



The Future of GWT Report

Statistics of today and desires for tomorrow
2015

GWT.create brings out the best in GWT

The Future of GWT survey has now been running for three years in a row with over 1000 respondents each year, letting the GWT Steering Committee and the rest of the community get a glimpse of what kind of apps are being written with GWT, with what tools and by which kinds of teams. This has proven to be valuable input for the steering committee when talking about community usage and how some decisions might affect usage in the community.

While running this survey, we've learned a lot about GWT and want to share the findings with you while going through your answers. It's amazing to see the amount of engagement you have put into this survey and we wanted to find ways to crystallize your collaborative thoughts in as understandable a way as possible. In order to do this, this year we've introduced tag clouds and picked out some comments that you've written that depict the rest of your comments as clearly as possible. It won't do justice to all of you but it should give you an idea of what the GWT community at large thinks.

This survey was conducted as part of the GWT.create conference in 2015 – the largest GWT event of its kind. GWT.create will be a large stepping stone for GWT, with GWT 2.7 released in 2014, and also news on Inbox for Gmail being built on GWT and numbers such as over 150 000 developers actively using the SDK. GWT as a technology is going places all the time. It will be awesome to see what 2015 and the upcoming version 3 have to bring for GWT.

We'd also like to humbly thank all our corporate sponsors of GWT.create that have enabled the creation of this survey, namely Apptio, ArcBees, IBM, JetBrains, Magnolia, Sencha and Vaadin. But the biggest thank you goes to You. Without You, there wouldn't be GWT and there wouldn't be answers to a survey from which to make a report. I just wish we have been able to do your work justice in this report. Enjoy!

Fredrik Rönnlund,
VP of Marketing & GWT fanboy, Vaadin



Contents

1.

GWT score

2.

About you

2.1 How long have you been using GWT?

2.2 What GWT based framework are you using?

2.3 What is your role in the company?

2.4 Your team size

2.5 How has your GWT team changed in the past 12 months?

2.6 GWT usage globally

3.

Look at those Apps!

3.1 Application size

3.2 Why isn't this thing responsive?

3.3 Browser support in 2015

3.4 Mobile, tablets and desktop

3.5 Whom are GWT apps written for?

3.6 Where do you deploy your applications?

3.7 What cloud platform do you use?

4.

Development time

4.1 IDE, IDE, IDE

4.2 What version of GWT are you using?

4.3 Java 8 is great, how about you?

4.4 Java, XML or visual designer?

4.5 How are you talking to your backend?

5.

Other frameworks and GWT

5.1 Will you include GWT in your next project?

5.2 What other frameworks

/technologies will you consider using?

6.

Steering committee and survey

6.1 How about the Steering Committee members?

6.2 Feedback on the survey

7.

Rotten tomatoes and bacon covered rockets

7.1 The Bright side of GWT

7.2 The Dark side of GWT

7.3 The longest constructive feedback for the survey was of 1083 characters. We think it deserves a page of its own.



Section 1:

GWT score

With 1101 cast votes, on average people would give GWT the score



4.47 out of 5.00

82% would rate it at 4 or higher. We'd like to think this is quite extraordinary for any framework!

Section 2:

About you



2.1 How long have you been using GWT?

The first version of GWT having seen daylight already in 2006 lets us know how stable GWT really is. And of all the over 1100 respondents to the survey, 78% have already been using GWT for over 2 years. Only a little minority of 8% have come across GWT during the past 12 months, but this minority also tells us that there are new developers coming into GWT at a very healthy pace.



2 years or more



1 to 2 years

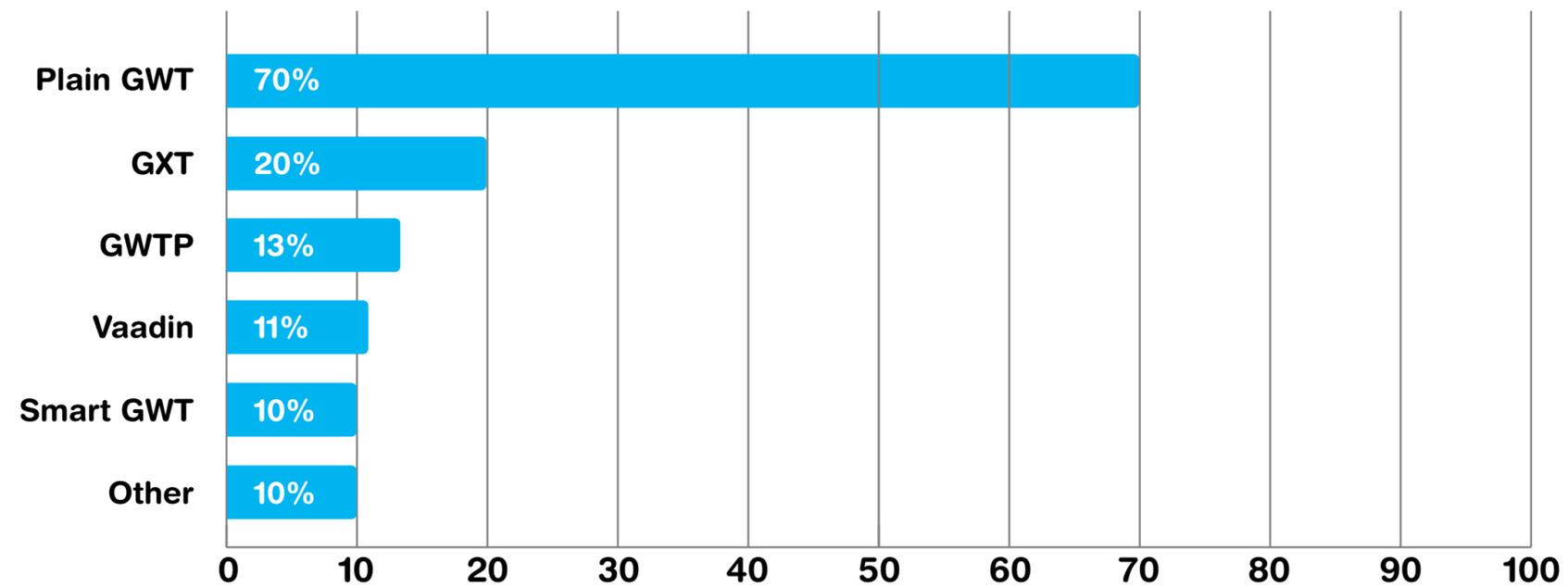


less than a year

2.2 What GWT based framework are you using?

In order to get a better understanding of what you are using in the community and to better understand the rest of your answers, we decided to start the survey by finding out the “flavour” of GWT that you are using. As GWT really is a community project, driven by not one company but a steering committee and the community, it makes sense that there are many varieties of GWT around and that people are combining GWT with other GWT derivatives as well.

