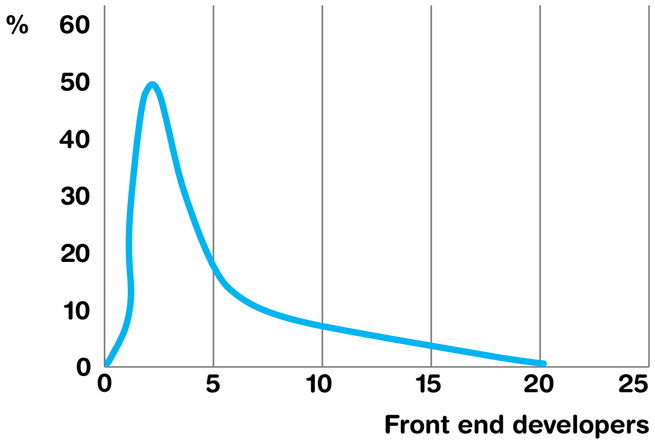
Team size distribution



“It is natural that GWT is used by fairly large teams, because the benefits of modularization GWT provides are most important for large projects.”

Joonas

Front end developers distribution



“Giving the economical context in North America, consulting businesses in general has gone down. At the same time in my experience, GWT project team sizes have either stayed the same size or grown in the past 12 months.”

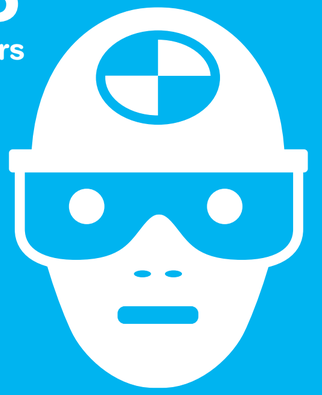
Christian

1.2 The average GWT team composition

The average team on a GWT project is made up of just over 8 people. This is a combination of Back End Developers, Front End Developers, Designers, Testers, Project/Product Managers, and a few others. The project managers still have a bit of distance to go before they overtake the amount developers.

2.7
Back End Developers

1.3
Testers

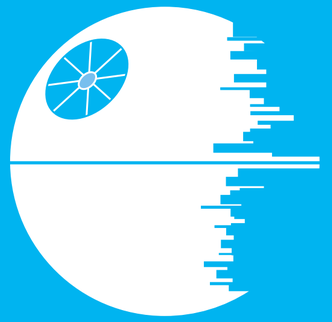


0.6
Designers

2.0
Front End Developers

0.9
Project / Product Managers

0.6
Other



1.3 Geographical division

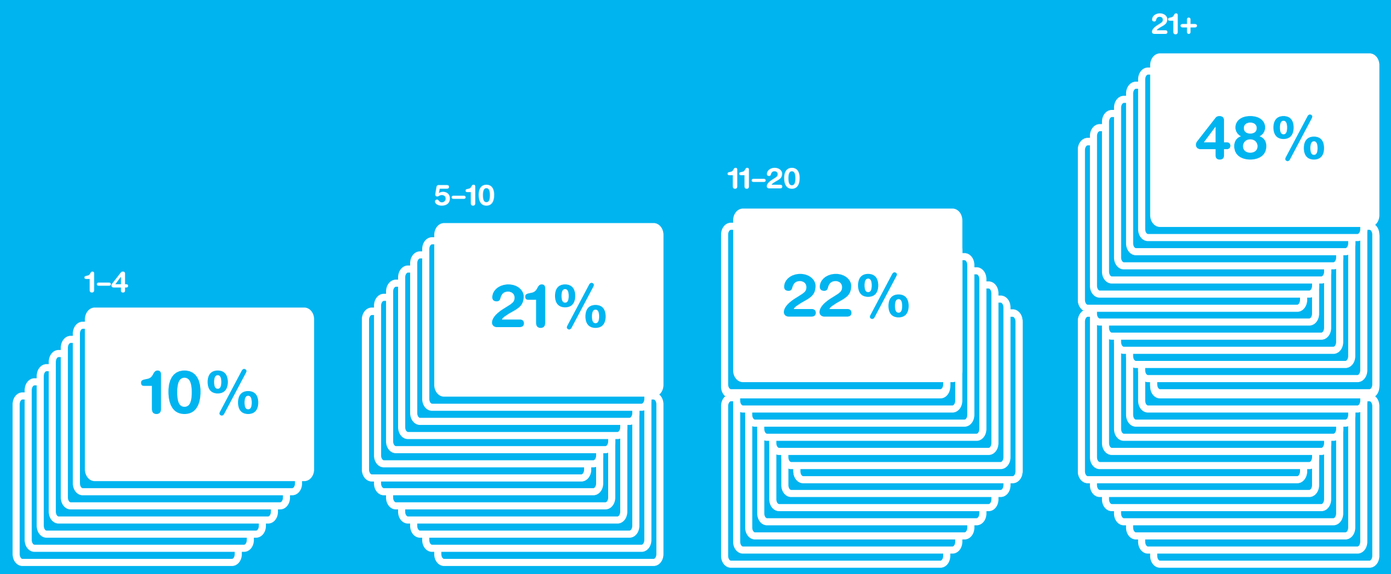
Europe still covers the lion's share of GWT respondents (58% of all respondents, up 8% from last year) with North America as number 2. The rest is shared between Asia, South America, Australia and Africa. GWT's strong hold still seems to be Europe and North America.

- **58% Europe**
- **25% North America**
- **8% Asia**
- **4% South America**
- **2% Australia**
- **2% Africa**

Geographic distribution of survey respondents

Section 2:

What kind of apps is GWT used for?



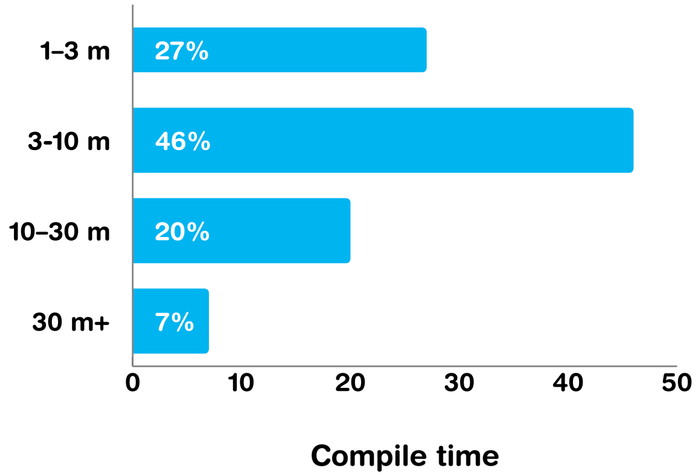
2.1 Application size matters

20 screens is the magic number for application size for GWT apps. 50% of all apps have more than 20 screens and 50% have less than 20 screens. We didn't ask about the size of the largest apps, but it would be interesting to see the size of the largest apps written in GWT.

> *"It would be interesting to know why GWT was chosen for apps with less than 5,6,7 screens, as these are quite small apps."*
>
> **Christian**

2.2. Application compile time

The size of the application is directly proportional to the time it takes to compile the application. Makes sense right? The largest part of all GWT apps take 3 to 10 minutes on average to compile (46% in 2013 compared to 48% in 2012). Also the low-end seems to be shrinking, most probably because people are writing larger applications. Last year 31% of the apps were in the quickest category which this year only gets 27% of all apps.



"GWT was designed to be a monolithic compiler, changing this is very hard. Fortunately, we are close to getting modular compilation working for GWT, which should result in dramatic speed up in compilation time some time next year."

Bhaskar

"Improving compile time has been our biggest task, but it will take time. A new incremental / modular compilation system is coming that should help a lot, but certain features in GWT like global code generators will make it less effective than it could be, until the generator system itself is improved."

Ray

"We are working on a new feature called separate compilation that will take out the pain of compiling a GWT app and will yield the same edit / refresh cycle you are currently getting with JavaScript."

Daniel

"I feel a bit sad looking at this. 30 minutes or more to compile an application is really too much."

