Web Usage Patterns of Developers

Christopher S. Corley
ABB Corporate Research
Raleigh, NC, USA
christopher.corley@us.abb.com

Federico Lois, Sebastián Quezada
Corvalius
Buenos Aires, Argentina
{federico.lois, sebastian.quezada}@corvalius.com

***Abstract*—Developers often rely on the web-based tools for troubleshooting, collaboration, issue tracking, code reviewing, documentation viewing, and a myriad of other uses. Developers also use the web for non-development purposes, such as reading news or social media. In this paper we explore whether web usage is detriment to a developer's focus on work from a sample over 150 developers. Additionally, we investigate if highly-focused developers use the web differently than other developers. Our qualitative findings suggest highly-focused developers use the web differently, but we are unable to predict a developer's focused based on web usage alone. Further quantitative findings suggest that web usage does not have a negative impact on a developer's focus.**

***Index Terms*—web activity; interruptions; developer focus; personal software process**

I. Introduction

Today's developer often relies on a myriad of tools for software development, and increasingly, these tools are located on the web. Whereas in yesteryears, a developer required hard copies of documentation in the form of books and manuals, today's developer relies on online documentation and search engines to find relevant information [1]. Tools for software configuration management are becoming ubiquitously hosted online as web applications, whether via a host such as GitHub or Visual Studio Online. Even expertise finding has been moving from asking around the office and mailing lists to online Q&A sites such as StackOverflow [2]. Indeed, researchers are aware of the usefulness in investigating how developers are using these new tools [3, 4, 5, 6].

With a new set of tools comes a new set of problems. Namely, the Internet hosts much more than just useful tools. These websites might result in the productivity loss of developers by being distractions. Developer interruptions are a common occurrence and can cost a developer between 10 and 15 minutes to resume their task [7]. It would be useful to know which websites are likely to lead to interruptions in order to proactively prevent interruptions from occuring.

Storey et al. [3] put forth a set of questions about the social media category of web sites. We look to investigate their final question, "Does social media lead to interruptions or information overload that could impair a developer's performance?" However, we expand this question into a more general question: *does web usage lead to interruptions that could impair a developer's performance?* Towards the goal of answering this question, we mine data from the users of the online web tool *Codealike*[1]. Codealike is a quantified-self platform for developers that monitors activity, such as coding, reading, or debugging, within the integrated development environments (IDEs) Visual Studio and Eclipse. From a developer's IDE activity, Codealike can infer the level of concentration, or *focus*, a developer has while working. Additionally, Codealike also provides a Chrome extension that tracks a developer's web navigation and debugging. The extension allows the developer to view time spent on various web activies, such as troubleshooting on StackOverflow.

In this paper, we study the web usage data of developers during periods of IDE activity. In particular, we present a qualitative and quantitative study of 168 Codealike users. We find that developers that maintain long periods of focus have distinct patterns in the way they use the web. Further, we also find that, even for a typically low-focused developer, many categories of web sites are not detrimental to their focus.

We first describe Codealike at a high level and how it uses IDE activity to measure developer focus. We next describe our study and results. We also discuss and draw recommendations developement teams and project managers should consider. Finally, we conclude and give direction for future work.

II. Codealike

Codealike monitors IDE activity to track the activities of developers while coding in an unnoticeable, undisruptive, and non-intrusive way; building metrics upon the recorded data which can bring insightful and actionable information on the individual and collective software development process.

Among other measurements Codealike has a built in metric that aims to estimate individual developer's focus, or flow [8]. The focus level is modeled by considering a user's active and inactive states over time with a growth and a decay function.