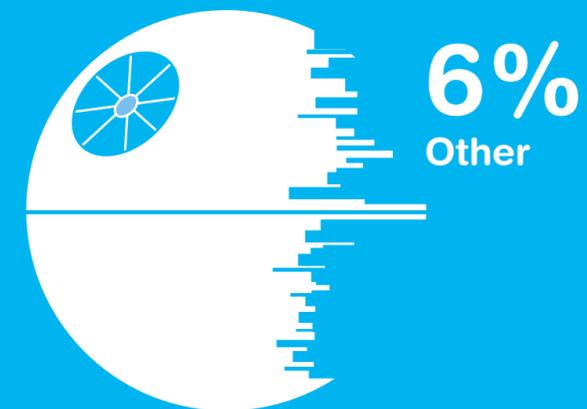
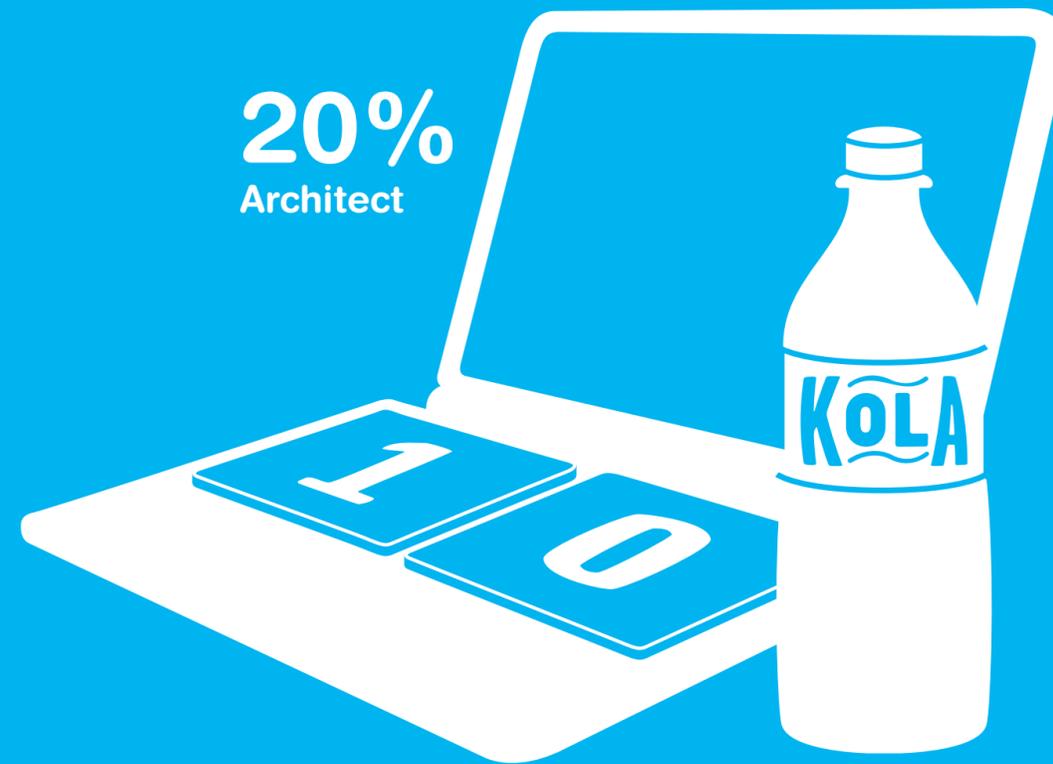
What derivative of GWT are you using?



As you can see it is quite typical to be using more than one framework, e.g. through components from different libraries.

2.3 What is your role in the company?

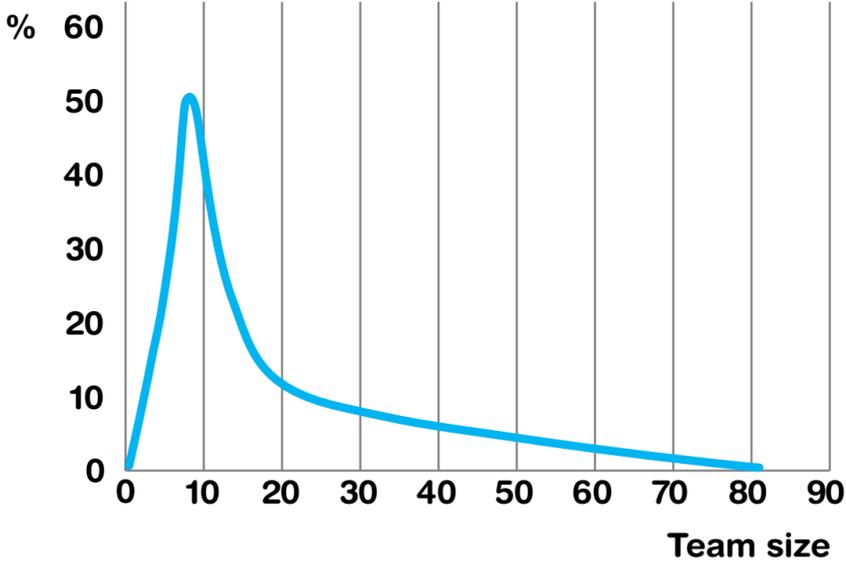
We also wanted to get a better understanding of you answering the survey. Naturally the lion's share of you are developers but there was also a healthy portion of team leads and architects in the team.



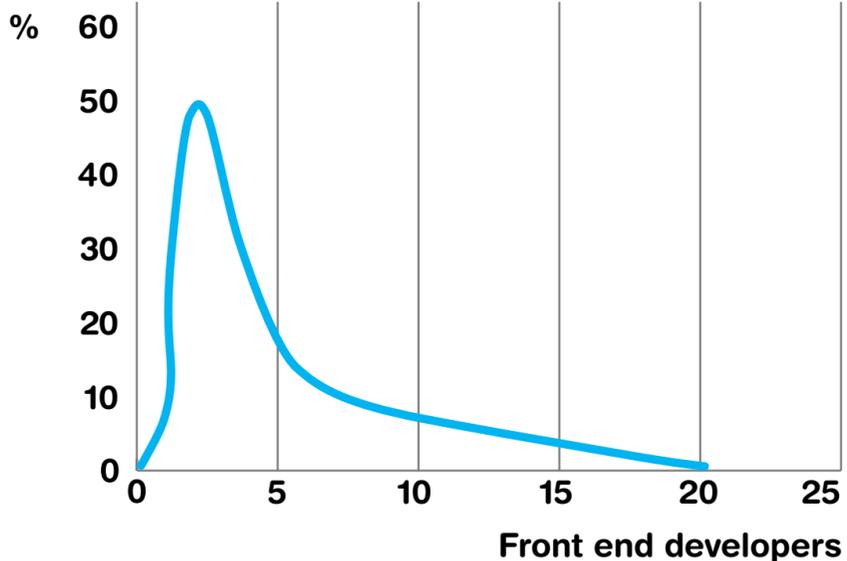
2.4 Your team size

It is quite evident that GWT is being used in some really large enterprises, in some very critical applications. The average company size of the respondents is 4500 with a development team of 19 (median 5) people of which 13 (median 4) are using GWT. On average there are 10% of developers in any given company using GWT.