Artur

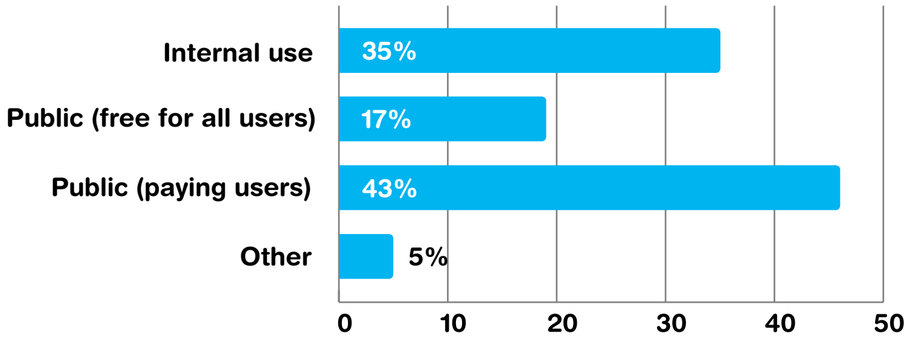
"Designing the whole app as several small modules that can run individually and that are built on top of a common framework makes it easy to have compilation times that are around 2 minutes even in applications that have several hundreds of thousands of lines of code. After that point only comes compilation time optimization."

Christian

2.3 Whom are GWT apps written for?

As might be expected, GWT users are much more likely to be building business applications (79%) over portlets or games, but content-rich websites also make up for a significant portion of usage (13%). Compared to last year's results, we now also have the spread between business internal and business external applications, which shows that GWT used in the business world still is more popular for internal applications (46% vs 33%).

What you might not have expected, is that more applications are being built for public use (65%), rather than for internal usage (35%). Even the publicly used applications are very business oriented, 71% of them are for paying customers.



Who are the end-users of the application your building?

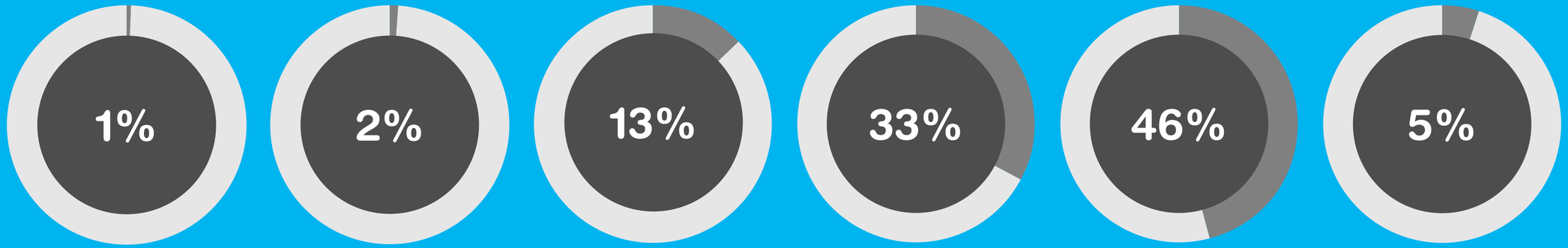
"78% of GWT apps are built for paying customers or internal usage. This means that there is a healthy ecosystem since many businesses and companies are invested in GWT."

Daniel

"GWT continues to be business critical for many users. As GWT Contributors, it's our responsibility to ensure we continue to make GWT better."

Bhaskar

How is GWT used today?



"Tons of huge business applications have been built with GWT. They represent such an investment that the option of not having GWT around just does not exist."

Joonas

"It's nice to see that there actually is a usage for games with GWT. One explanation could be that for Java programmers, GWT allow sharing code between the Web and Android, and libraries like PlayN extend that to iOS as well."

Ray

2.4 Browser support in 2014

IE 6/7

14% / 9%

IE 8

54% / 44%

IE 9

79% / 66%

IE 10

80% / 76%

IE 11

- / 70%

Safari

18% / 62%

Chrome

94% / 93%

Firefox

94% / 92%

iOS

- / 49%

Android

- / 50%

> *"The feedback I've seen on the field is that IE8 is slowly being replaced by IE9 and 10 with most clients that we've been working with (Banking and Healthcare)"*
>
> **Christian**

> *"I guess we can conclude that IE6/7 are finally dead enough to be dropped from GWT 3.0 entirely. It is sad that IE8 and IE9 are still widely required in 2014 as they are painful to support."*
>
> **Joonas**

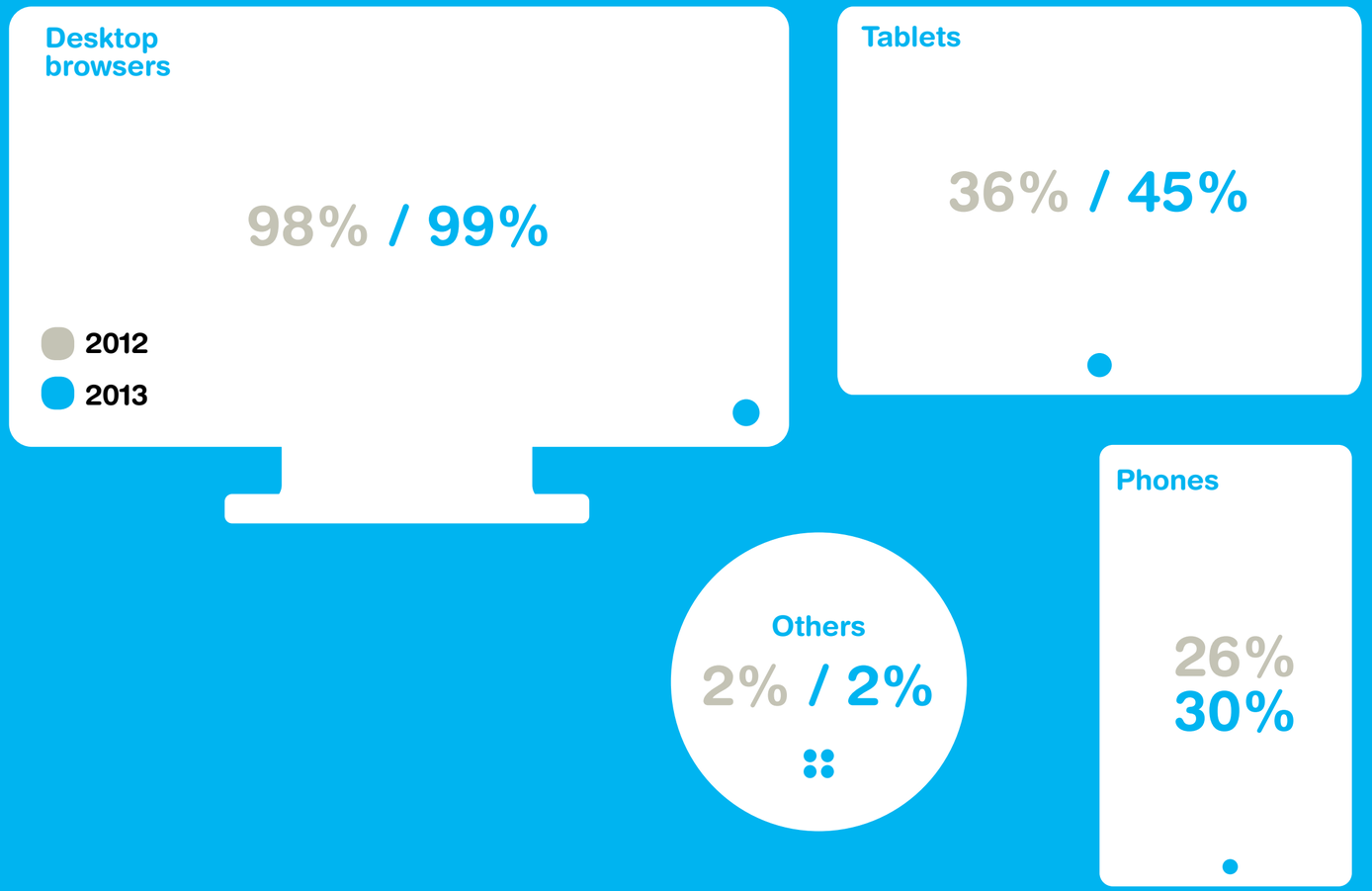
> *"The continued installed base of IE6/7/8 represents a regrettable drag on improving GWT's core, as there are many APIs we could improve if we did not have to deal with older browsers. The Web is in transition to Mobile and HTML5, and the inability to migrate off of legacy browsers makes it more difficult to navigate this transition."*
>
> **Ray**

> *"IE6 and IE7 still represent 9%. This, combined with the 43% of IE8, makes me sad; but I'm not surprised given the large 'Enterprise' user base."*
>
> **Thomas**

Browsers developers expected to support in 2013 **Browsers developers expect to support in 2014**

2.5 The mobile revolution, lead by tablets

Last year we asked about which platforms you need to support and 98% answered Desktop with Tablets at 36% and Phones at 26%. Desktop has stayed strong this year as well at 99% but both Tablet and Phone have gained speed. Almost half (45%) of the apps written in GWT need to support tablets already today.