Codealike models 4 primary states: coding, debugging, building, and idle. The state machine is outlined in Figure 1. The coding state is when the IDE has focus and is not in debug or build modes. Debugging occurs when the IDE has focus and is in debug mode. Building occurs when the IDE has focus and is in build mode. The idle state occurs after the developer has

[1] http://www.codealike.com/

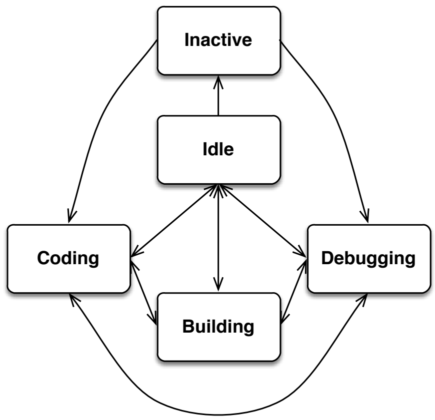

Fig. 1. State machine of the primary active states

been in any of the previous three states and the IDE does not show activity.

For the main states we consider a tolerance threshold to smooth activities transitions. The user enters the idle state whenever the IDE has lost focus longer than 1 minute. When the IDE loses focus for a short interval and then regains focus into the previous state, the transition is ignored and considered a continuous interval. During these short interruptions we assume that the short term memory is not affected by context switching. When the IDE loses focus for a longer than 7 minutes, then the transition is made from idle to inactive.

From these four states, two other states can be derived: reading and editing. Reading is when the user is in a coding state, but not changing any file. Likewise, editing is when a user is changing a file.

The growth function is modeled with the assumption that the working memory assigned to a task will grow very slowly at first, while the recovering related facts [9]. We consider a developer to have transitioned to a high memory state after approximately 10 minutes. After first transitioning to a high memory state, the growth function increases at a linear rate. After 30 minutes of a high memory state the growth potential accelerates until it reaches an asymptotic maximum and the user is very likely to have hit flow state based on their activity pattern.

Similarly the decay function is modeled after the expected behavior of the working memory impact of an interruption. There is little chance the user can sustain in memory to the task at hand [7]. Hence, the focus begins to decay exponentially.

Codealike also considers web activity as part of a developer's focus. A developer may need to look for particular information related to the task at hand. Here, we can to do a better classification of the idle state when web usage data is available.

III. Study

In this section we describe the design of a study in which we investigate the web usage of developers. We describe the case study using the Goal-Question-Metric approach [10]. We then describe and discuss the results of the study.

A. Definition and Context

Our *goal* is to better understand the web usage patterns of developers and the affects web usage has on productivity. The *quality focus* of the study is on informing development decisions and policy changes that could lead to software with fewer defects. The *perspective* of the study is of a researcher, developer, or project manager who wishes to gain understanding of the best way to support developer's web usage needs. The *context* of the study spans 168 developers and users of Codealike.

Toward the achievement of our goal, we pose the following research questions:

- *RQ1* Do highly-focused developers use the web differently than lesser-focused developers?
- *RQ2* Does web usage during coding sessions affect focus?

At a high level, we want to know if web usage is a factor that determines whether a developer can maintain long periods of high focus.

In the remainder of this section we introduce the subjects of our study, describe our methodology, our data collection and analysis procedures, and report the results of the study.

B. Subjects

Our sample includes IDE activity and web usage data over a 4 month period between December 17th, 2014 and April 17th, 2015. We sampled subjects from all Codealike users that used Codealike within an IDE and Codealike's Web Tracking extension for Google Chrome during the 4 month period. The sample includes 168 developers total.

Codealike allows users to self-identify various professional developing experience as part of their online profile. The experience levels available for selection include no experience (amateur), less than 5 years (junior), greater than 5 years (senior), or as a Microsoft MVP. Of the 168 subjects, 41 identify as amateur (24%), 36 as junior (21%), 78 as senior (46%), 5 as Microsoft MVP (3%), and 8 had not made a selection (6%).

Further, of these 168 subjects, 156 develop in Visual Studio (only non-Free releases) and 12 develop in Eclipse. Additionally, all subjects developed in only one of the two IDEs during the sampled period.