Team size distribution



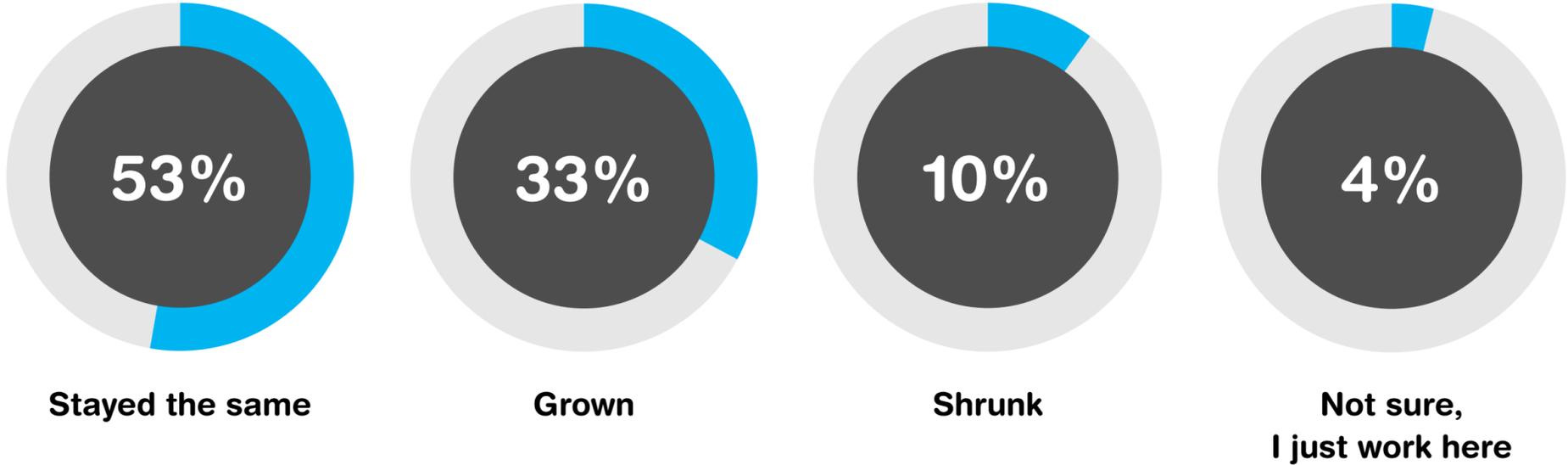
Front end developers distribution

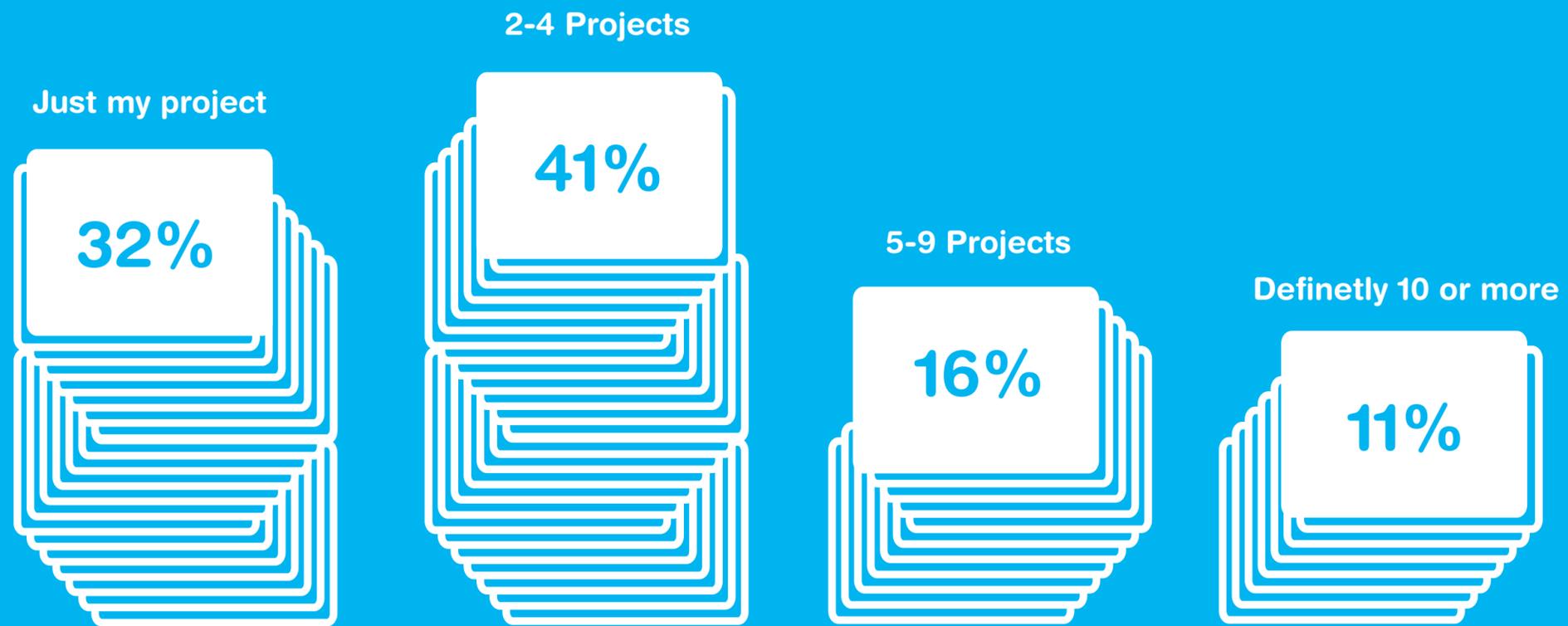


2.5 How has your GWT team changed in the past 12 months?

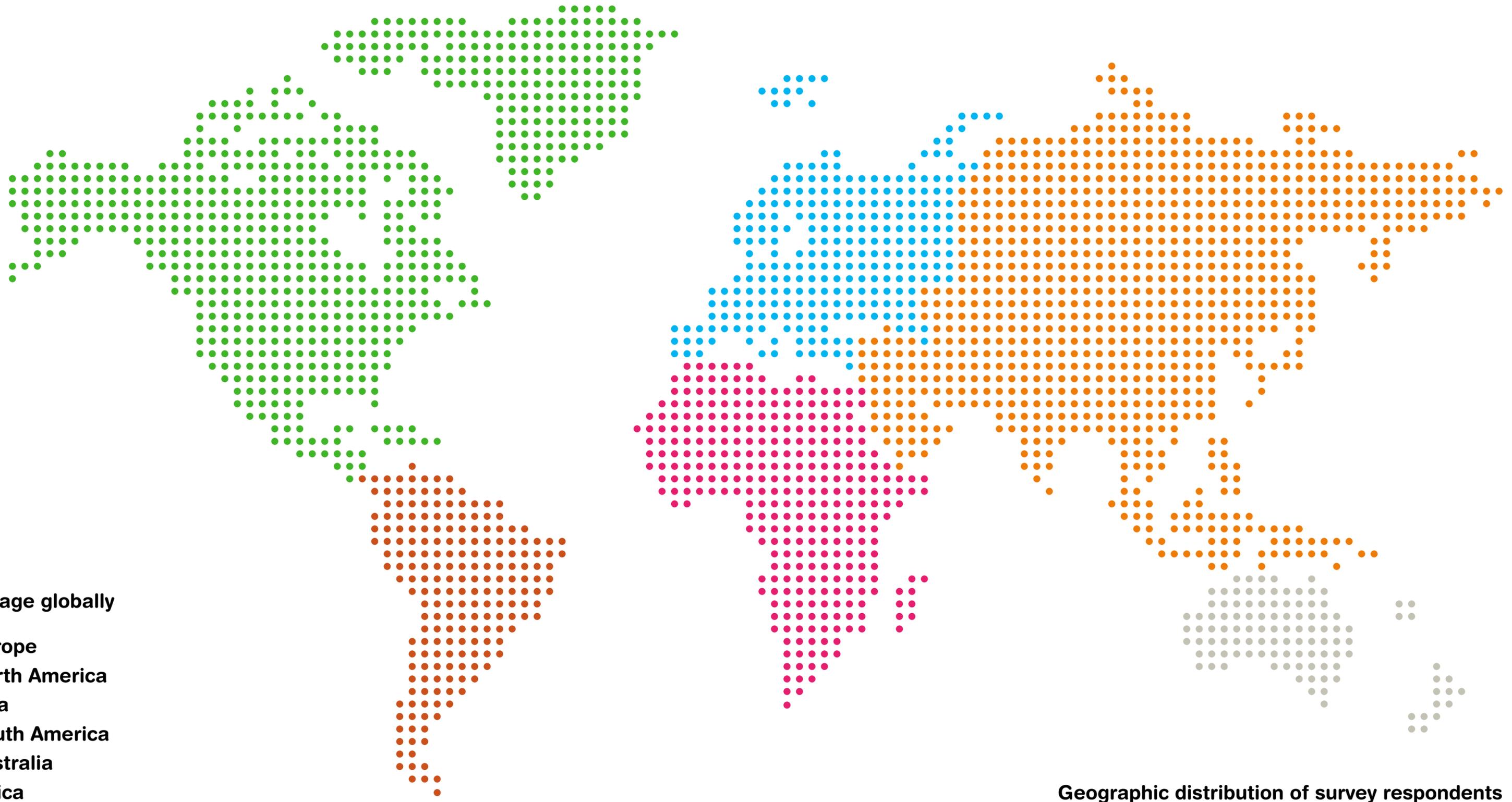
When we asked the same question last year, we got roughly the same answers. A few more of you have teams that have stayed the same while one percent more just works here... it must be that you're so into your app that you don't see what's going on around you. At the same time, there is a surprisingly large amount of companies that have 10 or more projects written in GWT.

Team change in past 12 months





How many projects in your company are using GWT?



