The Forgotten Many? A Survey of Modern Web Development Practices

Moira C. Norrie, Linda Di Geronimo, Alfonso Murolo, and Michael Nebeling

Department of Computer Science, ETH Zurich
CH-8092 Zurich, Switzerland
{norrie,lindad,amurolo,nebeling}@inf.ethz.ch

Abstract. With an estimated 21.9% of the top 10 million web sites running on WordPress, a significant proportion of the web development community consists of WordPress developers. We report on a survey that was carried out to gain a better understanding of the profile of these developers and their web development practices. The first two parts of the survey on the background and development practices were not exclusive to WordPress developers and therefore provide insight into general web developer profiles and methods, while the third part focussed on WordPress specifics such as theme development. We present the results of the survey along with a discussion of implications for web engineering research.

Keywords: web engineering, web development practices, WordPress developers.

1 Introduction

Second-generation content management systems (CMS) such as WordPress[1] and Drupal[2] are based on a crowdsourcing model where vast developer communities share themes and plugins. It is possible for endusers to create a web site without any programming effort by selecting an existing theme and adding content, even adding or customising the functionality through the user interface. At the same time, developers with programming skills and knowledge of the platform can create or edit PHP templates, CSS stylesheets and JavaScript functions to extend the functionality or create their own themes and/or plugins.

The availability of these platforms has radically changed the web development landscape with estimates that 21.9% of the top 10 million web sites are running on WordPress which has 60.3% of the CMS market share[3]. While many sites running on WordPress are personal web sites, the platform also supports everything from web sites created by professional designers for small businesses

[1] http://www.wordpress.com, http://www.wordpress.org

[2] http://www.drupal.org

[3] http://w3techs.com/technologies/overview/content_management/all (10.4.2014).

S. Casteleyn, G. Rossi, and M. Winckler (Eds.): ICWE 2014, LNCS 8541, pp. 290–307, 2014.
© Springer International Publishing Switzerland 2014

to large, complex sites created by teams of developers. WordPress has gone well beyond its origins as a blogging platform and its web sites include popular online newspapers, e.g. Metro UK[4], as well as e-commerce sites, e.g. LK Bennett[5].

Yet, WordPress and its developers have received little attention within the web engineering research community. Information gleaned from books about WordPress, online articles and forums as well as talking to personal contacts, suggests that many WordPress web sites are developed by individuals with a mix of technical and design skills. Books on developing WordPress themes such as [1] propose an *interface-driven* approach where the main steps are to develop a mockup of the interface, add client-side functionality and then migrate to the WordPress platform. This contrasts with the *model-driven* approaches [2] widely promoted within the web engineering research community.

Since WordPress developers form a significant part of the development community, we think it is important to get a better understanding of their development practices with a view towards identifying requirements and research challenges. We therefore decided to carry out a survey of web development practices which, although not exclusively limited to WordPress developers, made efforts to reach out to this community.

The results of our survey show that there is a need to support alternative methods to model-driven web engineering that are more in line with widely-used interface-driven practices and can be integrated with platforms such as WordPress. Further, since many developers seek inspiration from existing web sites and frequently reuse elements of design and implementation from other projects, a major issue is how to provide better support for reuse in all aspects of web engineering.

In Sect. 2, we discuss the background to this work including previous surveys of web development practices. Section 3 provides details of our survey and how it was carried out. The results are presented in three sections. Section 4 reports on results related to developer profiles in terms of experience, educational background and the size of team and organisation for which they work. Results on general methods and tools used in development are then presented in Sect. 5. The third part of the survey was specific to WordPress developers and we report on the results for this part in Sect. 6. Implications for web engineering research are discussed in Sect. 7, while concluding remarks are given in Sect. 8.

2 Background

The discipline of web engineering emerged in the late 1990s with calls for systematic methods for the development of web applications, e.g. [3,4]. This in turn led to the first of the ICWE series of conferences in 2001 and the appearance of the Journal of Web Engineering (JWE) in 2002. A position paper [5] in the first issue of JWE defined web engineering as "the application of systematic, disciplined and quantifiable approaches to the development, operation and maintenance of

[4] http://metro.co.uk

[5] http://www.lkbennett.com

web-based applications". The paper presented the characteristics of both simple and advanced web-based systems, discussing how the development of such systems differed from traditional software engineering. The authors concluded that "web engineering at this stage is a moving target since web technologies are constantly evolving, making new types of applications possible, which in turn may require innovations in how they are built, deployed and maintained."

It is certainly true that both web technologies and the kinds of web-based applications in everyday use have changed dramatically over the last decade. Further, the emergence of second-generation CMS such as WordPress which offer powerful platforms for both the development and operation of all kinds of web sites has also changed how a significant proportion of web sites are built, deployed and maintained. By offering a WordPress hosting platform[6], it is even possible for endusers to literally create and deploy a web site in a few clicks. Meanwhile, developers with technical skills and knowledge of the WordPress model can develop both plugins and themes offering rich functionality for their own use and to share with others.

Examining the research literature in web engineering over the past decade reveals less radical changes in proposals for how web sites should be developed. Model-driven approaches such as OOHDM [6], UWE [7], WebML [8] and WSDM [9] were introduced in the 1990s and early 2000s. Many of these still prevail although the modelling languages may have been extended to cater for new kinds of technologies and applications. For example, the web modelling language WebML has been extended to cater for service-enabled applications [10] and context-awareness [11]. The continued emphasis on model-driven approaches may be due to the fact that the main focus still appears to be on development within, or for, large enterprises using multi-disciplinary development teams involving programmers, database architects and graphic designers. In such settings, it might be expected that the model-driven approaches widely used in software engineering and information systems would be familiar to both programmers and database architects and hence adaptations for web engineering would be more likely to be adopted. However, it is interesting to note that in a recent paper analysing model-driven web engineering methodologies [2], they comment on the fact that model-driven web engineering approaches have still not been widely adopted and they accredit this mainly to the lack of tools.