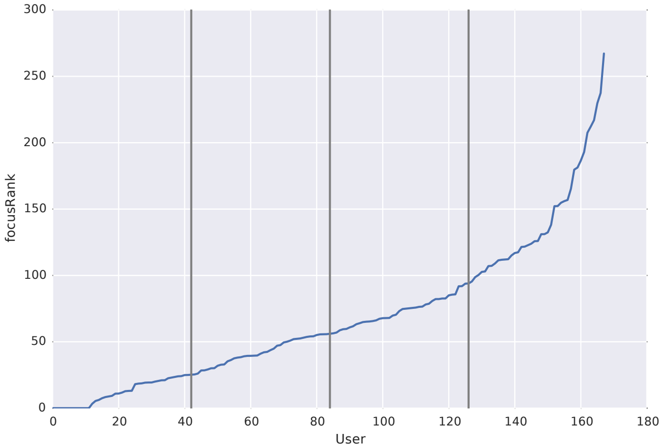

Fig. 2. Focus rank, or mean daily minutes over threshold, of each subject

TABLE I
CATEGORIES AND TOTAL TIME TRACKED FOR EACH

Category	Total time tracked		
Blogs, News & Reading	49d	23h	57m
Courses & Education	15d	6h	38m
Developer Tools	10d	23h	4m
Gaming	19d	5h	55m
Informational	8d	8h	35m
Internal	53d	4h	9m
Localhost	88d	4h	36m
Management Tools	7d	0h	46m
Music & Videos	78d	1h	29m
Office Collaboration	26d	14h	50m
Office Tools	19d	4h	51m
Quantified Self	14d	0h	13m
SCM & ALM	86d	8h	54m
Search Engines	49d	11h	54m
Shopping	17d	2h	50m
Social Networks	135d	0h	33m
Troubleshooting	104d	22h	3m
Webmail Clients	68d	19h	24m

C. Methodology

In order to answer our research question of whether a user with high focus uses the web differently than a lower-focused user, we partitioned the users into quartiles. The quartiles are based on a ranking by the number of minutes a developer maintains high focus. We calculate each developer's ranking using the following formula:

$$focusRank(d) = \frac{\sum_t^T focus(t) > threshold?}{daysActive(d)} \qquad (1)$$

where the T is minutes of activity for developer d, $focus(t)$ returns the focus level (0.0 to 1.0) for an individual minute t, $daysActive(d)$ returns the total number of days a developer has activity, and threshold is set to the average focus level of all users of Codealike (0.52). Equation 1 gives us the daily mean a developer is able to achieve focus above the community average. Figure 2 shows the $focusRank$ for each subject and the quartiles.

We also needed to categorize each website visited by a subject. We first considered using the Open Directory Project (ODP), also known as DMoz[2]. However, even given the breadth of categories and sites known by ODP, many websites were not within the directory. Hence, we chose to manually categorize each website.

During categorization at least two authors tagged each site by hand. We tagged in iterations starting with the sites most visited and any conflicts of categories were resolved with discussion. We only tagged sites that had more than 3 hours spent total for all subjects. We generated 27 different categories during this process. For our analysis, we exclude any category with less than one week of total time spent, leaving 18 categories. Table I shows the categories and total time spent visiting sites in that category by all subjects.

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

The "Troubleshooting" category includes Q&A websites such as StackOverflow. "Social Networks" includes websites such as Facebook, Twitter, and Google Plus. "SCM & ALM" includes websites for source code and application lifecycle management, such as GitHub and JIRA instances. "Blogs, News & Reading" includes websites such as Google News, BBC, Feedly (an RSS aggregator) and Medium (a blog hosting platform). Music & Videos includes music or video streaming services, such as Spotify, YouTube and Netflix. The "Internal" category includes any sites with a localized IP address, likewise "Localhost" includes sites such as `127.0.0.1`, `0.0.0.0`, or `localhost`.