2.6 GWT usage globally

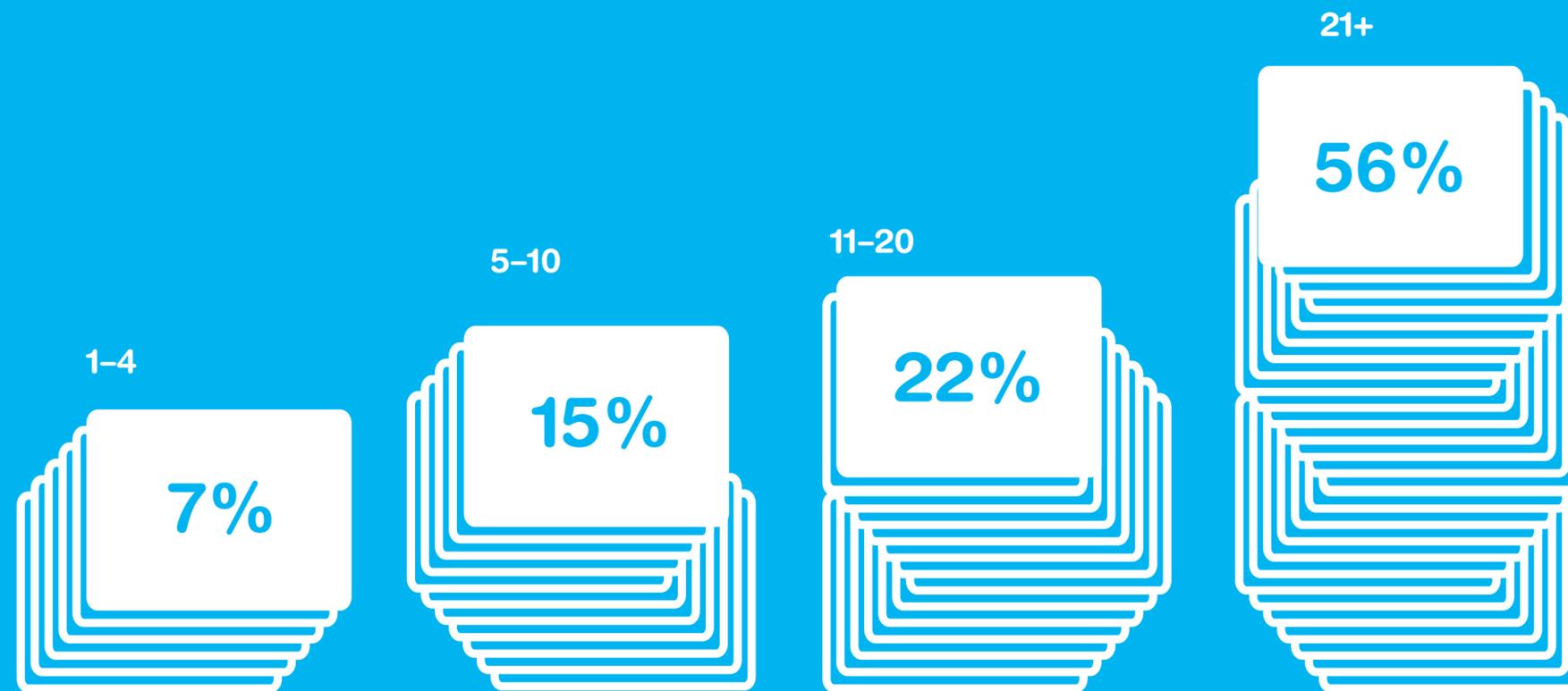
- 53% Europe
- 23% North America
- 9% Asia
- 9% South America
- 3% Australia
- 3% Africa

Geographic distribution of survey respondents

Section 3:

Look at those Apps!





Number of screens

3.1 Application size

GWT applications are huge! Most applications have over 20 views in them which correlates well with the fact that most applications are business applications that are data heavy and are worked with for many hours per day. This also correlates really well with GWT's values: It's easy to maintain and build large applications with multiple teams.

3.2 Why isn't this thing responsive?

It being almost 2015 when we ran this survey, one would have expected most apps having to support responsive layouts through Responsive Web Design (RWD) and that is also the case.



51%

Supports responsive

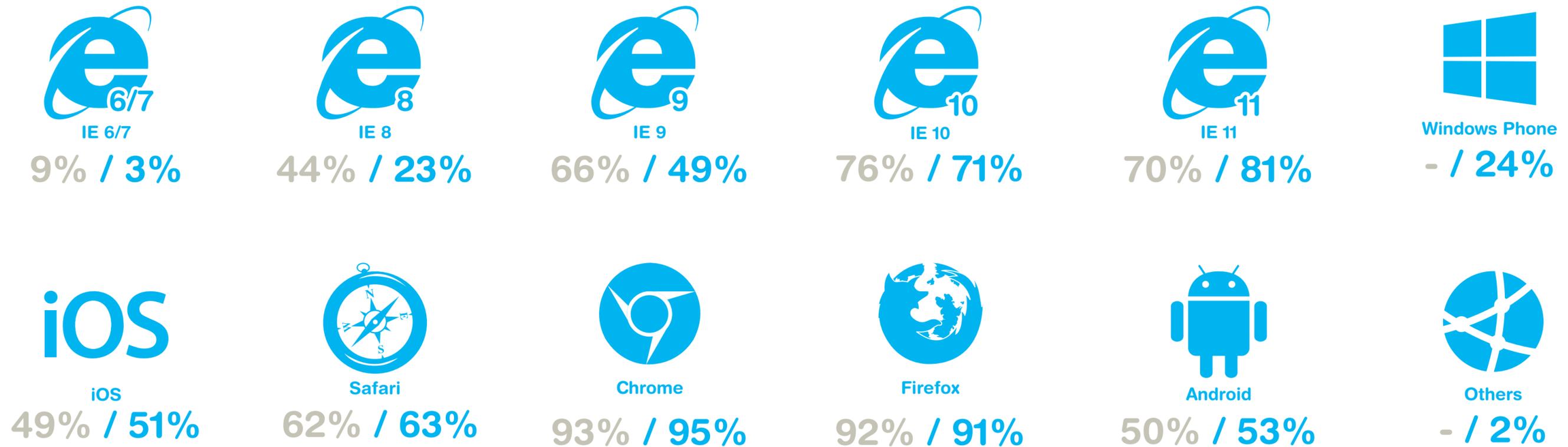


49%

Does not support responsive

3.3 Browser support in 2015

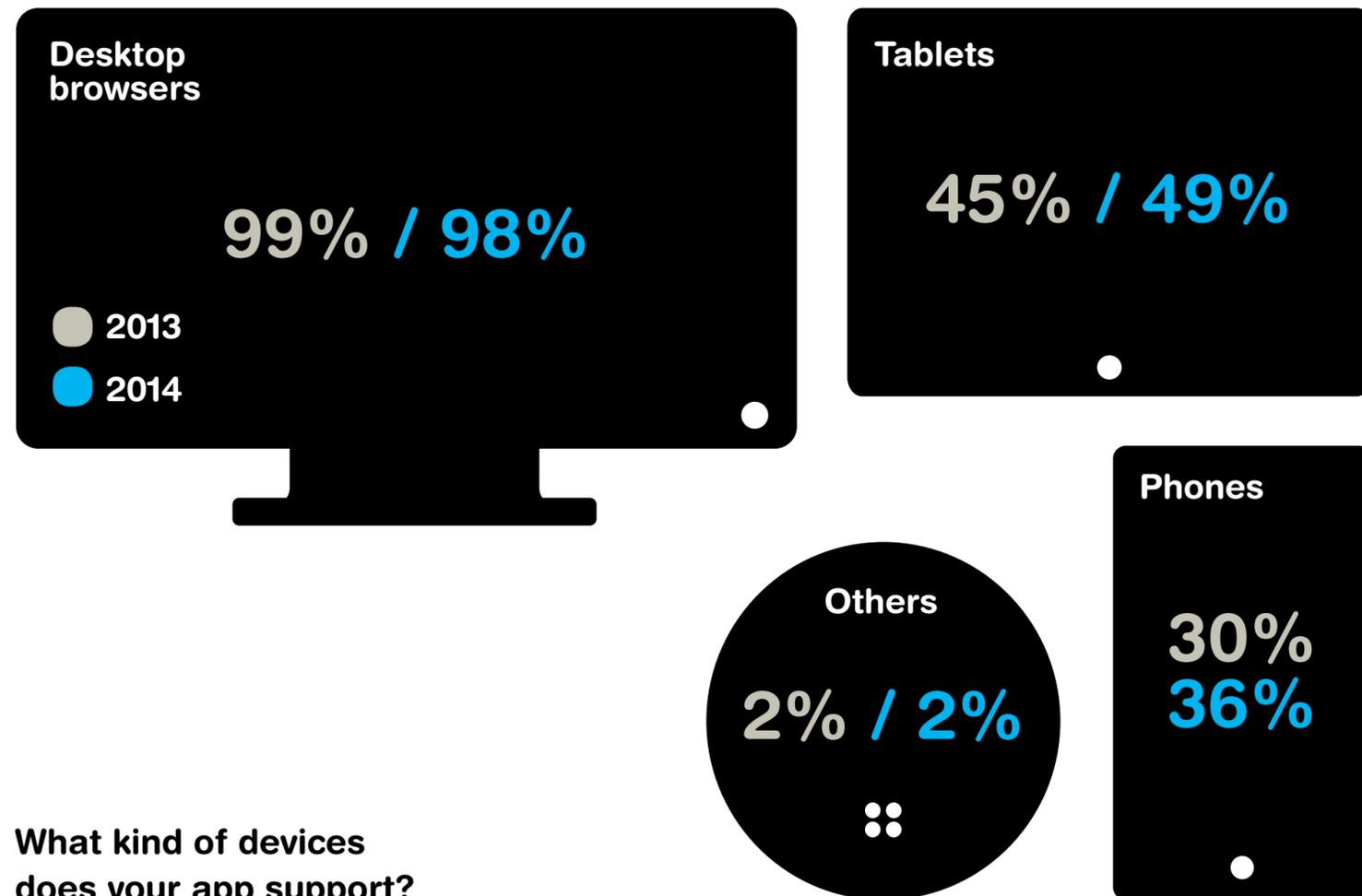
As browsers become better and better, evergreen browsers start to dominate the most supported ones. This is good news for all us developers as we can concentrate on solving business problems instead of environment issues, even though GWT has always been good at abstracting the browser itself away from us. Of all the questions in this survey, this question shows us most how we're moving forward: Back in 2013 IE6 had a market share of 14% whereas today only 3% need to support it.