There is little recent research literature reporting on modern web development practices, especially concerning the use of platforms such as WordPress. A number of surveys were carried out in the early 2000s in conjunction with the call for web engineering to be established as a discipline. Barry and Lang [12] reported on a study in Ireland on multimedia software development methods, which included web-based information systems. Almost a quarter (24.6%) reported that they did not use a methodology while the rest stated that they used an in-house variant, with most using what the researchers considered as outdated methods and only 6.2% using UML. Reasons given for not using methodologies were that

[6] `http://www.wordpress.com`

they were “too cumbersone”, “not suited to the real world” or “long training is required”.

Taylor et al. [13] carried out a study of web development activities in 25 UK organisations based on interviews. They found that few formalised techniques were used and most “web site development activities appeared to be undertaken in an ad hoc manner” with only 8 of the 25 using design techniques such as hierarchy charts, flowcharts and storyboards. They reported little or no use of established software development techniques. Around the same time, McDonald and Welland [14] carried out a study of web development practices based on in-depth interviews, in this case involving 9 UK organisations. Only 7 of the 15 interviewees claimed to have a development process in place, with only 2 of these 7 using industry standard software development processes. Although the majority of interviewees were using prototyping or user-centred design techniques, none of them mentioned involving endusers in validating the success of a project.

More recently, El Sheikh and Tarawneh [15] reported on a survey of web engineering practices in small Jordanian companies. The results of their study showed that many developers had 5 or fewer years of software experience and that the development processes were still mainly ad hoc, with little application of established web practices.

We wanted to find out how much the situation has changed over the years in terms of the profile of web developers and also the methods used. In particular, we were interested in the community of WordPress developers and whether their backgrounds, work settings and methods differ significantly from developers that use some form of web development framework rather than a CMS as the basis for their implementation.

3 Survey

The survey was designed to address both web developers and designers including those specifically developing with and for WordPress. We designed a questionnaire consisting of 31 questions distributed over three parts: *background*, *development practices* and *WordPress development*. We used a mix of 5-point Likert-scale questions for frequency-based answers or where agreement with different statements was to be expressed as well as open-ended questions.

The first part collected demographics by asking participants to provide their age, gender and country of residence and origin. We also enquired about any formal qualifications in computer science, design and web development. Other questions addressed the participant’s professional background and experience. We asked for the number of years working as a professional web developer, as well as the size of both the organisation they work for and their web development team. Participants were also encouraged to share any recent projects they developed and their role and specific contributions to the projects. These questions together enabled us to determine developer profiles that we will report in the next section.

The second part concerned their development practices with the goal of finding out about particular methods and tools used by participants. This part started with a question on how much they look at existing web sites for inspiration in the beginning of a project. Participants were asked how often they start by modifying an existing web site or theme as opposed to creating a new one from scratch. This was followed by questions on the use of sketching and digital mockups as well as the modelling of data and functional requirements. Participants were also asked to list any tools used for creating mockups and for modelling. These questions enabled us to better assess current development practices and identify trends between different groups of developers.

We also included a question on the reuse of resources published by other developers in terms of design or layout (HTML, CSS, etc.) and functionality (JavaScript, PHP, etc.). The goal was to get a better understanding of how different types of developers work and whether and how they make use of existing resources and material provided by other developers.

The second part closed with a question on the use of CMS such as WordPress or Drupal as opposed to web development frameworks as the starting point for web development. Participants were also asked to list the specific CMS and frameworks that they use. The answers to these questions were used to classify developers based on the software tools they typically use as the basis for development. These classifications were then used for comparison purposes in the analysis of other results.

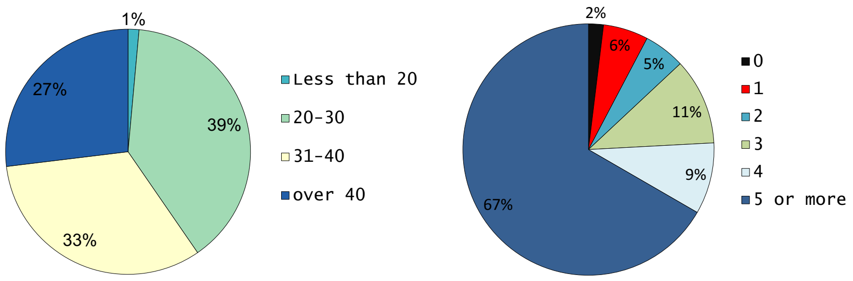
Finally, the third part specifically dealt with WordPress development. Only participants indicating that they were WordPress developers were asked to complete this part of the survey. We asked participants whether they mainly use an existing theme, modify an existing theme, create a child theme or create their own theme from scratch when they create a web site using WordPress. These questions tried to characterise the role of themes as one of the main concepts supporting reuse in WordPress.

The last set of questions allowed us to further profile WordPress developers and identify their specific needs and requirements. We asked how often they reuse code from previous WordPress projects and find themselves in the situation that they would like to mix parts of two or more themes. As before, we again distinguished between layout/style and functionality for mixing and matching parts. Finally, participants were asked to indicate the need for more customisation options of WordPress themes and which additional features they would like to see added in future versions of WordPress to support theme development.

