The Challenges of Web Engineering and Requirements for Better Tool Support

Jochen Rode[1], Mary Beth Rosson[2], Manuel A. Pérez-Quiñones[1]

[1]Virginia Polytechnic Institute and State University
Center for Human-Computer Interaction
3160 Torgersen Hall
Blacksburg, VA 24061, USA
jrode@vt.edu, perez@cs.vt.edu

[2]Pennsylvania State University
School of Information Sciences & Technology
330D IST Building
University Park, PA 16802
mrosson@ist.psu.edu

ABSTRACT

We report the experiences of semi-professional developers regarding the challenges, tools, and processes within the domain of web application development. The paper summarizes the main problems in web development, characterizes the habits of programmers and concludes with a "developer's wish list" for improvements to web technologies and tools. The report is based on two independent sources – a survey of 31 web developers and an in-depth interview study with 10 participants.

Keywords

Web application development, web development tools, web engineering

1. INTRODUCTION

Tim Berners-Lee designed the web as a collaborative tool [1]. However, his early vision was one of document sharing, and recognition of the web's potential as a platform for interactive applications has been an emergent phenomenon, with the result that much of the web's infrastructure is ill-suited for application development. Currently, development of a typical web application requires knowledge not only of traditional programming languages like Java, but also technologies and problems specific to the web, for example HTML, JavaScript, CSS, HTTP, and cross-platform, cross-browser compatibility issues. Tools that assist web developers with complex and tedious tasks are one obvious solution to the problem. In order to build better tools we need to understand the needs and habits of their users.

Motivated by a research focus on web application development by nonprogrammers ([7],[8]), we have investigated the challenges that *experienced* web developers are confronted with, when developing web sites and web applications. This is an important step in our requirements analysis of tools that might be used by individuals with no formal training in web programming. Our goals are two-fold: we expect that our findings can contribute to the ongoing development of web technologies and tools for professionals and semi-professionals, and furthermore, we want to anticipate and "hide" these problems as much as possible in our development of tools for nonprogrammers.

We have targeted our analysis at the web development activities within the academic computing community of Virginia Tech as it is a good source for web developers and diverse projects of different levels of sophistication.

We report the findings from two distinct studies – a survey and an interview study. After a brief review of related work we discuss our research methods and our major results. The survey and interview study jointly highlight key challenges such as: implementing security, cross-platform compatibility, debugging, and technology integration. We first report the findings from the survey and the interviews separately and then summarize to paint a coherent picture of the status-quo of web development.

2. RELATED WORK

The analysis of web developers' needs has received little attention in the web engineering literature. A survey conducted by Vora [10] is an exception. Vora queried web developers about the methods and tools they use, and the problems they typically encounter. Some of the key problems that developers reported included ensuring web-browser interoperability, and usability and standard-compliance of WYSIWIG editors.

Fraternali's work [2] is also related to our interests in that it proposes a taxonomy for web development tools that suggests some of the major dimensions of web development tasks.

Newman and Landay [3] investigated the process of web site development by interviewing 11 web development professionals. They found that these experts' design activities involve many informal stages and artifacts; designers employ multiple site representations to highlight different aspects of their designs; and they use many different tools to accomplish their work. They concluded that there is a need for "informal tools" that help in the early stages of design and integrate well with the tools designers already use. The research we report here also focuses to a certain extent on the nature of the design process but even more so on specific problematic issues within it. However, our participants are semi-professionals rather than expert developers.

3. METHODS & RESULTS

The research presented here was initiated with a survey of web developers that asked for ratings and examples of various web development activities. In order to enrich and explain the findings and to increase the total number of reported experiences, we later conducted in-person interviews with developers who had not participated in the survey (with the exception of one).

3.1 The Survey

The survey data analyzed here is a subset of the data collected in a survey titled "Interactive Websites". This survey had two distinct purposes. One purpose was to determine end-users' needs for web applications, the other to learn about the challenges inherent in web application development. This analysis focuses on the latter goal.

Out of a total of 67 participants, 31 responded to the section dedicated to web development challenges. The survey specifically asked participants only to respond if they had previously developed an interactive web site (a.k.a. web application).