● Browsers developers expected to support in 2014 ● Browsers developers expect to support in 2015

3.4 Mobile, tablets and desktop

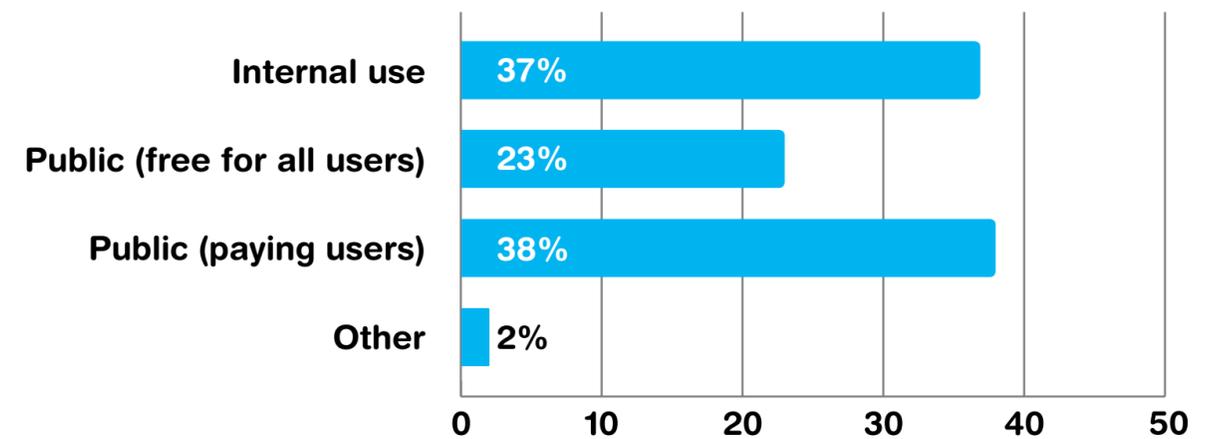
Desktop browsers are still mainstream but tablets and phones are rising steadily.



3.5 Whom are GWT apps written for?

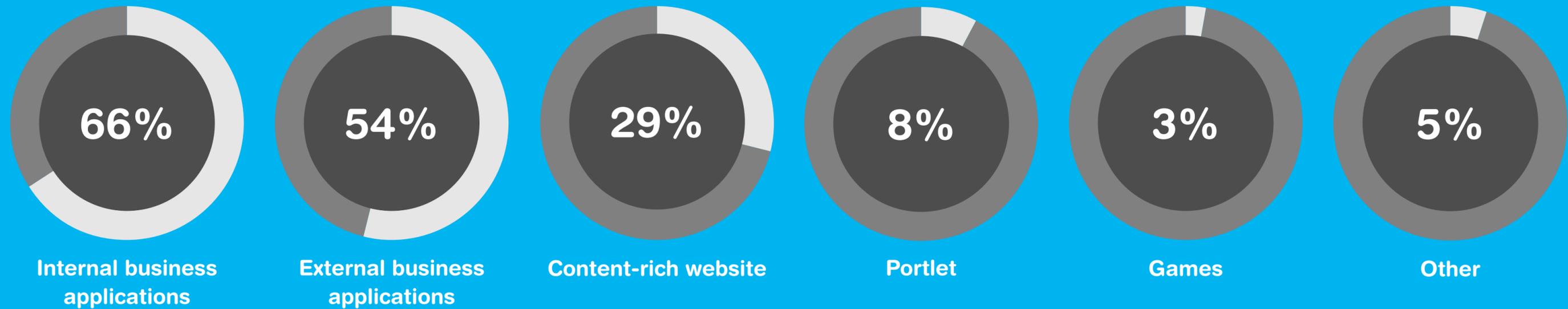
Most of the apps written in GWT are business applications, both ones being used internally and ones being used externally. Only a very small portion of apps are aimed for portals (8%) and an even smaller for games (3%) which shouldn't come as a surprise. What could be a surprise however is that a fairly large portion of apps are content-rich websites (29%).

The amount of public applications written in GWT is considerably higher than the amount of internal applications. Once again in line with last year's results.



Who are the end-users of the applications you're building?

How is GWT used today?

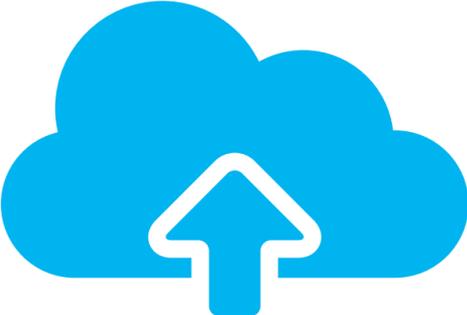


As this was a multiple choice question, the sums of the percentages are higher than 100%.