Before starting the online survey, we first asked members of our research group to fill it in and provide feedback on the design of the questionnaire. This allowed us to fix minor issues in the phrasing of some questions and calibrate the time typically required to answer all questions which was around 10 minutes. For dissemination, we primarily recruited via Twitter, reaching out to members of the web design and development community as well as the WordPress community, asking them to contribute to our survey and retweet our request for participation with a link to the online questionnaire. Targeted Twitter users ranged from users who frequently post and retweet links to articles related to web design and development to organisers of WordPress Meetup groups, giving us access to a network of several thousand followers of these active Twitter users. We also used Facebook and Reddit as well as directly contacting web developers known to us personally via email. Between January and February 2014, the survey was accessed 622 times and we received 208 complete responses that we included in the following analysis.

4 Developer Profiles

The 208 participants (83% male, 17% female) were from 24 different countries, with the majority living in the USA (49), Switzerland (45), Germany (39) or the UK (22). The age groups are shown in Fig. 1a and the years of professional web development in Fig. 1b.



(a) Age

(b) Years of experience

Fig. 1. Age and experience of participants

It is interesting to note that we had good coverage of the different age groups and the majority of our participants (67%) had 5 or more more years of experience as a professional developer. This contrasts with the survey of El Sheikh and Tarawneh [15] where 63% had 5 or fewer years of software experience.

Since one of the aims of our survey was to compare the profiles and methods of developers using a CMS as their main development platform with those using web development frameworks such as Django[7], Ruby on Rails[8] and Bootstrap[9], we asked participants how often they use each of these approaches.

The results in Fig. 2 show that the CMS developers are more likely to stick with this approach as 39% of participants answered that they always use this approach while only 18% always use a development framework. It is important

[7] http://www.djangoproject.com

[8] http://rubyonrails.org

[9] http://getboostrap.com

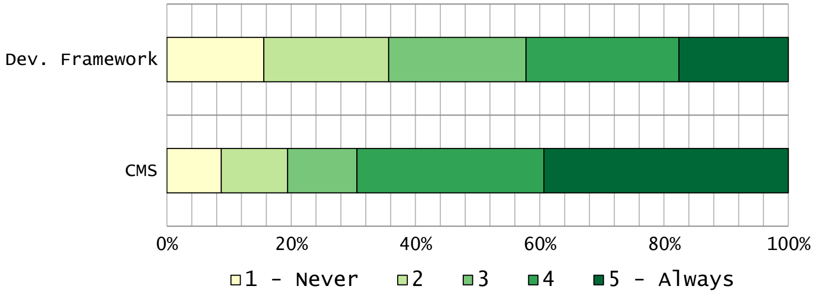


Fig. 2. Use of CMS or development framework

to note that these are not disjoint communities and 9% said that they always use both. 53% of the participants classified themselves as WordPress developers and 47% did not. Since we made efforts to target WordPress developers, it is not surprising that the majority were in this category, but we also achieved our aim to have a good mix of WordPress and non-WordPress developers. We note that some developers listed WordPress as a CMS that they use, but answered ‘No’ to the question asking if they are a WordPress developer. One reason for this might be that they interpreted the question as whether they are involved in developing the WordPress platform rather than whether they use it for developing applications. Another explanation could be that they classify themselves as endusers rather than developers since they create applications using the platform without actually doing any coding.

We classified the participants into three disjoint categories: those who answered ‘Yes’ to the question asking if they are a WordPress developer (WP), those who are not in WP but answered in the range 3-5 (sometimes to always) when asked if they use a CMS (CMS) and those who are not in WP and answered 1 or 2 (rarely or never) when asked if they use a CMS (Other). Thus developers who mostly use Drupal would be in the CMS category, while those who mainly use a web development framework and only occasionally use a CMS would be in the Other category. The sample sizes of each category are 111 (WP), 62 (CMS) and 35 (Other).

We asked participants how they would classify themselves in terms of whether they are designers, developers or both. Figure 3 reveals that, in all three categories, a significant proportion classified themselves as half-designer/half-developer (WP: 40%, CMS: 29%, Other: 34%), but there was also a significant proportion who classified themselves as ‘developer’ or ‘mainly developer’ (WP: 48%, CMS: 48%, Other: 63%). Since we were mainly targeting developer communities rather than design communities, we did not expect many participants to classify themselves as ‘designer’ or ‘mainly designer’. Nevertheless, this shows that, rather than considering themselves as pure developers, many web development practitioners would see themselves as a mix of web developer and web designer.