The web-based survey can be viewed online [4]; a general summary along with individual response data can be browsed online [5] as well.

We recruited participants via email by sending invitations to Virginia Tech faculty, staff and students who maintained an account on the university's web hosting system, administered a web site for a campus organization, or participated in Virginia Tech's 2001 Web Developers Conference. The email invitation stated the purpose of the investigation and contained a link to the web-based survey. In order to encourage participation we advertised a raffle of lunch coupons ranging from $5 to $15. The survey was open for participant input for approximately three weeks at the end of the spring semester in 2002.

On average, the 31 respondents who answered the questions about web development rated themselves just above the mid-point on a scale from 1 (no knowledge in web application development) to 5 (expert knowledge); the mean self-rating was 3.2 (SD=0.9). Their self-reported years of experience in web application development were approximately equally distributed between "*less than a year*" and "*more than 5 years.*" 19 respondents identified themselves as undergraduate students, 16 as graduate students, 13 as faculty, and 8 as alumni.
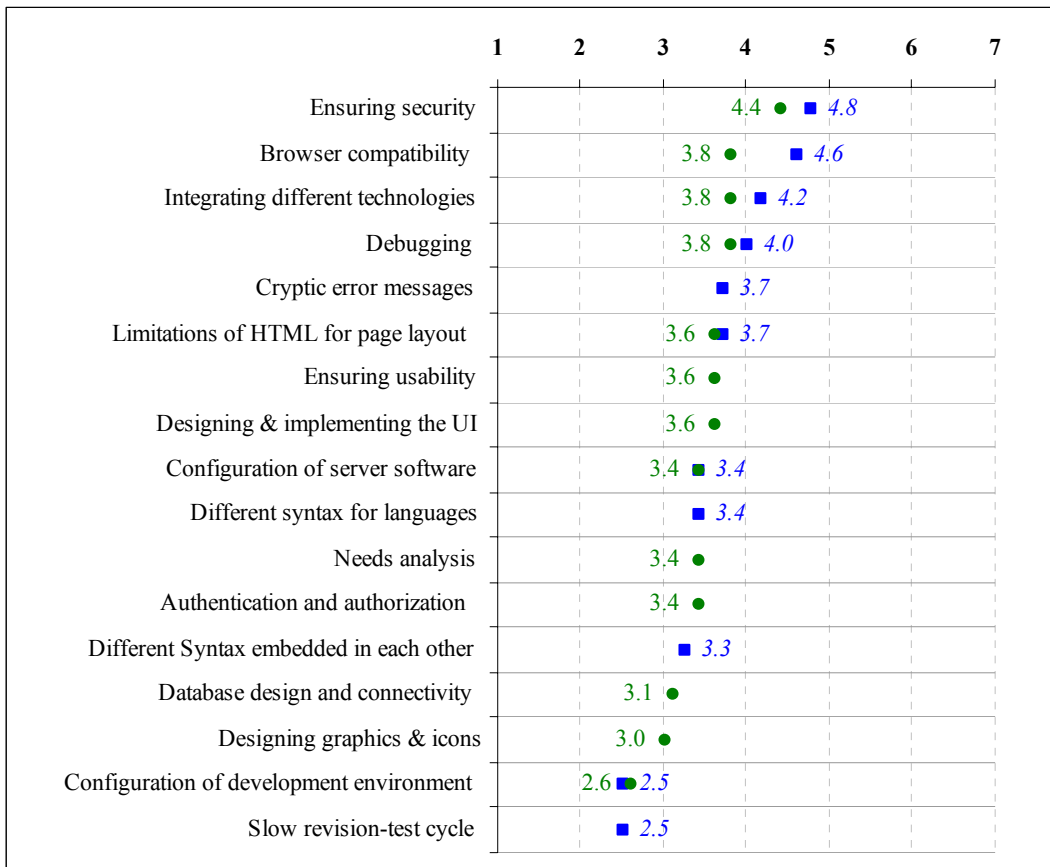
Figure 1: Responses to question about problems in web application development (1=not a problem at all; 7=severe problem). The square markers show the mean of the responses from the survey (value is right of the square marker in italics; N=31). The round markers show the mean of the responses from the pre-interview questionnaire (value is left of round marker; N=10). In order to facilitate comparison, the survey responses have been scaled from a 1-5 scale to a 1-7 scale.