3.6 Where do you deploy your applications?



65% Own hardware



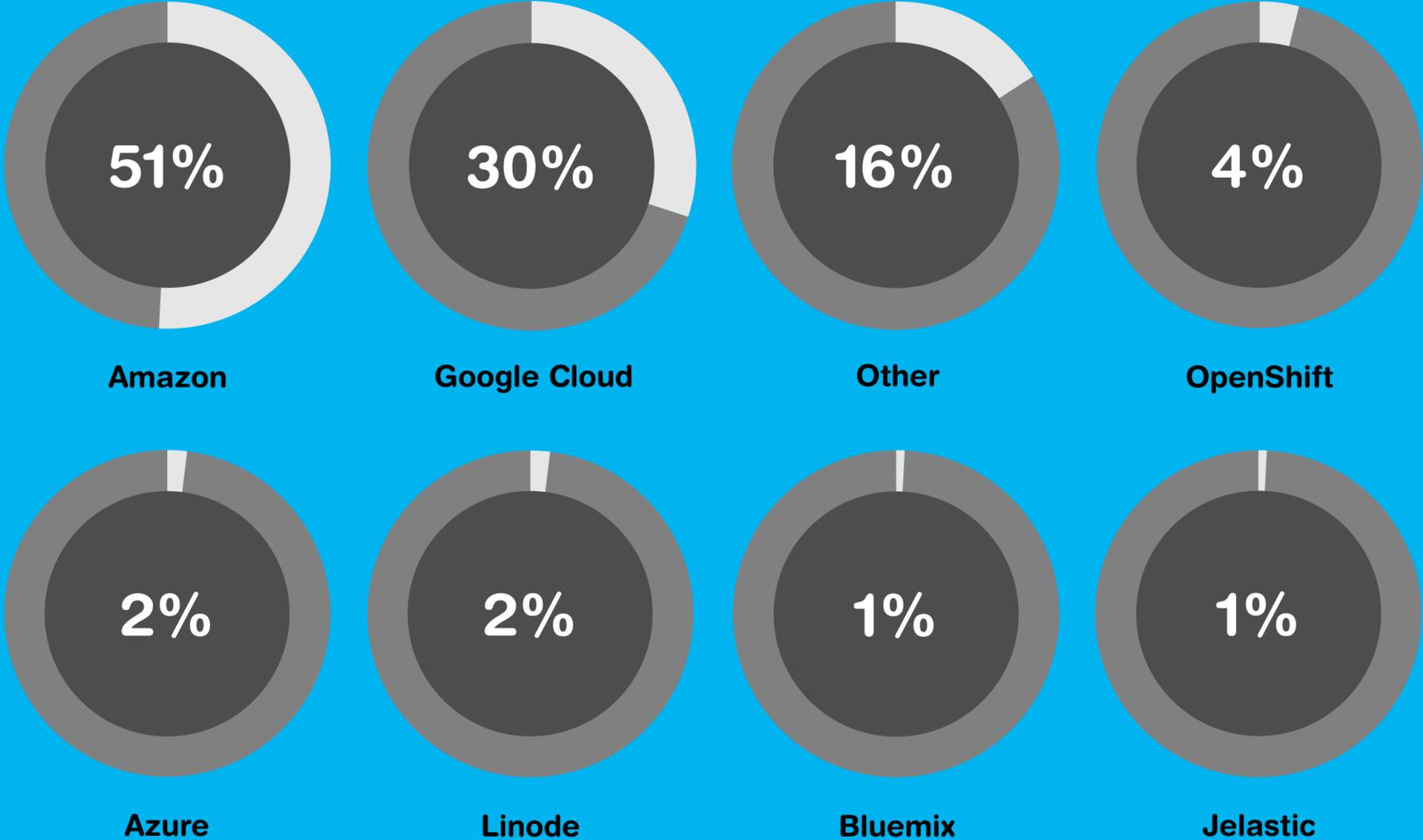
26% Cloud



8% Hybrid cloud

3.7 What cloud platform do you use?

There were those that answered the question on deployment being in the cloud and we wanted to know more about what cloud they're using. The most popular cloud platforms were Amazon and Google Cloud.



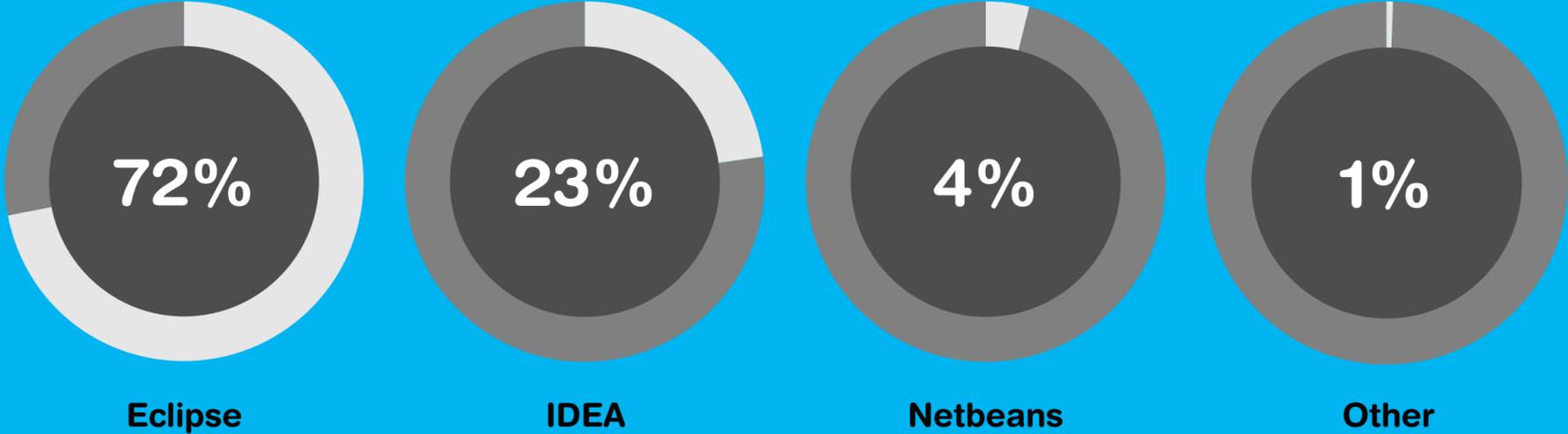
Section 4:

Development time

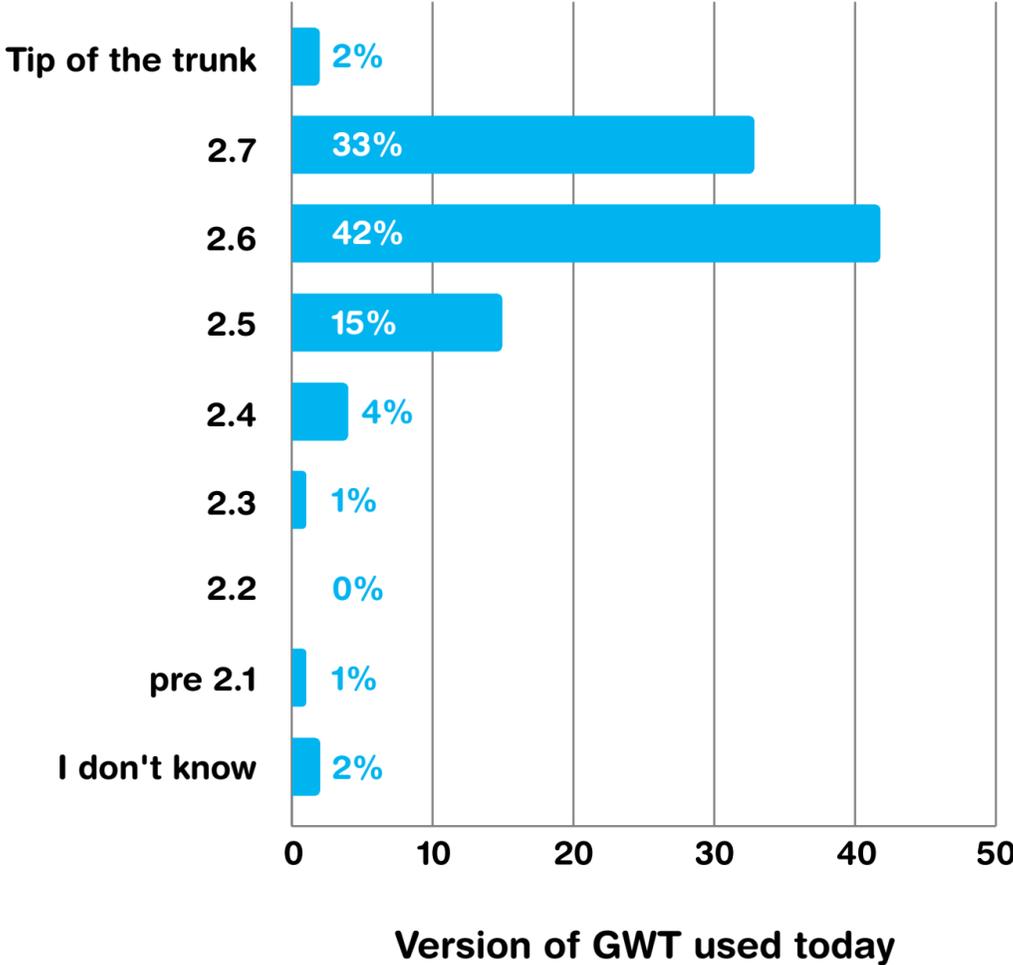


4.1 IDE, IDE, IDE

Just like location is everything in real-estate, IDE is everything in development and we can see some recent trends in IDE with GWT developers. Eclipse is slowly losing users, apparently to IDEA, whereas NetBeans still has its small but steady user base compared to last year.