"I am very pleased to see that the importance of tablets and phones is rising, because GWT can do an excellent job here."

Daniel

"The Mobile Web and GWT represents a way to develop mobile applications for Windows Phone/Surface, Android, and iOS without tripling development effort across three different programming ecosystems."

Ray

"GWT gives an amazing foundation for building mobile apps because it excels in optimizing for performance needed in mobile even more than on desktop and because it smoothens out many of the painful browser differences that would have to be dealt directly otherwise."

Joonas

"The increasing demand for tablet and phone support makes it important for GWT to bring touch gestures, momentum scrolling, and look and feel to existing applications."

Colin

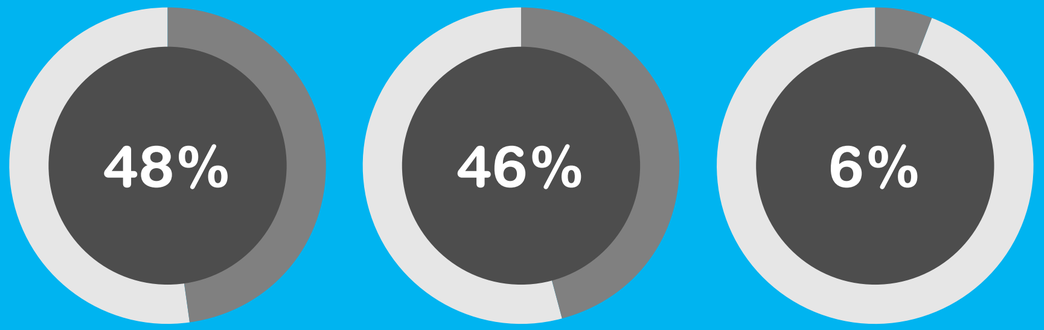
What kind of devices does your app support?

Section 3:

How to build an App with GWT?

3.1 Java, XML or Designer?

Coding Java by hand is still the most popular way of building a GWT application's UI. UI Binder is however as a strong number two right behind it. GWT Designer is losing ground even compared to last year and its share is quite marginal compared to the other two.



How UIs are built

"I don't know why people are doing their UIs by hand nowadays considering that every language has their declarative counterpart for the UI. ActionScript has MXML, Javascript has HTML and .NET has XAML"

Christian

"I am a little bit sad to see so many people writing their UI in Java code, when UiBinder is a better solution most of the time. We need to promote UiBinder better."

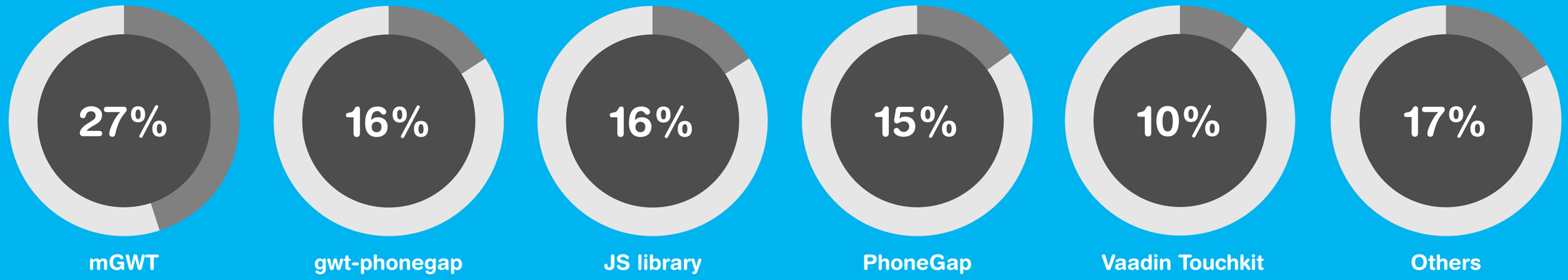
Daniel

"I, for one, am surprised UiBinder isn't used more."

Thomas

3.2 Mobile App Development

While you can develop mobile applications without any additional technologies in GWT, there are many frameworks around to help you. mGWT and PhoneGap are much more widely used than any others.



Mobile technologies

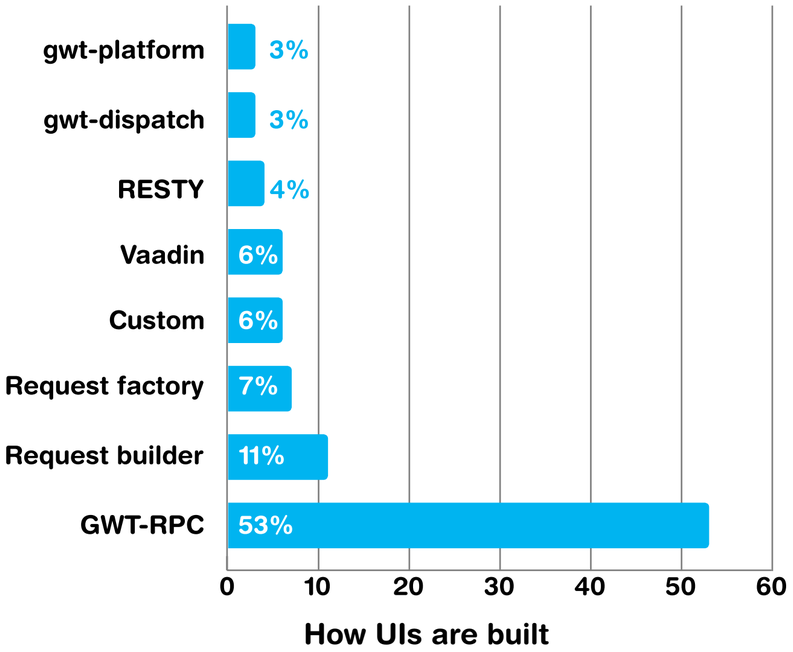
"Combining mgwt, gwt-phonegap and phonegap we can see that more than 50% use those to write mobile apps. I am very flattered by this and I am looking forward to make GWT an even better platform for writing mobile apps."

Daniel

"Great to see quite a few Vaadin TouchKit users. 10% is a fair share as the tool is commercial and specific to Vaadin."

Joonas

3.3 Backend communication



"GWT-RPC delivers the biggest bang for the buck in terms of speed of development when the app is small, but it has costs in compile time and interoperability over the long term. JAXRS is a candidate for GWT 3.0 inclusion that may be a good middle ground."

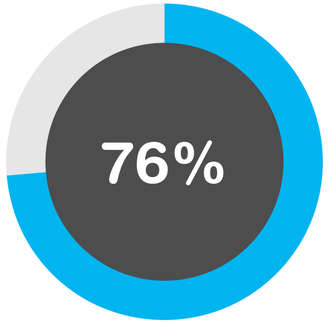
Ray

"When deciding on a new RPC system for GWT 3.0+ we should keep in mind that a lot of people like the simplicity and usability of GWT RPC."

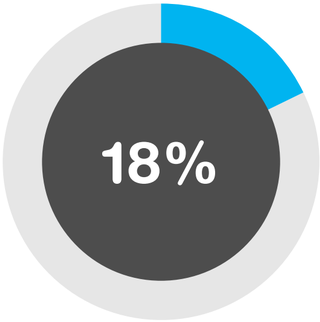
Daniel

3.4 Your IDE of choice

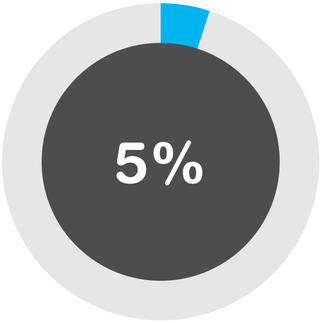
Eclipse is by far the most used IDE even in GWT-land.



Eclipse



IDEA



Netbeans

"I have a feeling the reason for Eclipse's popularity is the maturity of the plugin."

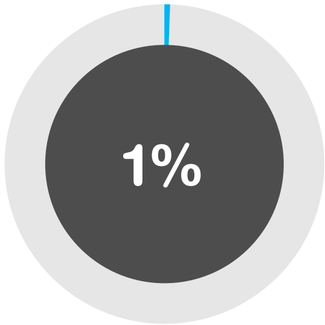
Artur

"I personally like IDEA's GWT plugin as it's more mature than anything else."