With the intention of finding those issues that our respondents perceive as the biggest challenges in web development we asked them to rate a list of potential concerns on a scale from 1 to 5 (1=not a problem at all; 5=severe problem). The square markers in figure 1 show these responses (along with those from the pre-interview questionnaire).

In order to facilitate comparison, the survey responses have been scaled up to match the 1-7 scale from the pre-interview questionnaire. The (scaled-up) standard deviations vary in the range from 1.3 to 2.1. As the average ratings suggest, no one concern stood out as generally severe; most of the average ratings were in the middle or lower half of the scale. The two top issues were ensuring security and browser compatibility. This suggests that these might be particularly common problems in web development, at least for developers at an intermediate level of expertise.

Regarding "Other problems that you typically encounter during web development:" we received the following answers (number in parenthesis indicates frequency): time available for development (2), web browsers bugs (2), race conditions (2), incompatibilities between development tools, preparation of images, hard-to-find "random errors", defining business/user requirements, time to learn new versions and upgrades, concurrency conditions, lack of consulting support.

We also asked developers: "From the interactive websites that you developed consider one that was particularly challenging. What were the top 3 most challenging issues you encountered while developing this website?" The answers to this question were very diverse and we summarized them into the following problem areas (number in parenthesis indicates frequency):

- Availability & setup of development environment and production servers (5)
- User interface layout, graphics (5)
- Integration issues (5)
- Needs analysis, user feedback and education (4)
- Database design and connectivity (4)
- Available time and funding (3)
- Concurrency (3)
- Authentication and authorization (2)
- Standard compliance, browser compatibility (2)
- Limitations of the web paradigm (2)
- Others (mentioned once each): security, fault tolerance, load issues, efficiency, maintenance of service, dealing with someone else's code

Among this group of web developers, 12 (39%) reported using Microsoft FrontPage as a web design tool on a regular basis, 11 (35%) said they use Macromedia Dreamweaver, and 5 (16%) indicated that they use Macromedia Flash (the multiple choice question with "others" option allowed for multiple selections). Apart from Microsoft Notepad which was mentioned 5 times, other tools were only mentioned once or twice.

The answers to the question: "Describe your "dream" web application development tool? How would it facilitate development? Consider this question a "wish list"!" were also quite diverse (number in parenthesis indicates frequency):

- Powerful layout & graphics functionality and asset management (5)
- Easy-to-use, "reads my mind", "intelligent" (4)
- Pre-build scripts, widgets, components (4)
- Integrated toolbox that bundles everything needed for web application development (3)
- Automatic generation of clean, standard, cross-browser compatible code (3)
- Good, context-sensitive help and tips (3)
- Automatic site maintenance and reduction of tedious and redundant operations (3)
- Build-in testing and debugging tools (3)
- WYSIWIG-based with code-view option (2)
- Others (mentioned once each): clear error messages, free, changes take effect immediately, website usage tracking, tool is a native Microsoft Windows application, version control, check-in/check-out, workflow support, user has control over tool's "intelligence", website overview function, forms wizard

In summarizing the survey responses, no single issue stands out as severely problematic. However, ensuring security, integrating different technologies, debugging, and cross-browser compatibility seem to be the top problems from the perspective of our audience. Interestingly, Vora's survey [10] identified the problem of compatibility already more than 5 years ago, yet it persists. Our survey has been a rather coarse measurement tool and did not reveal any details regarding the development process and general habits of semi-professional web developers. These questions were addressed by in-person interviews which we conducted later.

3.2 The Interviews

We conducted interviews with the same target audience (semi-professionals) for two reasons. First, we wanted to better understand the details of the web development process and have the opportunity to ask further questions. Second, we wanted to increase the total number of reported experiences which is why we interviewed developers that did not participate in the survey (with one exception). Many of the questions were equal or similar to the ones asked in the survey.

We interviewed 10 web application developers in the period between May and September 2003. Out of these 8 were conducted as one-on-one interviews, the remaining 2 (due to the unavailability of the participants) as online questionnaires with follow-up email communication. The one-on-one interviews lasted about one hour. Participants were selected by contacting webmasters of various web applications on the Virginia Tech campus as well by as contacting local web development businesses.