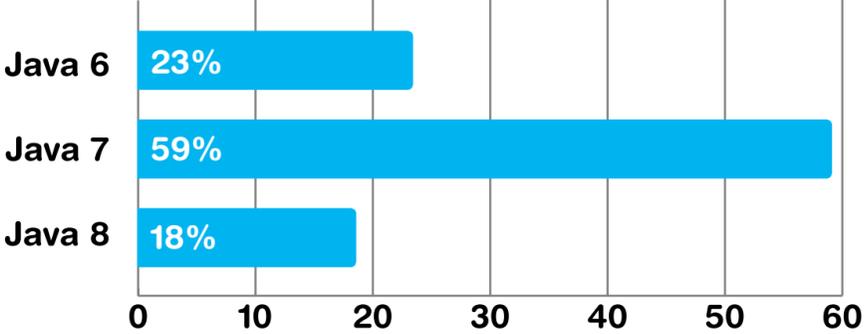


4.2 What version of GWT are you using?



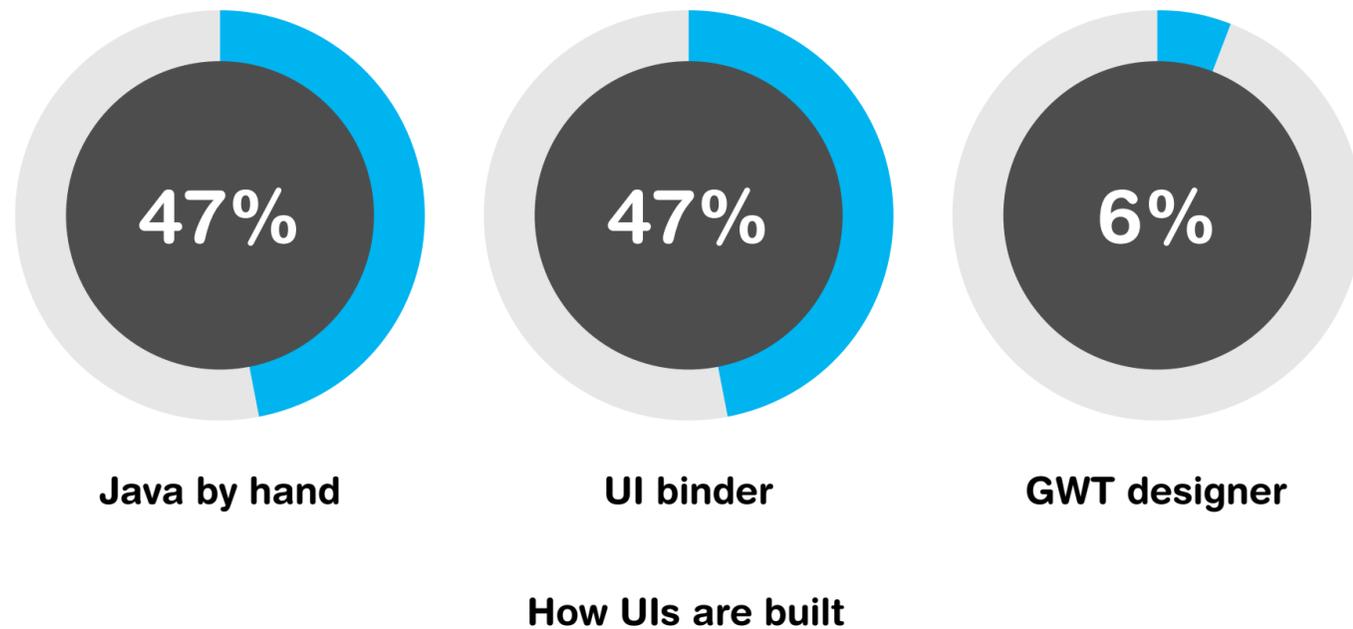
4.3 Java 8 is great, how about you?

It's been almost a year since the launch of Java 8 now but already last year we could see an interest in Java 8 in the survey. Even before its release, 8% of you said you would only need to support Java 8 in 2014. This year the percentage is considerably higher.



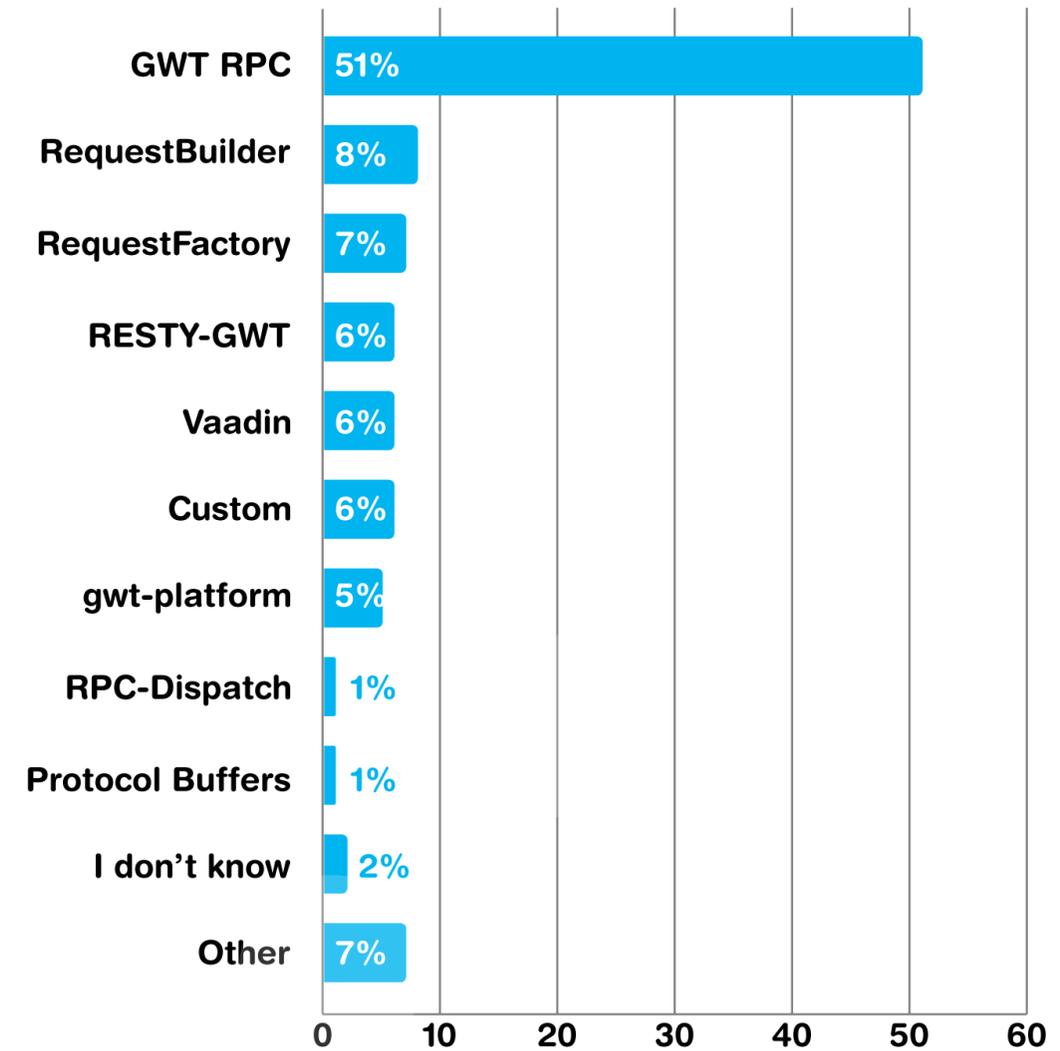
4.4 Java, XML or visual designer?

The numbers are clear for this point – building your UI happens in XML or in Java. Hands down and no questions asked. The nice thing to notice is that the world is quite visibly divided in two: the ones using UiBinder and the ones coding Java by hand.



We thought it was kind of sad to see GWT designer being so low already last year, so we wanted to know about your priorities, if there was a better designer. 56% would use one if it was good, the rest would still stick to Java or Declarative instead. 17% would even be ready to pay for a good designer. The ones building their apps in Java today are slightly more interested in a designer than those building their apps in XML today. There was also a 4% anomaly in the current GWT designer users, who would under no circumstance use a Designer. Some people just have a hard time making up their mind :)

4.5 How are you talking to your backend?



Section 5:

Other frameworks and GWT



5.1 Will you include GWT in your next project?

YES!