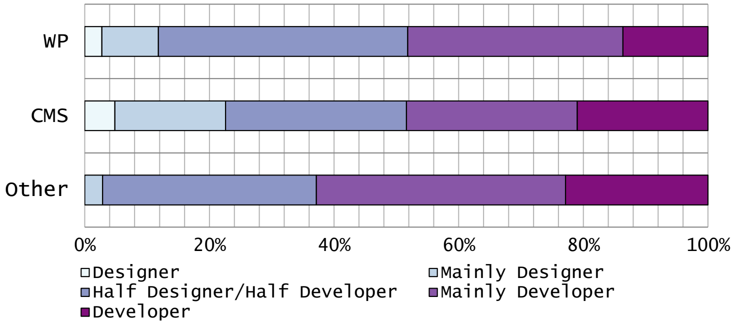


Fig. 3. Designer and/or developer

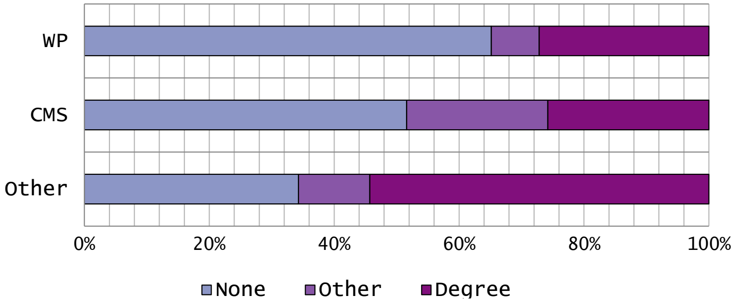


Fig. 4. Computer Science education

The educational background of the three categories is shown in Fig. 4 and 5. In all three categories, a significant proportion of participants have no formal qualification in computer science (WP: 65%, CMS: 52%, Other: 34%). Although we targeted developer communities rather than design communities, we also asked what, if any, qualification the participants have in design. As might be expected, relatively few have any formal qualification in design (WP: 31%, CMS: 25%, Other: 24%), although a significant proportion in each of the three categories classified themselves as half-designer/half-developer.

42% of all participants have no qualification in either computer science or design. In the case of participants who classified themselves as half-designer/half-developer, we also had 42% with no qualification in computer science or design. We also asked participants whether they have any kind of qualification in web development or specific web technologies. Taking this information into account, we still had 38% of participants with no formal education in computer science, design or web development.

Next, participants were asked to indicate the size of the organisation for which they work and also the size of their project team. The results shown in Fig. 6 indicate that a significant proportion of WordPress developers are either self-employed (42%) or belong to organisations with 5 or fewer employees (16%). On the other hand, it also shows that WordPress is not solely used by

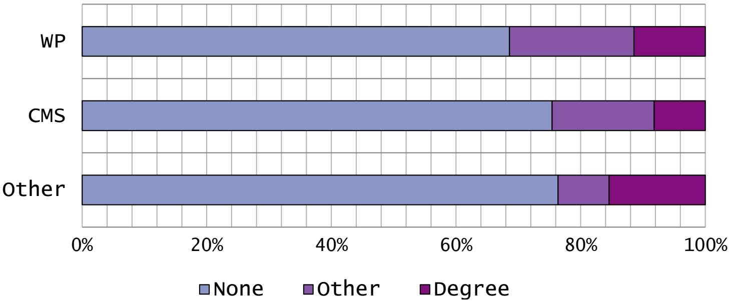


Fig. 5. Design education

individuals and small businesses since 10% of WordPress developers are working in organisations with more than 250 employees. While a significant proportion of non-CMS developers are also self-employed (14%) or in organisations with 5 or fewer employees (9%), the proportion working in organisations with more than 50 employees (51%) is far greater than for WordPress (20%).

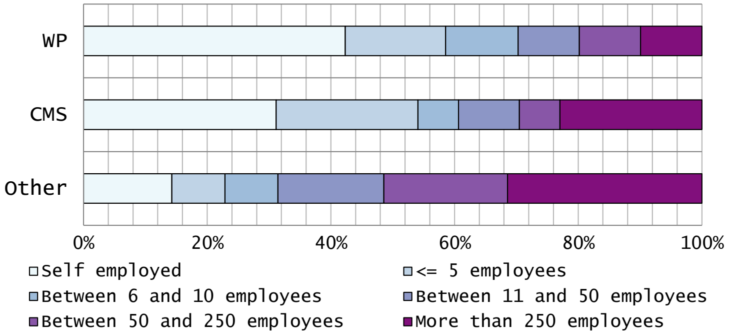


Fig. 6. Size of organisation

Previous surveys have tended to target organisations rather than individuals, and therefore have not involved developers who are self-employed. The smallest organisation involved in the survey by Taylor et al [13] had 20 employees. While the survey of El Sheikh and Tarawneh [15] targeted small companies in Jordan and 75% of companies had fewer than 10 employees, they also did not include self-employed developers.

Since many of our participants are self-employed or working in organisations with 5 or fewer members, clearly these developers either work alone or in very small teams and therefore the percentages for ‘no team’ and a team size of ‘5 or fewer’ would be expected to reflect this. Still, even in larger organisations, participants often work in small teams and 75% of WordPress developers work in teams with 5 or fewer members, with only 7% working in teams with more than 10 members. In the case of the non-CMS participants (Other), team sizes still tend to be small with 51% working in teams with 5 or fewer members, but 26% of them do work in teams with more than 10 members.

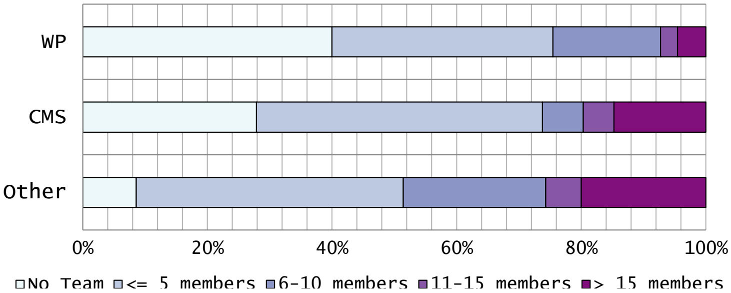


Fig. 7. Size of team

McDonald and Welland [14] estimated the average team size of web development projects in the organisations that they surveyed as 6. They argued that the small size of web development teams is one of the major differences to traditional software development teams, citing an article published in 2000 by Reifer [16] where he estimated the size of web development teams as 3-5 compared with traditional software development projects with hundreds of team members.