Prior to each interview, the participant filled in an online questionnaire which was targeted at collecting quantitative information and helped us to prepare for the in-person interviews. This questionnaire [6] also contained all the main questions asked during the interview (so that participants could be mentally prepared) but participants were asked to *not* answer them online. The questionnaire was similar to, but much more detailed than the survey.

Five of the interviews were conducted at the workplace of the participants, three in our laboratory. However, the atmosphere was always private. All of these interviews were voice-recorded and later transcribed in abbreviated form.

The two participants who were not available for in-person interviews were asked to complete the questionnaire as detailed as possible online. Where necessary, we later exchanged emails to clarify and elaborate on answers.

The participants included nine males and one female. Three participants were between 26-30 years old, three participants between 31-35 years, two between 46-50 years, one participant between 21-25 years, and one participant under 21 years old.

The question "How do you rate your overall knowledge in web application development? (1=no knowledge, 7=expert knowledge):" resulted in an average of 5.1 (SD=1.3) with only two participant rating themselves below 5. The average self-reported experience of the interview participants is somewhat higher than the (scaled-up) mean experience of the survey participants which was only 4.3 (SD=1.3).

Two participants reported that they have been developing web applications for 2 years, two participants for 3 years, one for 4 years, two for 5 years and three for more than 5 years. The participants included two full-time web developers, four IT personnel who develop web applications as part of their work, one professor who teaches web application development, two students who work in this area besides their studies, and one CEO of a small-business e-retail company who has autonomously created his e-business web site.

In the first question of the interview we asked about the most challenging issues in web development without pre-defining any categories. Top answers were (the number in parenthesis indicates how many participants mentioned the concept): finding time to develop (2), debugging (2), compatibility (2), and keeping the application maintainable (2), creating an attractive user interface (2). Many more concerns were expressed in this question and throughout the interviews such as eliciting requirements, getting people to test an application, and political issues such as gaining access to data sources.

The feedback to several rating scales from the pre-interview questionnaire is shown in figure 1 and 2. Note that a 7-point scale is used throughout the interview study instead of the 5-point scale of the survey study. Figure 1 shows how the participants rate different web development concerns with regard to how problematic they are. As one would expect, the results are quite similar to the ones from the survey. The (mostly) small differences may be attributed to a rather small N of 10, to the higher level of experience of the interview participants when compared to the survey participants and perhaps to the different scales used (5 vs. 7-point scale).

Figure 2 shows the summarized responses to different questions regarding the habits of our participants. During the interviews our participants were asked to explain their answers to the ratings provided in the pre-interview questionnaire (figure 1 and 2). The responses confirm that the major concerns are security, compatibility, integration and debugging. The quote: "How do I know it's secure?" illustrates the primary concern and the fact that most of our participants do not seem to have an organized approach to ensuring security.

Cross-platform compatibility is still regarded as a major stumbling block for creating rich user-interfaces. Our participants are overwhelmingly conservative in the use of client-side technologies, mainly in fear of creating incompatibilities. For example, one participant remarked: "Most of my designs are simple because of that."

Most participants reported that they frequently test for cross-platform compatibility (see figure 2, question 2.8). However, eight participants remarked that their testing is informal, for example they typically use the 3-4 web browsers they currently have at-hand to check the main functionality of the application.

"Remembering all the little quirks" appears to be a considerable annoyance while integrating different languages (e.g. PHP, JavaScript, HTML, CSS). Furthermore, participants remarked that keeping a growing web application consistent and maintainable is difficult.

Regarding debugging web applications, our participants report that they find it difficult (or impossible) to step through the code line-by-line and to locate the exact source of the problem. Simple print statements appear to be the modus operandi.

Contrary to our expectations only one participant seemed to be dissatisfied with the use of HTML for user interface layout. He mentioned the difficulty of creating complex layouts with HTML tables. Again, the use of advanced client-side features (e.g. CSS2 positioning) appears to be an exception among our participants.