Christian

"I'm an IDEA fanboy, one of the things it does well is JSNI/Javascript code editing."

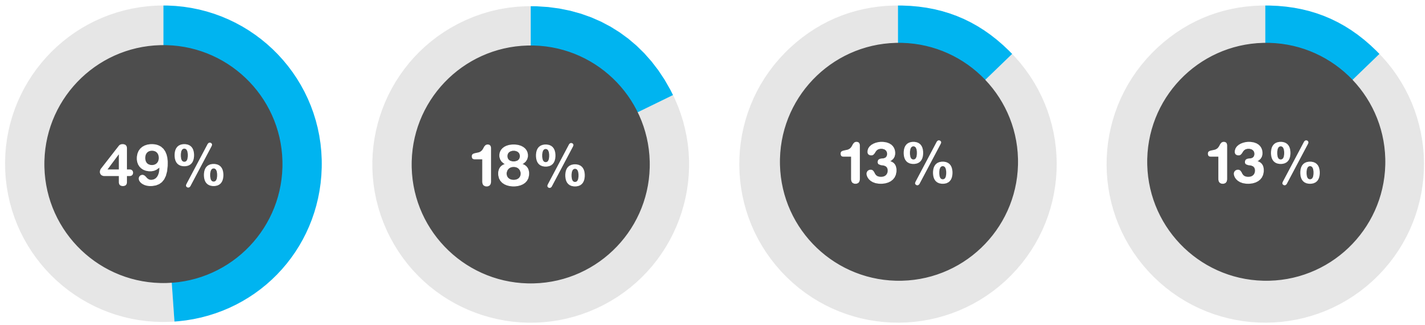
Ray



Other

3.5 How to test GWT apps?

Testing is handled exactly in the same way as last year - with still an alarming over 10% doing no testing at all and manual testing taking the lion’s share of all.



“Testing and automation should clearly be an area of focus for the community. We need a drastic reduction of the amount of manual testing.”

Bhaskar

“This pinpoints that writing tests is too cumbersome to be taken as a part of your standard development model in GWT.”

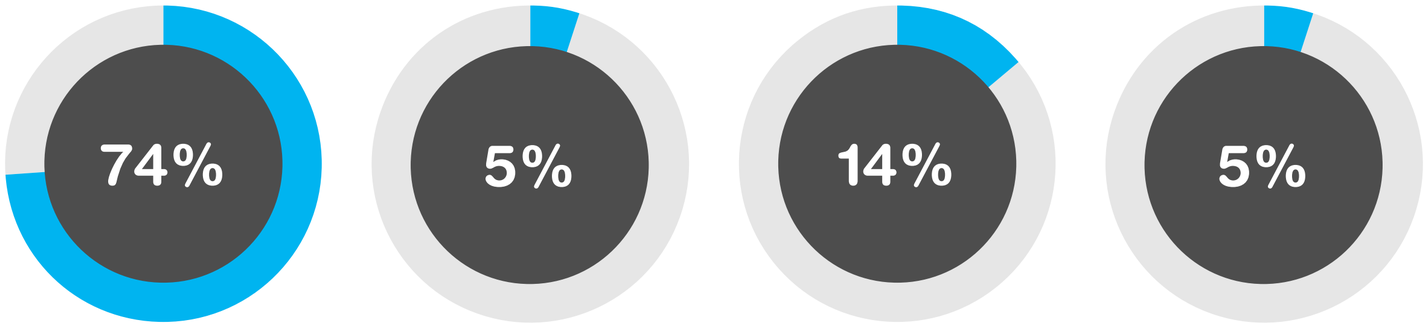
Artur

“I would love to see more people adapt things like GwtMockito to write their test instead of relying on manual testing.”

Daniel

3.6 DevMode or Super DevMode?

Development time is handled by an increasing amount in Super DevMode (20% in total, up from 11% last year). DevMode has synergies with server-side debugging as it can be done in the IDE, compared to Super DevMode's browser debugging. This is probably a reason to DevMode's popularity.



"Coming performance improvements to Super Dev Mode's start-up times will keep this figure rising."

Colin

"It is good to see that Super DevMode is rising, given that all browsers start to take measures that will make DevMode support impossible in the future. I suspect the rising support for mobile platforms to be, in part, responsible for this increase."

Thomas

"We really should push Super DevMode to be the default: running code in it gives a much better feeling of the real application performance. I guess DevMode is dominant because it is easier to setup at the moment."

Joonas

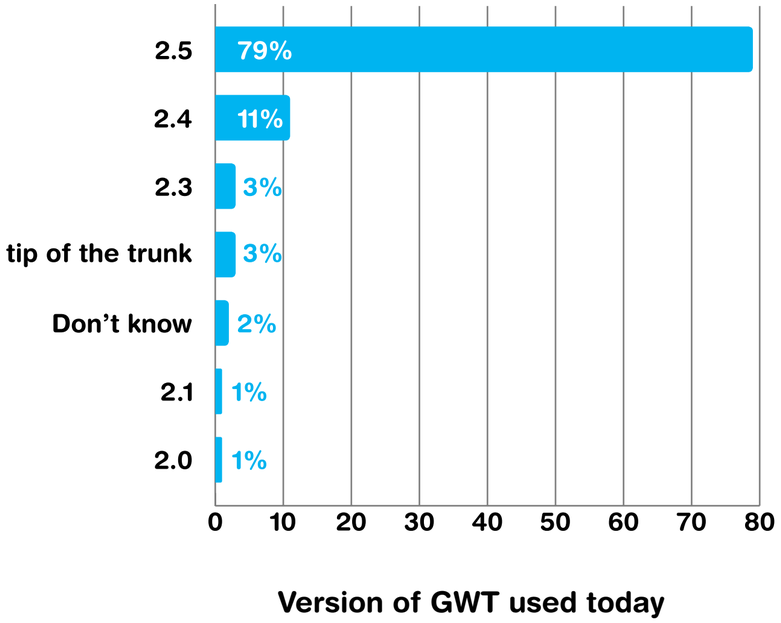
Section 4:

What version are you using?

4.1 GWT versions

To get a better understanding of how the release of GWT 2.5 has been adapted and how older versions of GWT live on, we decided to compare previous years' results to this year's.

GWT 2.4 users have quite well switched over to the newer 2.5 release within just one year.



Version of GWT used today

"I'm pleasantly surprised to see 3% of respondents using "tip of the trunk". We plan on publishing nightly builds which might make this number grow in next year's survey."

Thomas

"I am happy to see that so many people are using the latest version, which means that there is a lot of investment going on. Also this means that a lot of people will benefit from newer features we will ship."

Daniel

"It's great to see users embrace newer releases of GWT - this enables us to get features earlier into the hands of users and get quick feedback. Thank you, users!"

Bhaskar

"80% of the users on GWT 2.5 is very good to see, it means the GWT platform is less fragmented. Hopefully pushing out nightly snapshot builds to Maven Central will make it easier for more people to test against the tip of trunk."

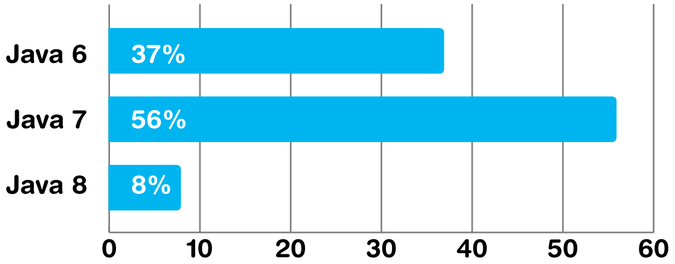
Ray

4.2 Deployment and Development server Java-versions

Backwards compatibility and new features enabled by enhancements in Java8, such as lambda expressions, don't always go hand-in-hand. That is why we wanted to know more about the environment you're working in, before making decisions on where GWT should be heading.

37% of you answered that you still need to support Java6 in 2014 in your development environment, when 56% still needed support for Java7 in 2014. This left a small 7% who could jump into Java8 in their development environment for 2014.

The same question was asked for production environment for 2014 where numbers where slightly more tilted towards Java6.



"If we can deploy a solution that allows Java8 in client-only code, but Java6 everywhere else, hopefully more people will be willing to try it. Java8 lambdas are fantastic and make dealing with asynchronous callbacks and collections far more delightful."