Interestingly, a survey of 200 Java developers carried out in 2011 by Munoz[10] reported that 40.7% worked in a team size of 1-5, 26.6% in a team size of 5-10 and 32.6% in teams larger than 10. These figures are actually not so different from the Other group where 52% work in teams of 1-5, 23% work in teams of 6-10 and 26% in teams larger than 10.

5 Methods and Tools

In this section, we report on the second part of the survey where all participants were asked questions about the methods and tools that they use in development projects. We started by asking them if they use existing web sites for inspiration at the beginning of a new project.

As shown in Fig. 8, more than 20% always look at examples of web sites for inspiration (WP: 23%, CMS: 24%, Other: 30%) and more than 50% answered 4 or 5 indicating that they often inspect examples (WP: 53%, CMS: 54%, Other: 67%). Although WordPress explicitly supports design-by-example through its notion of themes that can be easily accessed and previewed in online galleries, it is interesting to note that examples are used as much, if not more, in the Other group.

[10] `http://www.antelink.com/blog/software-developer-survey-first-chapter.html`

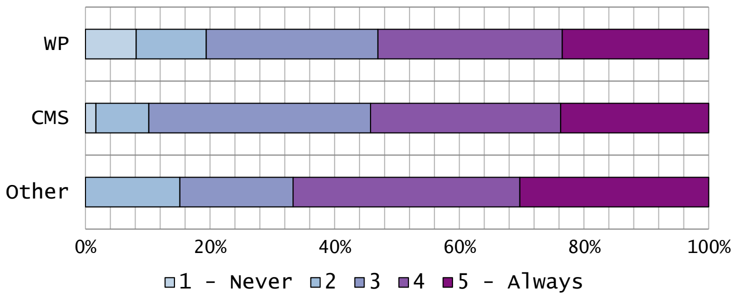


Fig. 8. Use of other web sites for inspiration

The survey by Taylor et al [13] published in 2002 also noted that examples of other web sites were often used for inspiration: “Roughly a third of those interviewed across 25 organisations studied indicated that they used other organisations’ websites for design ideas in order to supplement their website design activities”.

We also asked how often developers create a website or theme based on the modification of an existing web site or theme, either of their own or of another developer. As shown in Fig. 9, 61% of WordPress developers answered that they sometimes, often or always base the design of a web site or theme on an existing web site or theme. While none of the Other group answered that they always base a new design on an existing one, 50% said that they do this sometimes or often.

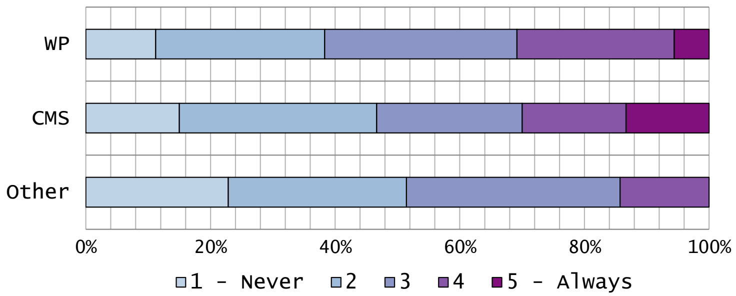


Fig. 9. Modify existing web sites or themes

Participants were asked how often they sketch mockups or create digital mockups and the results are shown in Fig. 10 and 11.

It is clear that sketching plays an important role with 43% of WordPress developers and 39% of the CMS group saying that they usually or always sketch. Sketching is used even more in the Other group with 57% stating that they usually or always sketch.

It is also common to produce digital mockups with more than 45% of the WordPress developers, 56% of the CMS group and 57% of the Other group answering 4 or 5 to indicate that they often or always use them. A range of tools

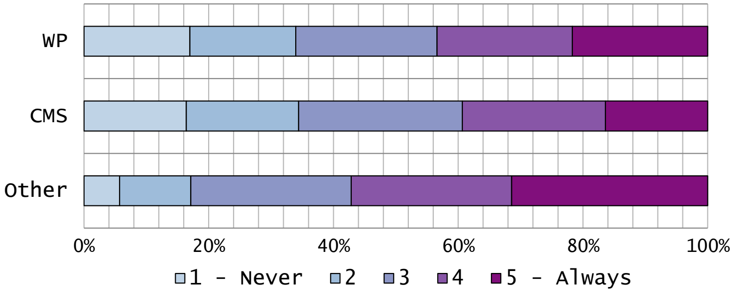


Fig. 10. Sketching mockups

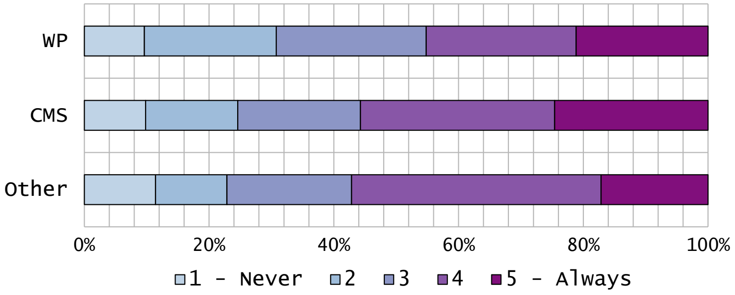


Fig. 11. Digital mockups

were listed including graphics editing tools such as Adobe Photoshop, diagram editors such as Microsoft Visio and wireframing tools such as Balsamiq[11].