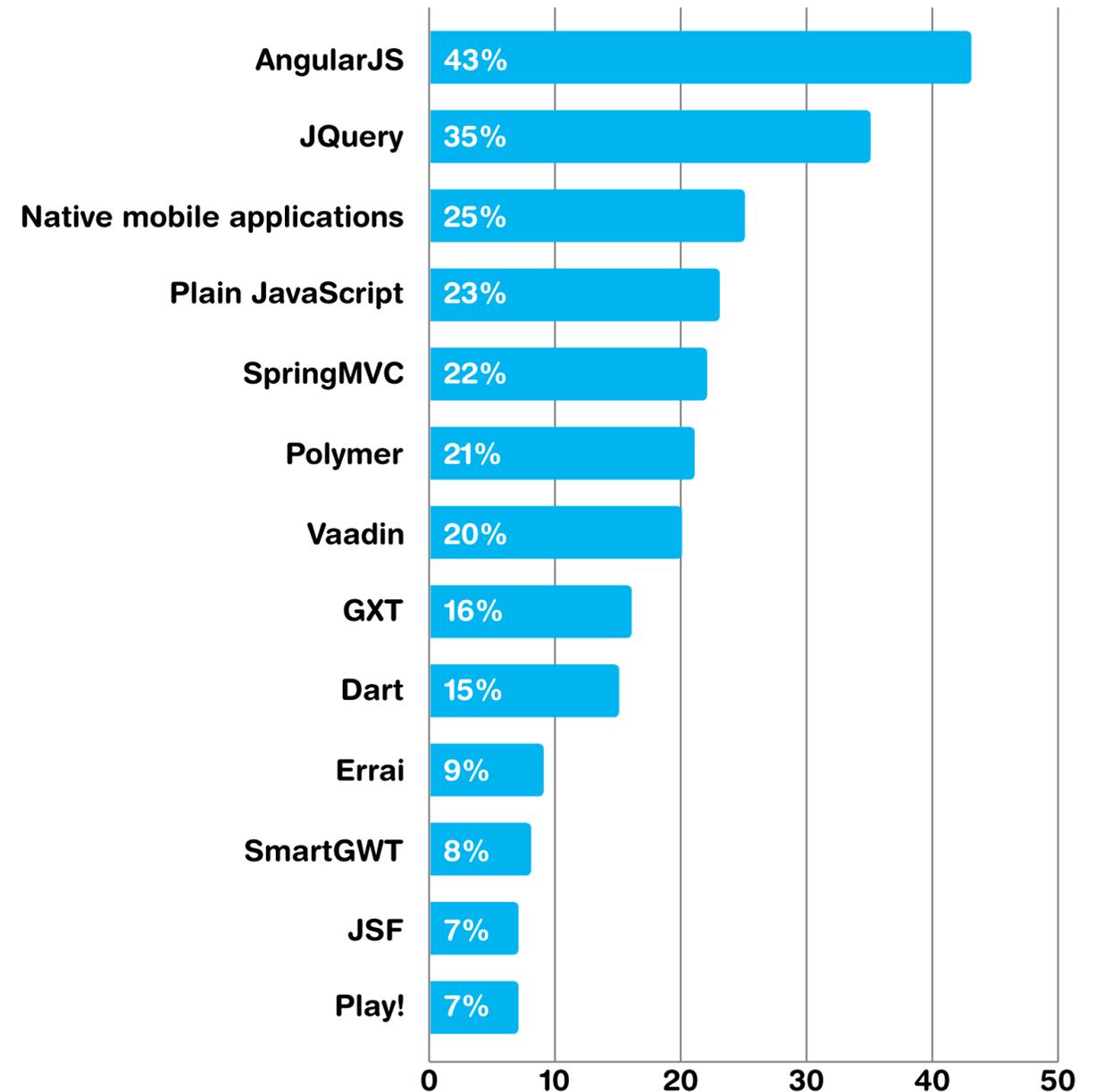
85%

NO!

15%

5.2 What other frameworks / technologies will you consider using?

Learning from last year, this time we posted this question as a multiple choice question. While Angular has maintained its top position, plain JavaScript on the other hand lost interest. None of the frameworks would be evaluated by over half of the developers although AngularJS got very close.



Section 6:

Steering committee and survey



6.1 How about the Steering Committee members?

As you might know GWT today is steered by a steering committee that consists of

Ray Cromwell, Google

Leif Åstrand, Vaadin

Colin Alworth

Justin Hickman, Sencha

Christian Sadilek, RedHat

Thomas Broyer

Stephen Haberman, Bizo

Daniel Kurka, Google

Christian Goudreau, Arcbees

Konstantin Solomatov, Jetbrains

In total you would rate them at a very good 3,5 / 5

"Keep up the good work"

"It would be nice to have a roadmap of what is planned one more year ahead."

"Progress is good but communications seems poor."

"Really appreciate all the excellent work you gents have put in."

"You should clearly communicate what the future of GWT is going to be like in 1-2 years. How will you respond to competition (Angular, Polymer, Atscript etc.)"

"Great work in the past 15 months! Keep that speed & community building efforts!"

"More focus on first time users"

"Great work! Continue!"

"You rock!"

6.2 Feedback on the survey

We also received lots of feedback on the survey itself and will improve the next one, just like this survey was improved based on your feedback from the previous one. The most noticeable feedback was some missing countries in the country selector (sorry!) and that the survey this year was considerably shorter than last year which was taken positively.

Share these results with your social network by linking to
<http://vaadin.com/gwt/report-2015>

*"Get rocks. Keep it coming.
Thanks."*

*"It's really great that you gather
feedback - please continue
to do that!"*

"very fantastic job"

*"Thank for asking / taking our
feedback, that will help to
improve GWT in future releases."*

*"Good initiative - I always look
forward to the results!"*

Section 7:

Rotten tomatoes and bacon covered rockets

Having run the survey for three years now, this is always the most interesting read. You in the GWT community have poured your hearts out in these sections and the passion you have for your framework is graspable in the comments. This year we decided to create tag clouds out of the comments and pick some of the most interesting ones out from the crowd.



7.1 The Bright side of GWT

"Thank for asking / taking our feedback, that will help to improve GWT in future releases."

"Being able to write code in a language well supported by the IDE."

"Allowing our 13-year old codebase to run on desktop, Android, iOS, ChromeOS"

"Avoids the unstructured mess that is easy to fall into with JavaScript for single page apps"

"Being able of developing web applications in pure Java"

"Optimized JavaScript Compiling from Java"

"Program in Java - compile to JavaScript."

"idiot-proof (java + cross-browser)"

checks avoid typesafe advantages js client
compiler cross apps sharing projects
type pure language ajax super
abstraction awesome css better rpc build
ability cross-browser codebase allows
components javascript applications
based GUI debugging
full everywhere compatibility design
create generation devmode learn java
able easy ecosystem eclipse model
development Oop gwt client-side
fast feature front frontend reuse browser
optimization front integration reuse
page powerful write ui code lot
refactoring performance rich ui code love
maintain structured runs web programming
tools structured runs web programming
support ide single strong uibinder
work static dev widgets server

7.2 The Dark side of GWT

"Full integration with Polymer and web-components."

"UIBinder. Needs a good WYSIWYG editor. Overly complicated learning curve."

"no java8 (yet), no clean interaction with native (yet), no fast compile times (yet)"

"IDE integration (specially debug)"

"coooooooooompiiiiiilaaaaaaaa-tioooooooooon time"

"I would add a GWT to Android compiler and I would like to use Groovy with GWT"

"Integration with Maven."

"I get a lot of programming news from reddit (see r/programming, r/java, etc.) but r/gwt is not very active. I wish there would be more activity."

"accelerate the evolution, faster releases, more manpower behind GWT"

"Change nothing."

code application provide java element layout
needs model difficult lack gui editor
api missing google better browser mobile
based large build classes complex curve
app change debugging create community
dev designer client css hard gwt maven
example development issues html
features native devmode components support
compilation nice page javascript
documentation test integration super
ide improve load ui performance widgets
mode poor implementation super
project files plugin verbose style slow uibinder
simple source rpc server superdevmode work tools
sdm libraries product speed web eclipse
framework getting learning js times

7.3 The longest constructive feedback for the survey was of 1083 characters. We think it deserves a page of its own.

Thank you Architect from Belgium in a 200 person size company.

MVP/editor framework is too limited, and oriented towards CRUD-applications, object-graphs that need to be sent back and fro between client and server. Absolute unquestionable need for some kind of Observable Model or Reactiveness (*we used <https://github.com/chris-martin/reactive-gwt>*).

Editor framework is not perceived as 'simple' at all. Too much focus on RequestFactory. Very limited and incomplete once you start using it without RequestFactory.

EventBus api, due to the many historical versions and api-changes, is confusing and messy. Needing to de-register handler-registrations is cumbersome and opens possibility for memory leaks.

"We need more/better REST-enabler api's; stuff like Promise-s etc; RestyGWT is okay but limited, but have a look at Traverson etc for inspiration. We abandoned GWT-RPC because it is so closed/obfuscated and hard to (*stress/load-*)test. Moreover, it's 'just' RPC, where REST goes much further (*we have a Spring REST HATEOAS backend which serves us very well, but the front-end api to consume this API is homebrew, based on JQuery's Ajax class*)."



With over three years of data gathering and over 3000 individual answers from the GWT community, we're pretty sure this is the most complete survey on the GWT community ever done – maybe even all other web framework communities. If you like what you see, please share it by linking to vaadin.com/gwt/report-2015/ in your social networks.

Read more about Vaadin's role in GWT at vaadin.com/gwt