Ray

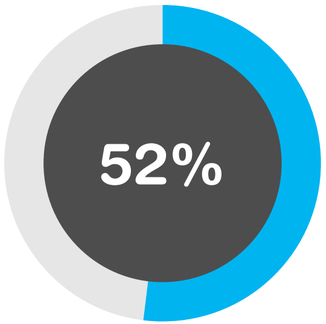
"Deployment requirements probably affect the development numbers as well - people want to have the same environment in development as they have in deployment."

Artur

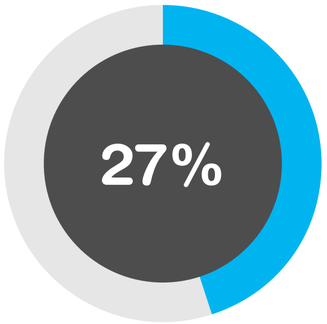
"Lambdas will turn GWT and Vaadin UI development around by making code so much more compact. While Vaadin 7 already supports Java 8, I hope we could see the same benefits in GWT 3."

Joonas

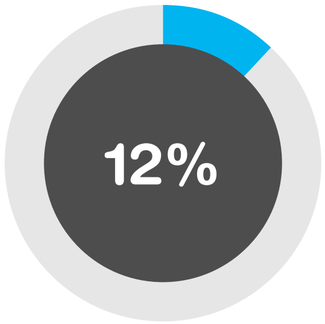
Would you use Java 8 features in the client-side of your GWT app?



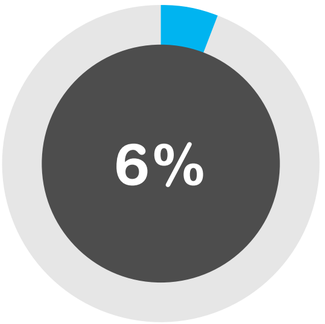
Only if this will not complicate my project setup



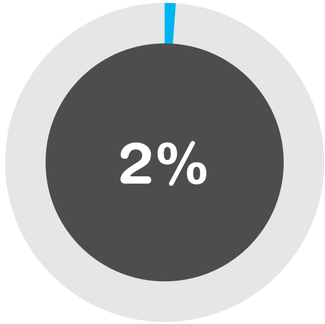
Willing to use different Java versions for client and server



Not interested



Other



Did not understand the question

“80% of the people would like to see clearer code with Java8 in GWT. That is really great. We have to find a good way to make it work for people that require Java6 server side.”

Daniel

“It is interesting to see that there is so much interest in the community towards Java8, even though it hasn’t even been released yet.”

Artur

“Only 27% would be willing to use different versions in client and server. That combined with only 5% willing to use Java8 in server implies quite clearly that requiring use of Java8 would be out of the question. That said, Java8 should definitely be supported, but not required.”

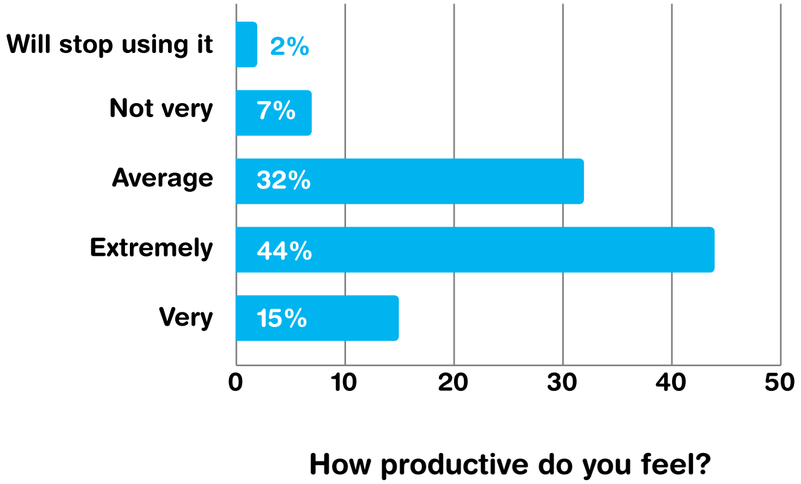
Joonas

Section 5:

Add-ons, extensions and other frameworks

5.1 Productivity with GWT

How productive do you feel using GWT?



“Happy to see that more than 90% feel productive and that 84% would choose GWT for their next project. This shows how committed our community is.”

Daniel

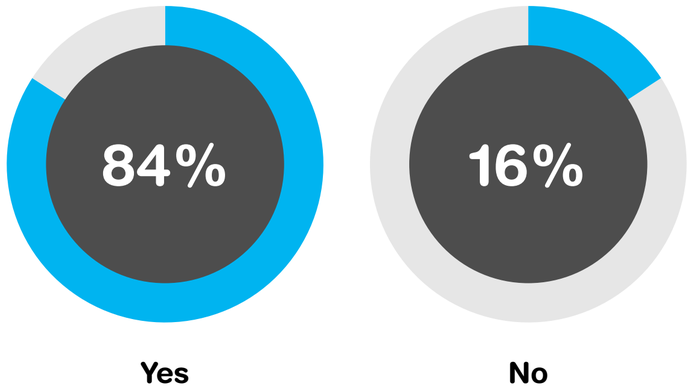
“Glad to see so many people feeling productive with GWT. I would guess that those who feel unproductive are mainly inhibited by excessive compile times/refresh speeds. More speed needed. :)”

Ray

“Community is solid. We just need to attract more people in the community. :)”

Joonas

5.2 Would you use GWT for your next project?

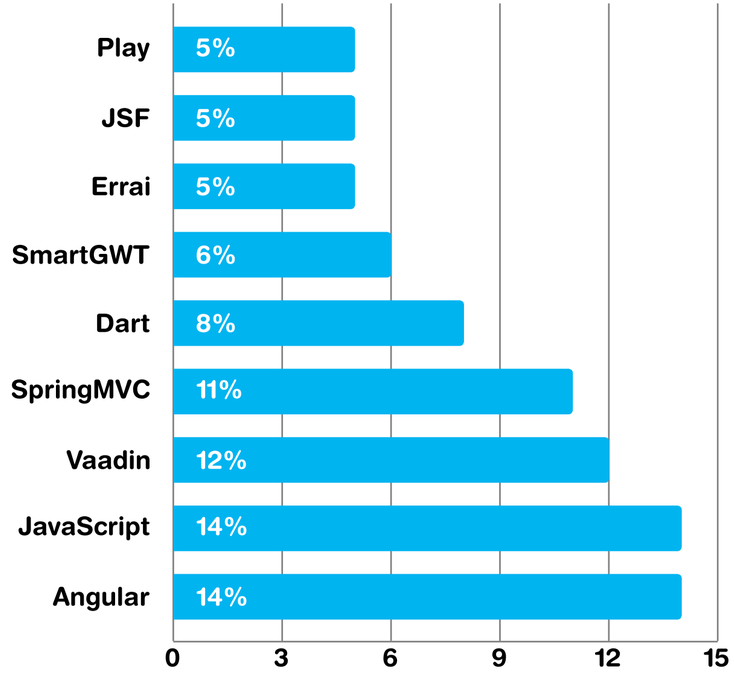


“People that have been working with GWT get super productive once they get used to the GWT way.”

Christian

5.3 What other frameworks would you consider for your next project?

Another take to the question of using GWT in the next project would be: if not, what then? So we asked about the other frameworks people could consider using.



"Nice to see Vaadin so high on the list. Most probably because it is still Java in the same way as GWT so they are close to each other (compared to Angular, Javascript which are JS)"

Artur

Now when Vaadin 7 contains GWT, it is a natural transition path: one can benefit from the superior server-side productivity while keeping everything they enjoy in GWT."

Joonas

"GWT is used mostly for big applications where static typing and all the development tools are really helpful. I can certainly understand that people like AngularJS which is good for small applications that do not have a long live cyle since it is more lightweight, but GWT is really good at big, complex applications."

Daniel

GWT 3.0 will feature a new JS Interop system which makes it much easier to interact with JS libraries like Polymer or Angular without needing to write JSNI bridge code. It should allow people to choose whatever frameworks best suit their purposes."

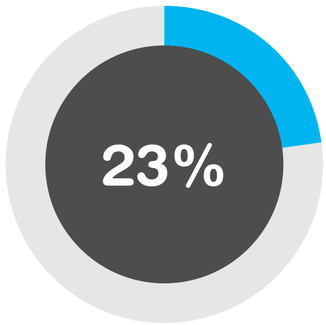
Ray

"I like that many people also consider AngularJS, since the Googlers that are working on this do a great job, but with the new JS interop you can use Angular inside of GWT without any friction. As a side note: There is even an Angular implementation for GWT out there."