D. Data Collection and Analysis

To answer RQ1, we qualitatively investigate the web usage patterns for an average developer in each quartile. We create heatmaps of each hour in the week based on the total amount of usage for that hour per category. An example heatmap can be seen in Figure 5, which shows the times a user spends on troubleshooting-related websites. We use the heatmaps to discover differences between the quartiles. Further, we wanted to know if a particular category has any correlation with the amount of focus for any given hour of the week. We calculate Pearson's r for each category to determine it's correlation to focus.

To answer RQ2, we use machine learning to determine whether we can predict which quartile a developer would be in given the web usage activity for that hour. In particular, we use a support vector machine (SVM) [11] to classify web usage into quartiles. Since SVMs are effective in high dimensional spaces, we create vectors of the total time spent in each category for every hour any developer is active. We randomly split the activity vectors into a training set with 90% of the total size and a test set of the remaining vectors.

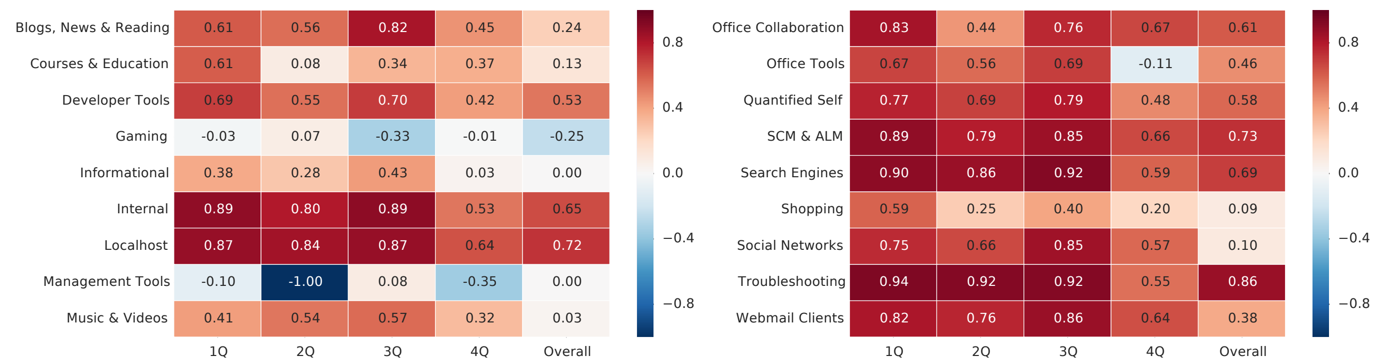

Fig. 3. Correlation heatmap of categories for each quartile and overall

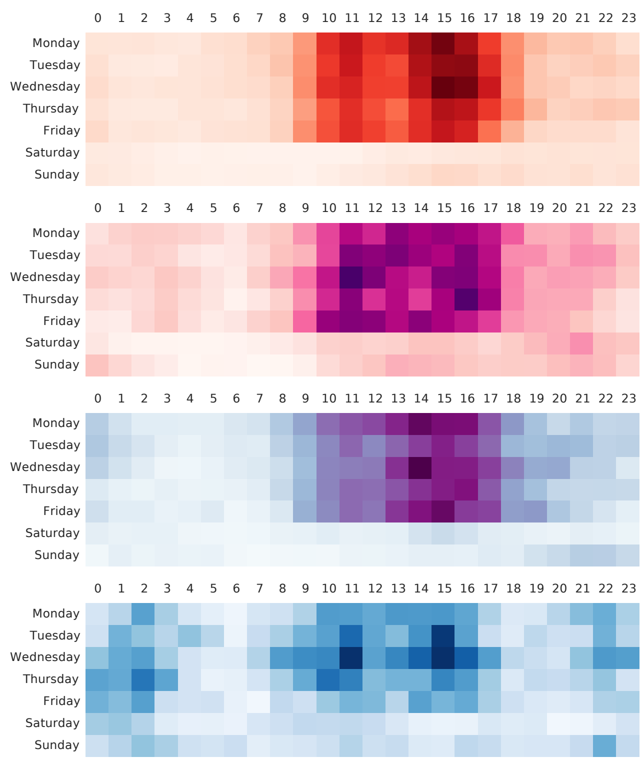

Fig. 4. Hours of high focus for 1st (top, red), 2nd, 3rd, and 4th (bottom, blue) quartiles