Some developers wrote that they do not sketch or create digital mockups because they have a pure development role and implement the mockups produced by a graphic designer.

Since model-driven web engineering is widely promoted in the research community, we were interested in how frequently data and functional requirements are modelled. The results are shown in Fig. 12 and Fig. 13, respectively.

While only 28% of WordPress developers answered 4 or 5 to indicate that they often or always model data, 52% of Other developers answered that they often or always model data. The percentages answering that they often or always model functional requirements were also higher in the Other group (60%) compared to the WordPress developers (42%).

26% of participants answered that they never model data or functional requirements, leaving 74% who indicated that they use modelling at least some of the time. However, further analysis of the written comments provided by participants showed that the figures presented in Fig. 12 and 13 are very misleading as, in many cases, the participants had no idea what was meant by "modelling data" or "modelling functional requirements". We asked participants to list tools

[11] http://balsamiq.com

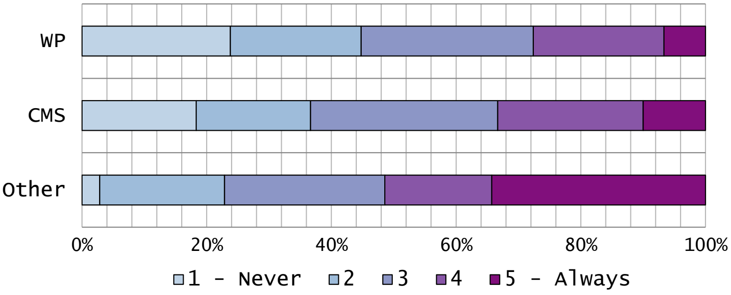


Fig. 12. Modelling data

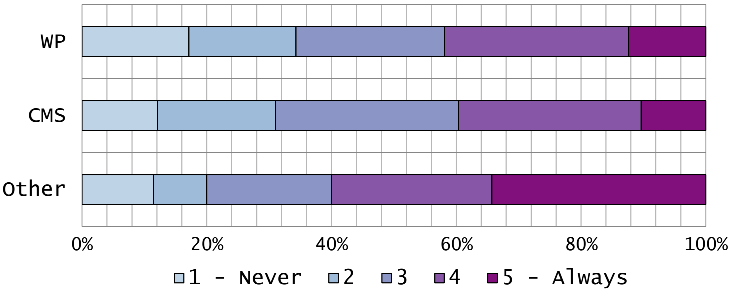


Fig. 13. Modelling functional requirements

that they use for modelling data and/or functional requirements and answers included "text documents", "spreadsheets and/or code editors", "Django to create prototypes" and "WordPress". One participant wrote "Not sure if I misunderstand this, but I usually just write requirements out—paper, text edit, google doc spreadsheets etc." Some listed project management tools and one participant even wrote something about testing and deployment. Only 11% of all participants listed an application or suite of tools that provides support for data or functional modelling. A further 5% wrote something general such as "paper and pen" or "whiteboard" that could also be considered as tools for modelling. This suggests that the number of developers actually doing some form of modelling of data or functional requirements is well below the figures reported.

This leads us to conclude that many of the participants are not even aware of software engineering practices, let alone applying them in even an informal way. This could be a consequence of the fact that a significant proportion of participants (WP 65%, CMS 52%, Other 34%) have no formal education in computer science.

6 WordPress Development Practices

The third part of the survey was only for WordPress developers as it deals specifically with the development of WordPress themes. A theme is a set of

PHP templates, CSS stylesheets and media objects that define the structure, navigation, functionality and presentation of a web site. The media objects included in a theme are generally static images used in the presentation of a web site such as the arrows used in sliders, buttons used in navigation and images that appear in the header. Endusers can select a theme from a gallery and create their own web site by simply adding content. A theme can also have a number of associated parameters to make it customisable through the general administrative interface. A professional developer will typically develop a theme to meet the requirements of a client, but they may also develop a theme for a particular class of clients such as restaurants, photographers or professional societies and make it customisable to the needs of a specific client.

The questions in this part of the survey were designed to find out more about how developers generate themes and specifically the forms of reuse that they employ or would like to have supported. Figure 14 presents an overview of the answers to a set of questions asking if and how they develop new themes for a specific project.

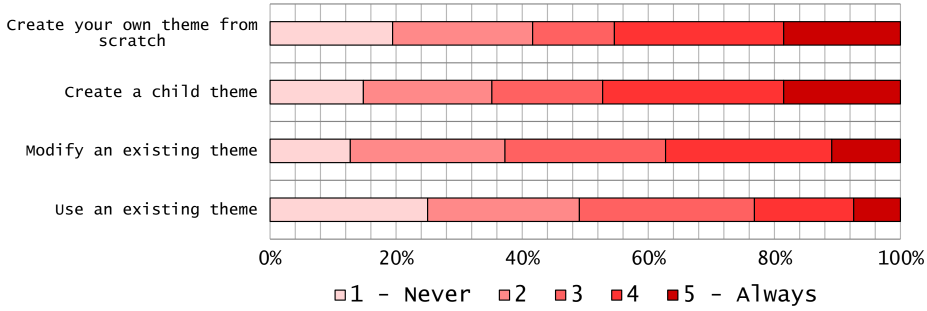


Fig. 14. Developing themes

7% of developers answered that they always use an existing theme. This means that these developers simply select a theme already provided by another developer and customise it for a client. This could involve the design of logos and other presentation features as well as the choice of layout, navigation and content.