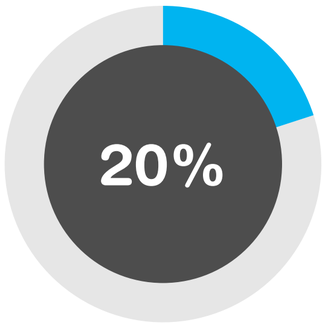
Daniel

5.4 Extensions everywhere

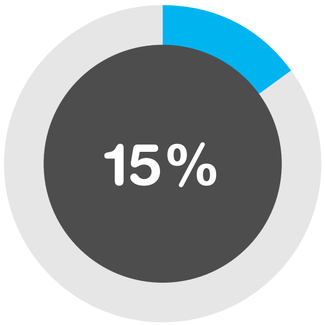
People are using a lot of extensions with GWT. Here are the key areas where extensions are used most.
As many as 87% of all respondents are using some add-ons or extensions in their GWT projects.



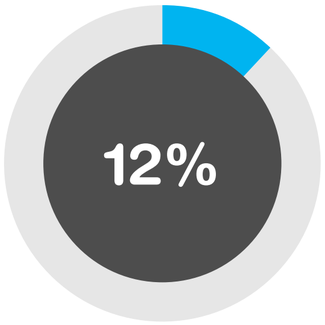
UI functionality (like GWT-dnd, GWT-fx)



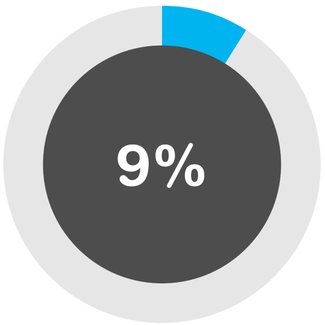
Application structure (like gin, GWT-presenter)



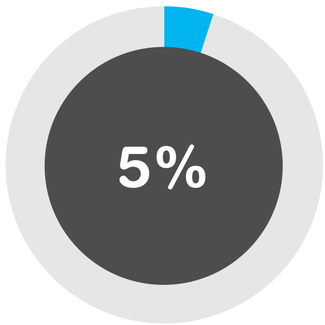
Data access (like GWT-rpc, GWT-dispatch)



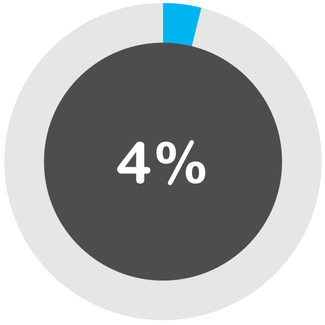
Access to services (like Google APIs, GWT-bootstrap)



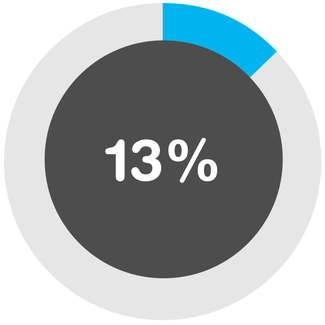
Application functionality (like GWT-platform)



Other (please specify)



Data handling (like gilead)



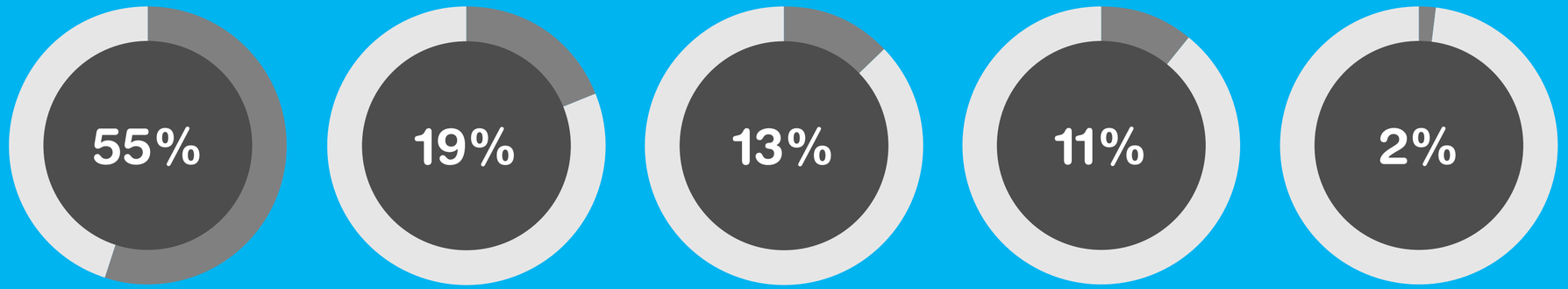
Don’t use any extensions

“The fact that people are using extensions proves that GWT provides a very good core that developers then can build on.”

Artur

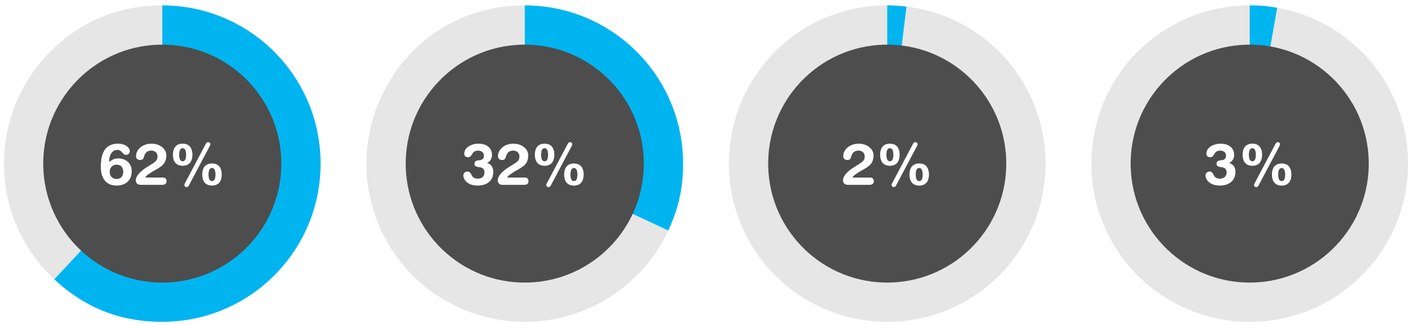
5.5 Where are the extensions found?

To get a better understanding of where you find your extensions we went ahead and asked. Turns out you find your extensions in the same way as you find your pizza; you google for it or create your own.



5.6 Do you integrate existing JavaScript into your project?

GWT gives you the possibility to combine Java and native JavaScript – but how many of you are actually using JavaScript either directly or wrapped?



"The new JS interop will make including JavaScript much easier and a newer Widget system will hopefully help people with reducing their JS dependencies."

Daniel

"There's always something that's very well done in JS and you don't have the time to write your own widget so you just wrap it."

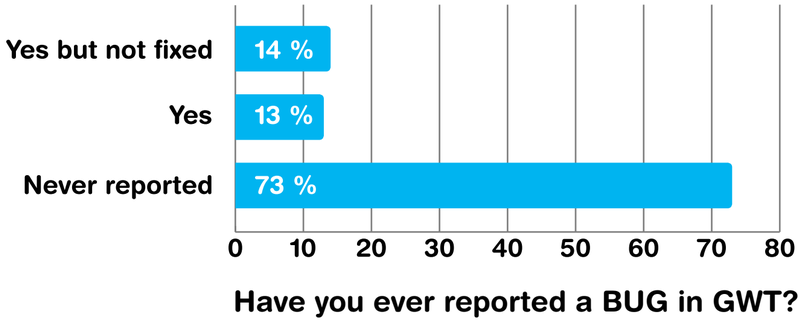
Christian

Section 6:

The future of GWT

6.1 Bugs in GWT?

All software has bugs but we wanted to get numbers on how many of you actually report bugs that you find. GWT being an open source project means that the future of GWT relies on the community's activity and eagerness to report findings.