Nr. Question (Scale)	Mean (Std-dev)
2.4. I search the web for snippets of code that I copy, paste & edit. (1=never, 7=very frequently)	3.9 (2.0)
2.5. I consult and scavenge code I have previously written myself. (1=never, 7=very frequently)	5.8 (1.6)
2.7. Do you use a HTML code validator to verify the standard-compliance of your code? (1=never, 7=always)	2.7 (2.0)
2.8. Do you check for cross-browser compatibility? (1=never, 7=always)	4.9 (2.1)
2.9. Do you check for usability? (1=never, 7=always)	4.7 (1.3)
2.10. Do you check for accessibility (for users with disabilities)? (1=never, 7=always)	2.7 (1.9)
2.11. Do you check for scalability & performance issues? (1=never, 7=always)	3.5 (2.1)
2.12. When learning about a new web technology I prefer learning from examples over learning from more general and verbose descriptions. (1=I strongly disagree, 7=I strongly agree):	5.7 (1.3)

Figure 2: Responses to questions asked in the pre-interview questionnaire

The interviewees' answers to question 2.5 (as well as the follow-up discussion) revealed that almost all of our participants quite often reuse code from previous projects. According to the interview responses this reuse is of an informal nature that might be characterized as a simple "copy & paste" strategy.

In addition to inquiring about the frequency of code reuse (see figure 2, Question 2.5) we asked our participants which components they reused most frequently. They responded as follows:

- HTML templates, snippets, header, footer (6)
- Various JavaScript functions (4)
- Database code (4)
- Authentication code (3)
- Validation code (2)
- Code for encoding/decoding data (2)

In order to determine what web developers regard as the key concepts within web application development, we asked our participants what questions they would raise and address in an FAQ (Frequently Asked Questions) for novice programmers. Our participants cited the following concepts:

- Database connectivity and operation (5)
- Difference between client-side and server-side scripts; when to use one or the other (2)
- Page transition, receiving input data (2)
- Practical examples (2)
- Maintaining state (1)
- One-to-many relationships (1)
- Integration of different languages (1)
- User-centered design (1)
- Validation (1)

However, we should note that in general, our participants seemed to have difficulties answering this question even after repeated questioning. We speculate that they had mastered the basics of web development too long ago as to put themselves into the mindset of a novice. Also, they may have had little reason or opportunity to assist novice users.

The results from question 2.9 (figure 2) indicate that our participants frequently assess the usability characteristics of their web applications. As in their software debugging efforts, such evaluation is normally of an informal nature, for example asking colleagues or friends to try out the application and send them feedback. Often, our participants forgo extensive testing in advance and rely instead on gathering user feedback once the application has been provided for actual use. In general, the majority of our participants conveyed that they saw no clear distinction between the activities of prototyping, development, testing, and production. Rather, the common development approach is an informal requirements elicitation phase through one or more meetings with the client and the evolutionary development of the application. Often an early prototype is gradually developed into the final application.

Checks for proper accessibility are even more informal than compatibility and usability testing. In most cases our participants followed what they viewed as "known principles" of accessible web design throughout the development process (e.g., using image-alt tags, considering table linearization by screen readers). Only two participants mentioned using accessibility validation tools like Bobby, text-only browsers or screen-readers to verify the compliance with accessibility standards. From their comments we got the impression that most developers perceive these tools still as being to cumbersome. Six participants said that they never or only rarely checked their HTML code against a code validator.

When asked what they enjoyed about web application development our participants mentioned the following factors:

- Enjoy the challenge; like building things ("It's like playing") (4)
- Quick feedback; ease of checking work (3)
- Quick results (2)
- Diverse work; always something to learn (2)
- Providing useful services to the user (2)
- Quick use of results (1)
- Ease of sharing (1)
- Richness of the medium (1)

Only one participant (CEO of a small-business e-retail company) said that he did not enjoy web development any more and resented the monotony.

In the pre-interview questionnaire and during the interview we asked the participants to identify, rate, and discuss their favorite web development tools. The tools mentioned ranged in complexity from simple text editors, to HTML-code editors like Homesite or BBEdit or Emacs to WYSIWYG editors like Dreamweaver and FrontPage, to development environments like Microsoft Visual Studio. Since five out of the ten participants named Macromedia Dreamweaver MX as their first tool of choice we will discuss it in more detail. Figure 3 shows how the five participants rate Dreamweaver along the dimensions ease-of-learning, ease-of-use, functionality and overall satisfaction.