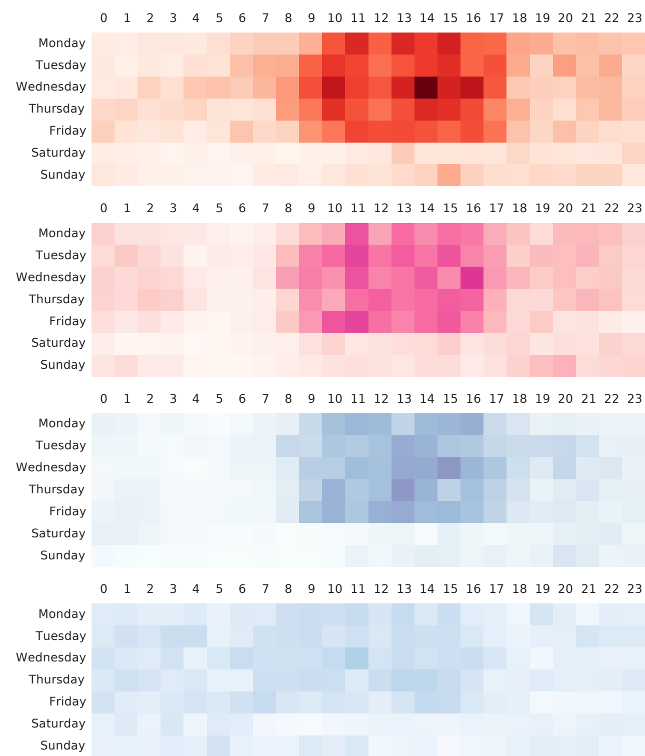

Fig. 5. Hours spent visiting Troubleshooting websites for 1st (top, red), 2nd, 3rd, and 4th (bottom, blue) quartiles

E. Results & Discussion

Figure 3 shows the Pearson's correlation between time spent visiting a website for each category and the developer's focus during that hour. There are several categories that have high positive correlation, such as "Troubleshooting", "Search Engines", "SCM & ALM", "Office Collaboration", "Office Tools", "Quantified Self", and "Developer Tools". Categories with low or negative correlation with focus include Gaming-related sites, "Music & Videos", "Blogs, News & Reading", and "Social Networks", as well as others.

1) Patterns and Focus: For example, the "Troubleshooting" category has a 0.86 correlation with focus overall, with the 1st, 2nd, and 3rd quartiles at 0.94, 0.92, and 0.92 correlation, respectively. Figure 4 shows the hours of highest focus for each quartile, while Figure 5 shows hours of highest usage of the "Troubleshooting" category. From these figures, we can see that the 1st quartile, or developers with very high focus, routinely use the web to troubleshoot. The correlation value confirms that focused developers are able to maintain focus on the task at hand while simultaneously troubleshooting on the web.