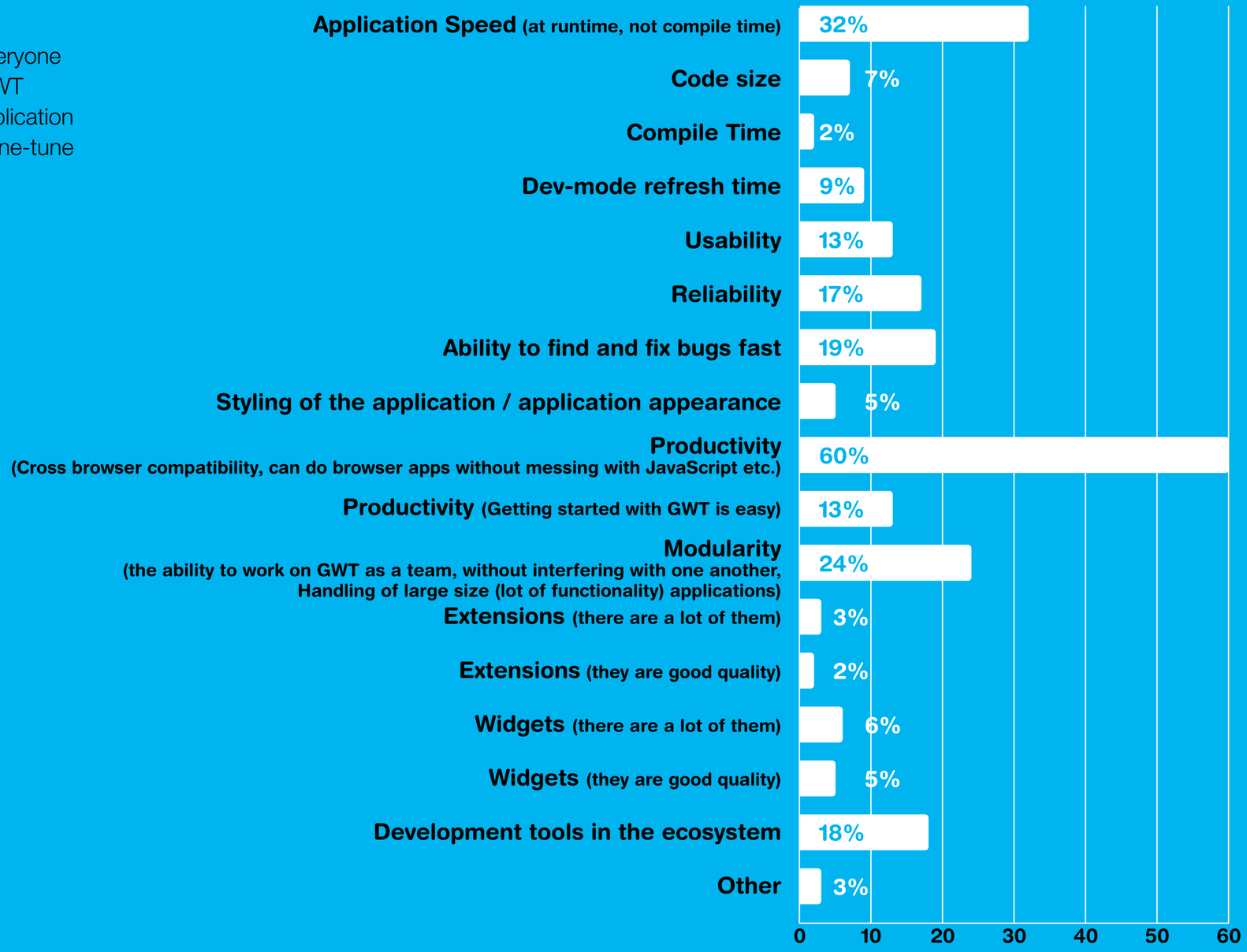
Have you ever reported a BUG in GWT?

"I'm not sure how to interpret this: is GWT "good enough" that people don't run into bugs, or are they too "lazy" to report those bugs?"

Thomas

6.2 Top features in GWT

What are the strong points where GWT really shines? We asked everyone to name two of their favorite features. Although compilation with GWT takes time, the framework saves time by magically making your application cross browser compatible and well optimized without the need to fine-tune JavaScript by hand.



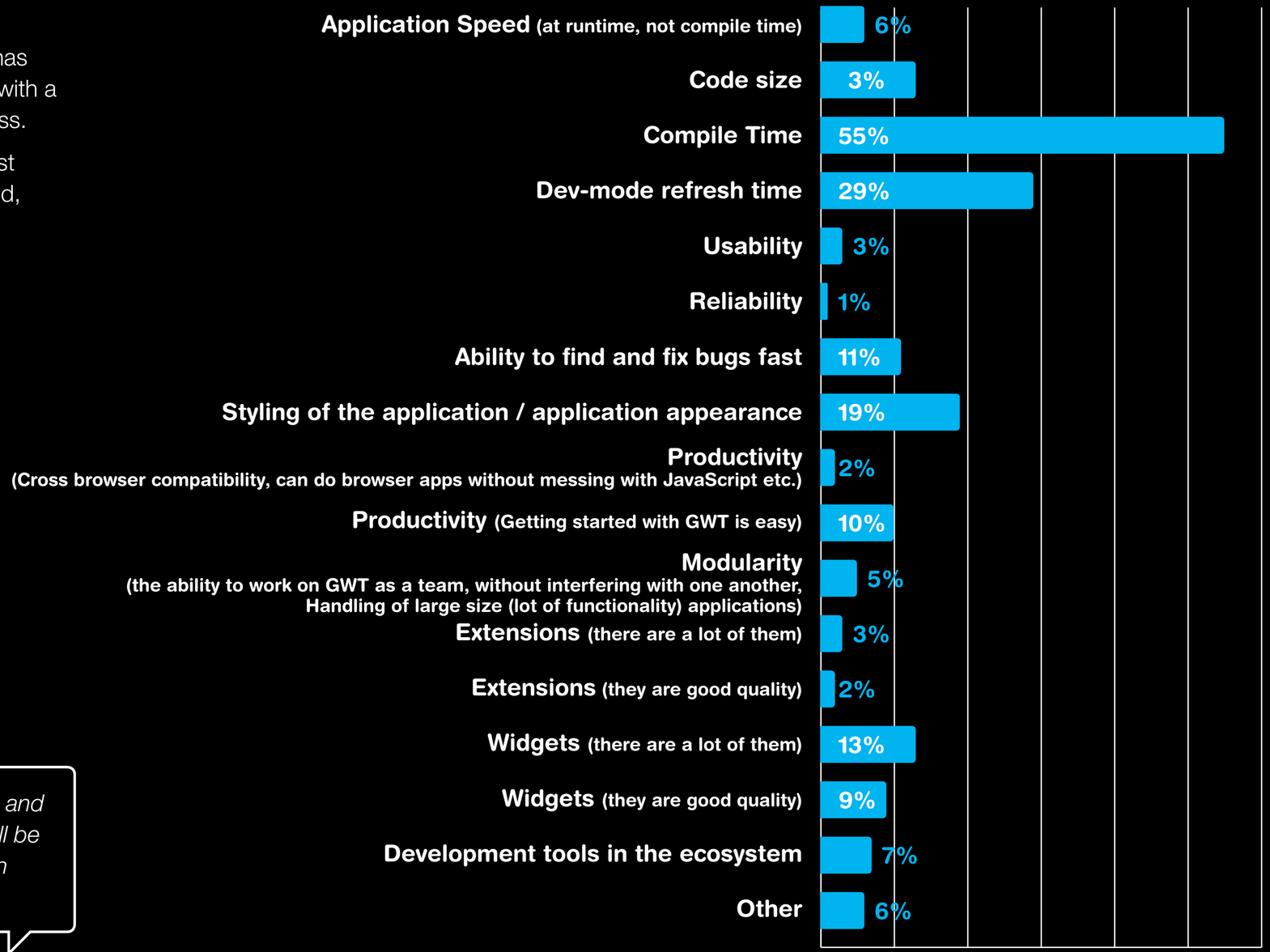
"I am actually working on making GWT apps run faster, so we will get even better application speed in the future."

Daniel

6.3 Welcome to the dark side of GWT

You thought there were no dark painful secrets? Every technology has them. When someone claims a technology to be perfect, take that with a grain of salt: reality distortion fields are known to exist in this business.

To discover pain points, we asked people to name the top two worst features of GWT. Conveniently, this gives us a list of things that could, should, must and eventually will be improved.



"We will introduce a new feature that will give users the same refresh / edit cycle that they can achieve in JavaScript. Going forward with super dev mode will allow for blazing fast refresh times, while developing. Also GWT 2.6 already shipped with some improvements to compile time."