Macromedia Dreamweaver MX Evaluation (1=low, 7=high)	Mean (Std-dev)
Ease of learning	4.8 (1.8)
Ease of use	5.2 (1.3)
Functionality	5.8 (0.4)
Overall Satisfaction	5.6 (0.5)

Figure 3: Responses from 5 participants regarding their appreciation of Macromedia Dreamweaver MX as a web development tool

Overall, these users of Dreamweaver seem to be satisfied with the tool. They mentioned the site management features (3), the template mechanism (2), its WYSIWYG editing style (2) and general feature-richness (2) as its main strengths. However, they also named some weaknesses. Three participants complained about the stability and reliability of the tool (it crashes or "destroys code"), one of them said that Dreamweaver "feels whimsy" (as opposed to other standard Windows productivity applications).

Another often-heard complaint was that Dreamweaver occasionally generates unnecessary complex code (while including JavaScript "behaviors") and some mentioned a feeling of lack of control over the code. One participant remarked: "The code that gets written is not the code that I'd write myself. My code is cleaner."

Towards the end of the interview we asked what could be done to simplify web application development if there were no limits as to changing standards, resources etc. Many issues were identified but few more than once (with the exception of consistent support for HTML, JavaScript and CSS across all platforms). The list includes: simplified debugging, introduction of high-level components like calendars, better support for reuse, better database connectivity, separation of layers (presentation, application logic), automatic maintenance of state information, and more code-assistance.

The next question inquired about the developers' "wish-list" for their "dream" web development tool. The answers reflected the same issues named in the previous question. Participants also emphasized the desire for better integration of tools, and a responsive, WIMP-style user interface (including copy-and-paste and drag-and-drop functionality) with many predefined components. The exceptions were ideas for natural language style user interfaces, application behavior visualizations, or, at the other extreme, the total abstinence from WYSIWYG in favor of a robust text-only tool.

4. DISCUSSION AND CONCLUSION
The 31 survey responses were provided on a self-selection basis by students, faculty and staff associated with Virginia Tech. Nine out of the ten people from the interview study were associated with Virginia Tech. This may limit the applicability of our results although many of the issues discussed are likely to extend beyond the boundaries of our campus.

Furthermore, the foci for both studies were semi-professionals rather than professional web developers. Although we hesitate to generalize our findings to all web developers (with novices on one end of the spectrum and experts on the other) we do not see many reasons why semi-professional web developers outside of the academic environment should have much different needs and habits than our participants.

Nevertheless, because of the small N of 31 and 10 we classify the survey and the interview survey as pilots, which could be used as templates for more rigorous investigations on a larger scale.

Although the two studies revealed a multitude of issues we see the following ones as most important.

4.1 Ensuring security
Web applications are vulnerable against exploits on many different levels (e.g. operating system, web server software, database, dynamic scripting language, interactions of the aforementioned). Today it is very difficult to build even a "reasonable" secure application or just to assess when an application is secure. Web developers are not confident about the security of their applications and therefore very concerned.

4.2 Cross-browser compatibility
The inconsistencies between different browsers, versions and platforms are not only a major time-sink for web developers but also seem to be the reason why most developers avoid enriching the user experience with advanced features that are only possible with JavaScript, CSS2, or Flash.

4.3 Integrating different technologies
While classical desktop applications are typically based on the syntax of only one programming language (perhaps two when considering database interactions), most web applications combine five or more (HTML, JavaScript, CSS, server-side language, SQL, and perhaps Flash, Curl, Java applets, Active X). The resulting complexity leads to code that is hard to develop and maintain. It also raises the bar for users who want to transition from static page design to more advanced web development.

4.4 Debugging
Most software developers have to deal with bugs. Web developers however face an extra challenge due to the number of technologies involved (see above) and the fact that a web application consists of a part that runs on the server and another on the client.

4.5 Developers' Habits
While the natural tendencies and habits of web developers are not a problem by themselves they can become problems if technology and tools do not account for them.