19% of developers indicated that they always develop their own themes from scratch while 19% specified that they never do this. A developer can create a new theme based on an existing theme. This can be done by formally creating a child theme, but often developers will simply modify the PHP templates and CSS stylesheets provided. The results show that it is common for developers to build on existing themes using either of these approaches. 47% indicated that they often or always create a child theme of an existing theme for a project, while 37% answered that they often or always create a theme for a project by modifying an existing theme.

Since a theme defines an entire web site, a developer can only select a single theme as the starting point for a web site which means that they support all-or-nothing reuse. Once a theme has been selected, reuse of features from other themes can only be done by copying pieces of code and making any necessary modifications to integrate it into the theme under development.

We asked developers how often they find themselves in the situation where they would like to be able to mix parts of two or more existing themes. In the questions, we differentiated between the reuse of layout as specified by HTML and CSS and the reuse of functionality which could be either PHP code or JavaScript.

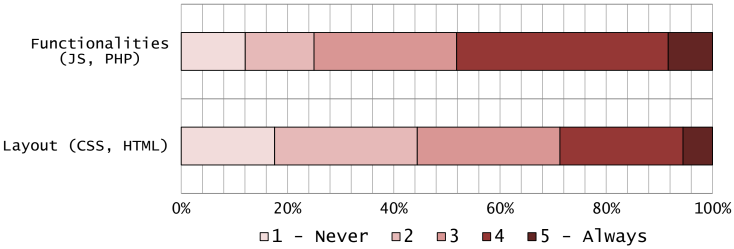


Fig. 15. Desire to be able to mix themes

Only 12% said they never find themselves in the situation where they would like to mix functionality from different themes, while 18% said they never want to mix layout. 75% answered 3-5 indicating that they sometimes, often or always find themselves wanting to mix functionality, while 56% answered 3-5 for layout.

We included a question asking participants to list what, if any, features they would like to see added to WordPress to support theme development. Most participants left this empty and the suggestions covered a range of issues from better means of managing media to easier ways of handling custom post types. One participant wrote: “I think where WordPress needs to go is to click-and-play development. Get rid of the need to code and it will take over the Internet”. This comment can be interpreted as a request that it should go further in its support of enduser development.

7 Discussion

The results of our survey confirmed our impression that a significant proportion (40%) of WordPress developers work alone and act as both designer and developer. Since previous studies targeted organisations and tended to omit self-employed developers, it is impossible to say whether this is an increasing trend. However, the tendency for web developers to work in small teams as reported in earlier surveys is still the case, with 75% of WordPress developers working in teams with 1–5 members and only 26% of non-CMS developers working in teams with more than 10 members.

The fact that a significant proportion (40%) of both WordPress and non-CMS developers classified themselves as half-designer/half-developer, taken together with the fact that 41% of participants in this category have no qualification in either computer science or design, suggests that many of these developers have a mix of some design skills and some technical skills. Without a formal education in computer science and working alone or in very small organisations where there is likely to be a lack of in-house training, it may well be the case that many of these developers are not aware of modern software engineering methods, let alone using them. This would certainly be suggested by the answers that we received to our question about the tools that they use for the modelling of data and/or functional requirements. It is interesting to compare this with the results of the survey of Java developers carried out in 2011 by Munoz[12] where he reported that almost all participants had either a Bachelor or Masters degree in Computer Science.

This raises the question of whether efforts to adapt and promote software engineering methods, and specifically model-driven approaches, for web engineering are ever likely to have an impact in the web development community at large. Not only are many of these developers unaware of the underlying principles and techniques as well as the details of the methods, but many CMS developers have good reasons to employ interface-driven approaches rather than model-driven approaches. Therefore, while model-driven approaches may have their place in larger enterprises, we believe that the research community should also be exploring alternative methods that target practitioners at large.

One of the key findings of our study is how much developers build on the work of other designers and developers in their projects. This includes everything from using examples of other web sites for inspiration down to the detailed reuse of code. At the moment, there is little engineering support for reuse in CMS other than the concept of themes which support all-or-nothing reuse. Even the concept of child themes which is intended to provide a controlled way of developers building on existing themes is frequently not used and themes modified directly instead.

Within the web engineering research community, support for reuse has mainly been at the level of services. For example, WebComposition [17] allows applications to be built through hierarchical compositions of reusable application components. There has also been a lot of research in the area of web mashups to allow applications to be created through compositions of existing web sites, e.g. [18,19]. While this research is certainly relevant, the focus is purely on reuse rather on the design and development of new web sites as a whole and, as far as we know, there has been no attempt yet to adapt or integrate these methods into platforms such a WordPress. It is however important to mention that the work on mashups is also significant within the web engineering research community in its efforts to support enduser development.

Some researchers within the HCI community advocate a design-by-example approach [20,21] where the focus is very much on the reuse of the design aspects

[12] `http://www.antelink.com/blog/software-developer-survey-first-chapter.html`

of a web site. The idea is to allow users with little or no technical knowledge to develop their web site by selecting and combining elements of example web sites accessed in galleries. While the results of their studies are promising, they only deal with static elements and have not addressed the technical challenges of extracting and reusing functionality.

We believe that design-by-example is a promising paradigm worthy of detailed investigation within the web engineering research community. It is compatible with the interface-driven approaches that are currently in widespread use where mockups lead to prototypes that are gradually refined and migrated to platforms such as WordPress. With this goal in mind, we have started investigating how design-by-example could be supported in WordPress so that users could design and develop a fully functioning web site by selecting and reusing components of existing themes [22].