Daniel

"Fortunately Compile time and Dev-mode refresh time will be the major improvements in GWT 3.0"

Thomas

6.4 Your wishlist

In that light, we wanted to help guide future development, so we asked people to share their thoughts about the problems they are facing with GWT, missing features and extensions as well as their thoughts for the future killer features of GWT 3. And we got what we asked for - even more than we asked for! In fact, we got so many suggestions that it is virtually impossible to list them in this report on a page, two or even ten.

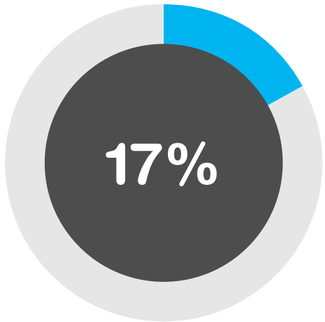
So instead of that, we decided to post the full wishlist of online. On last year's wishlist Faster Compile time made the Top-3 of the list and that is currently the thing we've been working on, to a large part thanks to your feedback.

What else should GWT 3 include?

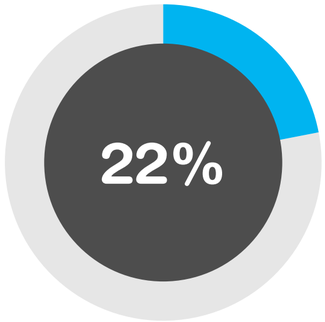
> **Just go to** vaadin.com/gwt/report-2013/wishlist **to see it all, contribute your thoughts and vote the best ideas.**

6.5 How about them steering committee members?

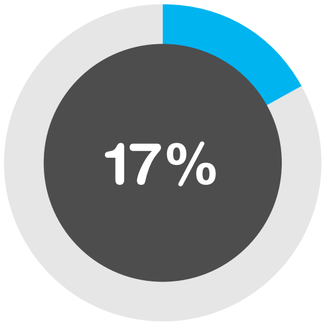
Last year the community still thought it was too early to tell about the changes the Steering Committee would bring to GWT (50%). However, 36% already then thought it would get better thanks to the Steering Committee. This year we asked about how the Steering Committee has changed things and 56% feel it has gotten better from before.



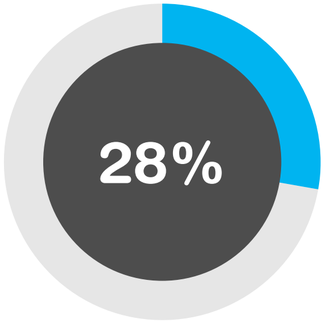
GWT feels much more alive once again



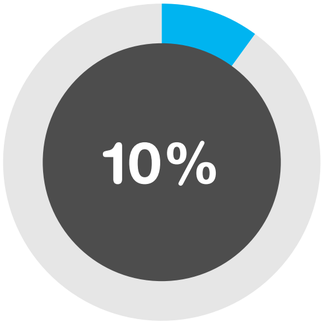
I feel much more confident about GWT's future



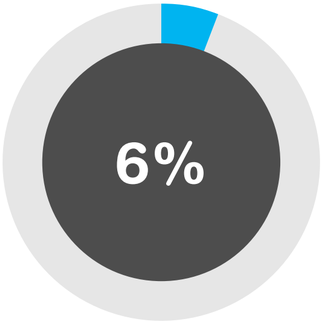
Much clearer roadmap



Nothing has changed



I have less confidence in GWT now



Other

"This shows that the Steering committee is doing a great job, since 90% seem to agree with its work. With a clearer roadmap, people can use GWT with more confidence."

Daniel

"It is great that people feel positive about steering committee. I feel that we still have lot to be improved. With more regular and better documented meetings we could get the community more involved."

Joonas

Conclusions

With over 30 pages of data, charts, stats, and commentary from some of the most well-respected folks in the world of GWT, we'd like to think that this report is the most complete survey of the GWT community to date. We've looked at everything from the composition of GWT teams to the worst features of GWT to predictions for the future but the one thing that stands out most about the Future of GWT is this: Now that the codebase for GWT is open for contributions, The Future of GWT really is in your hands, the hands of the community.

Here's how you can get involved:

Check out the project webpage at
http://www.gwtproject.org

Get your hands on the code at the GWT Repository
https://gwt.googlesource.com/gwt/

Follow the conversation of the Steering Committee
https://groups.google.com/forum/#!forum/gwt-steering

Report a Bug
http://code.google.com/p/google-web-toolkit/issues/list

Gerrit Voting
https://gwt-review.googlesource.com/

Share this report!
https://vaadin.com/gwt/report-2013

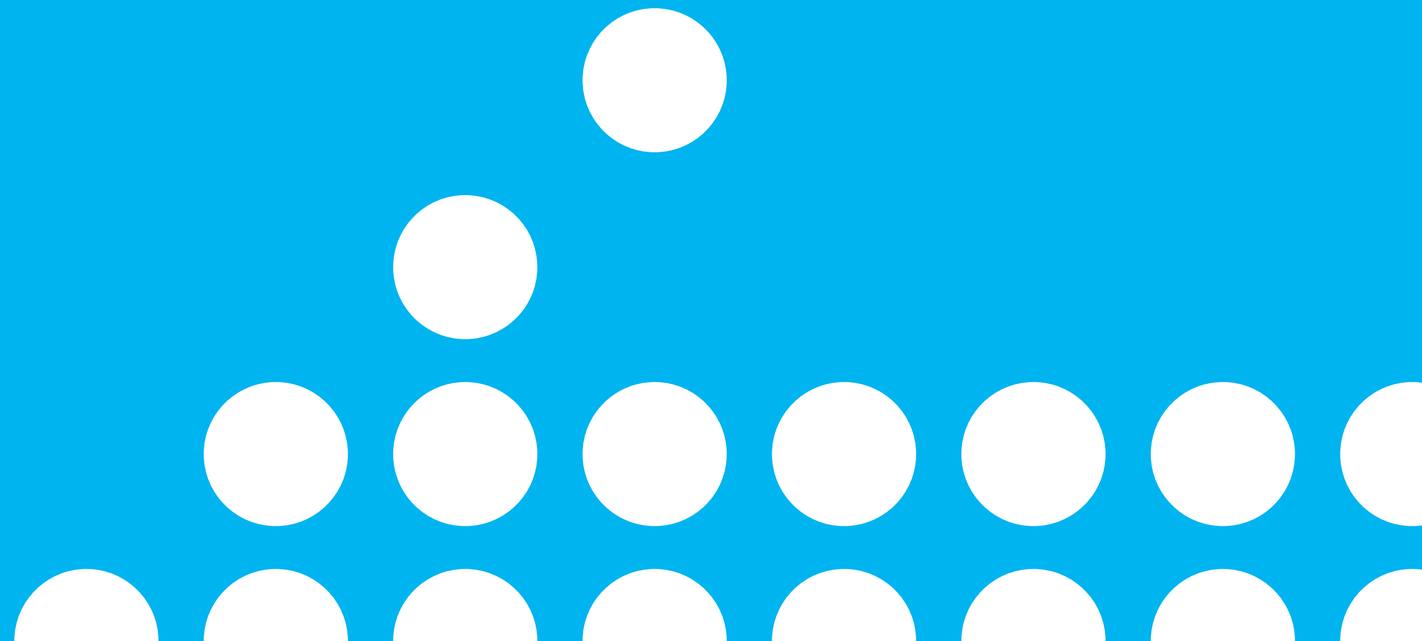
GWT wishlist
https://vaadin.com/gwt/report-2013/wishlist

About the survey

This survey has been created through the work of Vaadin, with the support and contributions of the GWT Steering Committee and GWT-minded individuals. We'd like to thank everyone involved in the survey and who contributed with ideas and comments. We couldn't have done it without you.

We humbly recognize that there may be problems with some of our data, due to the sample size (n = 1420) and the self-selecting nature of our respondents. None of our respondents were required to answer any question (no questions were mandatory), nor were they required to provide contact information. That being said, we'd like to graciously thank those who contributed their time and experiences into our 40+ question survey - and we look forward to creating a similar report next year, to track the changes that happen over the next 365 days and to get feedback on the next concrete steps for GWT.

The survey will also continue online. We asked many open ended questions about the most central enhancements in GWT and collect this into a Wishlist for GWT. Check out the wishlist at vaadin.com/gwt/report-2013/wishlist and cast your vote.



vaadin }> thinking of U and I

Pssst!
Want your bug in GWT fixed, a feature implemented or help in your GWT development?
Contact us at vaadin.com/support