As many members of our modern society, web developers have little time to waste. Tedious development tasks run the risk of being circumvented or neglected. An example for a quite tedious process is ensuring accessibility. Current accessibility validation tools do not take into account that most developers are unwilling to spend much time designing for accessibility.

Humans deal with concrete examples easier than with abstract concepts. Web developers like to learn by and work with examples but today many tools start up with not more than a blank screen and a myriad of buttons. Where-ever possible, web developers rather modify existing code than rewriting code from scratch. This is particularly true for code that they know well and trust – their own. "Copy & Paste" behavior is often considered an "unclean" engineering practice although it should rather be embraced, exploited and "water-proofed" against its pitfalls.

Semi-professional developers are much more informal than the experts observed by Newman and Landay [3]. Written requirements documents and dedicated prototypes are the exception and a process of evolutionary prototyping of the final web application the rule.

The participants in our interview study like the idea of tools providing abstractions such as ready-made components that speed up development. At the same time they are very critical if the tool limits their control over the development process. Functionality that introduces hard-to-read and complex code (or as one participant calls it: "junk" code) typically fails to win acceptance.

Last but not least, we believe that the productivity and the "fun-factor" in web development would be further increased with "speedy" tools. Web developers appreciate the fact, that they can quickly test ideas, and create programs by what Rosson and Carroll call "debugging-into-existence" [9]. Each extra step or delay that is required for each change has a negative effect.

4.6 Recommendations for the web tool industry

Although the state-of-the-art web standards and tools are generally seen as being appropriate we believe that developers would benefit from:

- Tools that assist developers in producing secure applications

- Tools that are more robust and *faster*, facilitate iterative development and better support debugging

- Tools that provide a large library of ready-to-go components while still giving the developer great control over the created code

- Tools that speed up and automate tedious tasks like HTML validation, cross-platform testing, accessibility checks (which may solve the problem of the general lack of testing)

- Tools that work and act very similar to standard productivity applications like Microsoft Word or PowerPoint, integrate well with those and readily exchange data

- Tools that account for and support the informal tendencies of web developers to learn and work from examples, copy & paste from the web and scavenge prior projects

Addressing the complexity caused by the plethora of web technologies and working towards better standard-compliance and cross-browser compatibility are challenges for the web engineering community as a whole.

5. REFERENCES

[1] Berners-Lee, T. (1996). WWW: past, present, and future. *IEEE Computer*, v.29 n.10, 69-77.

[2] Fraternali, P. (1999). "Tools and Approaches for Developing Data-Intensive Web Applications: A Survey." ACM Computing Surveys 31(3): 227-263.

[3] Newman, M., Landay, J. (2000). "Sitemaps, storyboards, and specifications: a sketch of Web site design practice." Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques, 263-274, August 17-19, 2000, New York City, New York, United States

[4] Rode, J. (2002a). Survey: "Interactive Websites", Input Form. July 12, 2002. http://filebox.vt.edu/users/jrode/publish/2002-05-survey/form/

[5] Rode, J. (2002b). Survey: "Interactive Websites", Results. July 12, 2002. http://filebox.vt.edu/users/jrode/publish/2002-05-survey/results/

[6] Rode, J. (2003). Pre-Interview questionnaire: "Web Application Development", Input Form. May 2003 http://filebox.vt.edu/users/jrode/publish/2003-05-interviews/

[7] Rode, J., Rosson, M.B. (2003). "Programming at runtime: Requirements and paradigms for nonprogrammer web application development." In *Proceedings of HCC'03*, Auckland, New Zealand, IEEE.

[8] Rode, J. M.B. Rosson, M. A. Pérez-Quiñones (2004). End users' Mental Models of Concepts Critical to Web Application Development. IEEE HCC 2004. Rome, Italy. Oct. 26-29.

[9] Rosson, M. B. and J. M. Carroll (1993). Active programming strategies for reuse. Proceedings of ECOOP'93: Object-Oriented Programming, 7th European Conference, Kaiserslautern, Germany, 26-30 July, Springer-Verlag.

[10] Vora, P. R. (1998). "Designing for the Web: A Survey." ACM interactions (May/June): 13-30.