8 Conclusion

With a view to providing an insight into modern web practices, especially among the vast communities of WordPress developers, we have reported on the results of an online survey involving 208 participants working with CMS and/or web development frameworks. Unlike many previous surveys, we were keen to reach out to self-employed developers as well as developers within larger organisations and this we achieved.

The results point to the need for alternatives to model-driven approaches with a stronger focus on interface-driven development and enduser tools suited to the large numbers of developers with a lack of formal education in computer science and a mix of design and technical skills. Further, there is a need for methods that support the reuse of all aspects of web engineering and can be integrated into platforms such as WordPress that already have a significant proportion of the CMS market share and are continuing to grow.

Acknowledgements. We acknowledge the support of the Swiss National Science Foundation who financially supported this research under project FZFSP0_147257.

References

1. Silver, T.B., McCollin, R.: WordPress Theme Development - Beginner's Guide. Packt Publishing (2013)
2. Aragon, G., Escalona, M.-J., Lang, M., Hilera, J.R.: An Analysis of Model-Driven Web Engineering Methodologies. International Journal of Innovative Computing, Information and Control 9(1) (2013)
3. Coda, F., Ghezzi, G., Vigna, G., Garzotto, F.: Towards a Software Engineering Approach to Web Site Development. In: Proc. 9th Intl. Workshop on Software Specification and Design (1998)
4. Gellersen, H.W., Gaedke, M.: Object-Oriented Web Application Development. IEEE Internet Computing 3(1) (1999)

5. Deshpande, Y., Murugesan, S., Ginige, A., Hansen, S., Schwabe, D., Gaedke, M., White, B.: Web Engineering. Journal of Web Engineering 1(1) (2002)
6. Schwabe, D., Rossi, G.: The Object-Oriented Hypermedia Design Model. Communications of the ACM 38(8) (1995)
7. Koch, N., Kraus, A.: The Expessive Power of UML-based Web Engineering. In: Proc. 2nd Intl. Workshop on Web-Oriented Software Technology (IWWOST) (2002)
8. Ceri, S., Fraternali, P., Bongio, A.: Web Modeling Language (WebML): A Modeling Language For Designing Web Sites. Computer Networks 33(1-6), 137–157 (2000)
9. de Troyer, O., Leune, C.: WSDM: A User-Centred Design Method for Web Sites. Computer Networks and ISDN Systems 30(1-7) (1998)
10. Brambilla, M., Ceri, S., Fraternali, P., Acerbis, R., Bongio, A.: Model-Driven Design of Service-Enabled Web Applications. In: Proc. SIGMOD Industrial (2005)
11. Ceri, S., Daniel, F., Matera, M., Facca, F.M.: Model-Driven Development of Context-Aware Web Applications. TOIT 7(1) (2007)
12. Barry, C., Lang, M.: A Survey of Multimedia and Web Development Techniques and Methodology Usage. IEEE Multimedia (April–June 2001)
13. Taylor, M.J., McWilliam, J., Forsyth, H., Wade, S.: Methodologies and Web Site Development: A Survey of Practice. Information and Software Technology (44) (2002)
14. McDonald, A., Welland, R.: Web Engineering in Practice. In: Proc. 4th WWW Workshop on Web Engineering (2001)
15. Sheikh, A.E., Tarawneh, H.: A Survey of Web Engineering Practice in Small Jordanian Web Development Firms. In: Proc. 6th Joint Meeting on European Software Engineering Conference and ACM SIGSOFT Symposium on Foundations of Software Engineering (ESEC/FSE) (2007)
16. Reifer, D.: Web Development: Estimating Quick-to-Market Software. IEEE Software (November–December 2000)
17. Gellersen, H.-W., Wicke, R., Gaedke, M.: WebComposition: An Object-Oriented Support System for the Web Engineering Lifecycle. Computer Networks 29(8-13) (1997)
18. Daniel, F., Casati, F., Benatallah, B., Shan, M.C.: Hosted Universal Composition: Models, Languages and Infrastructure in mashArt. In: Laender, A.H.F., Castano, S., Dayal, U., Casati, F., de Oliveira, J.P.M. (eds.) ER 2009. LNCS, vol. 5829, pp. 428–443. Springer, Heidelberg (2009)
19. Cappiello, C., Matera, M., Picozzi, M., Sprega, G., Barbagallo, D., Francalanci, C.: DashMash: A Mashup Environment for End User Development. In: Auer, S., Díaz, O., Papadopoulos, G.A. (eds.) ICWE 2011. LNCS, vol. 6757, pp. 152–166. Springer, Heidelberg (2011)
20. Hartmann, B., Wu, L., Collins, K., Klemmer, S.R.: Programming by a Sample: Rapidly Creating Web Applications with d.mix. In: Proc. 20th ACM User Interface Software and Technology Symposium (UIST) (2007)
21. Lee, B., Srivastava, S., Kumar, R., Brafman, R., Klemmer, S.: Designing with Interactive Example Galleries. In: Proc. Conf. on Human Factors in Computings Systems (CHI) (2010)
22. Norrie, M.C., Di Geronimo, L., Murolo, A., Nebeling, M.: X-Themes: Supporting Design-by-Example. In: Casteleyn, S., Rossi, G., Winckler, M. (eds.) ICWE 2014. LNCS, vol. 8541, Springer, Heidelberg (2014)