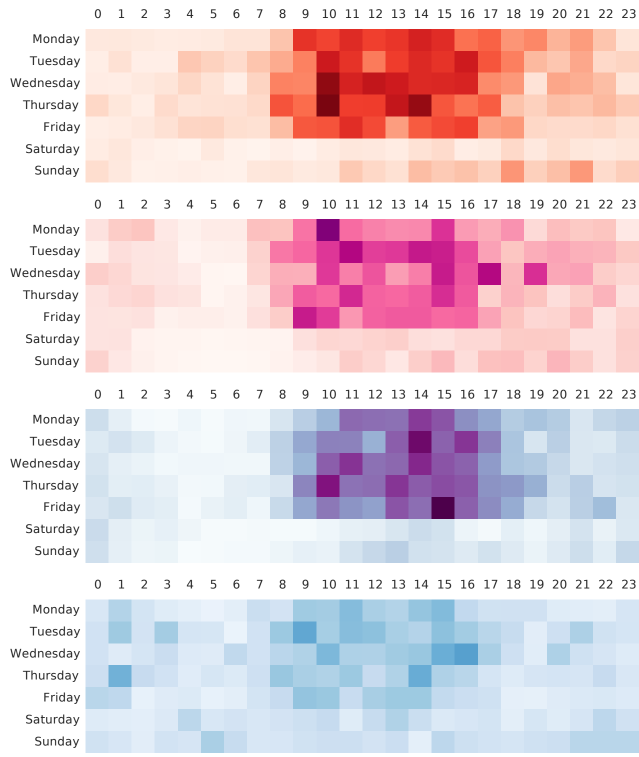

Fig. 6. Hours spent visiting Search Engines for 1st (top, red), 2nd, 3rd, and 4th (bottom, blue) quartiles

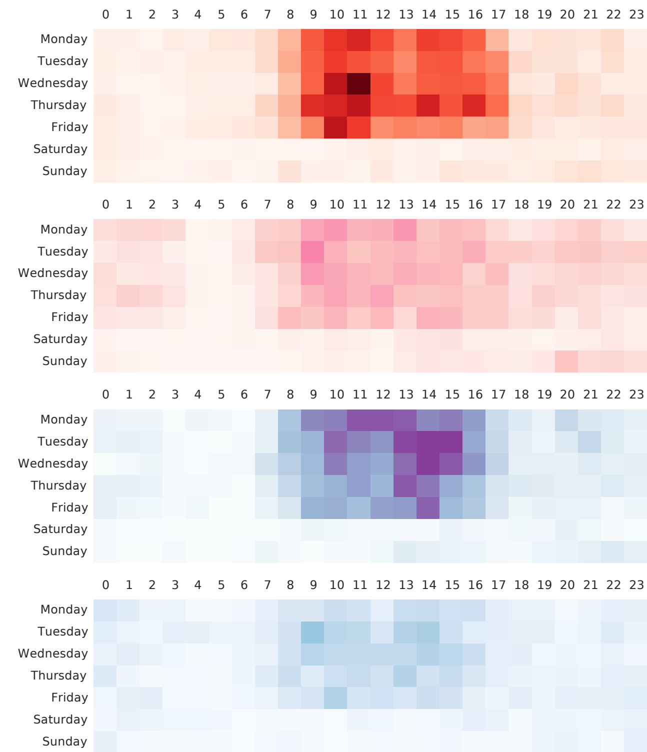

Fig. 7. Hours spent visiting SCM & ALM websites for 1st (top, red), 2nd, 3rd, and 4th (bottom, blue) quartiles

Likewise, developer usage of search engines also has high correlation to focus. We hypothesize that highly-focused developers waste no time to search and troubleshoot an issue they are having. We also note that highly-focused developers use both search engines and troubleshooting websites more than other quartiles, as shown by Figures 6 and 5.

There is also a high positive correlation, 0.73 overall, of time spent visiting "SCM & ALM" sites. Figure 7 shows a concentration of usage during work hours from the highly-focused quartile. The 2nd and 4th quartiles do not seem to visit "SCM & ALM" sites much at all, while the 3rd quartile does.

Figure 8 shows that highly-focused developers often spend time using websites in the "Office Collaboration" category, such as Slack (a chat platform), while other quartiles do not. Interestingly, the lowest quartile also spends some time using Office Collaboration sites. The correlation between focus and these sites is also high, at 0.61. The correlation is especially high for the 1st and 3rd quartiles, at 0.83 and 0.76, respectively.

Social Networks do not have a strong correlation at 0.10. Figure 9 shows that the lowest focus quartile, in blue, browse social networking sites much higher than the other quartiles. This implies that social networking is detrimental to a developer's focus, and the highly-focused developers are consciously not visiting these sites. However, when viewed by quartile, Social Networks actually have a high positive correlation for the 1st

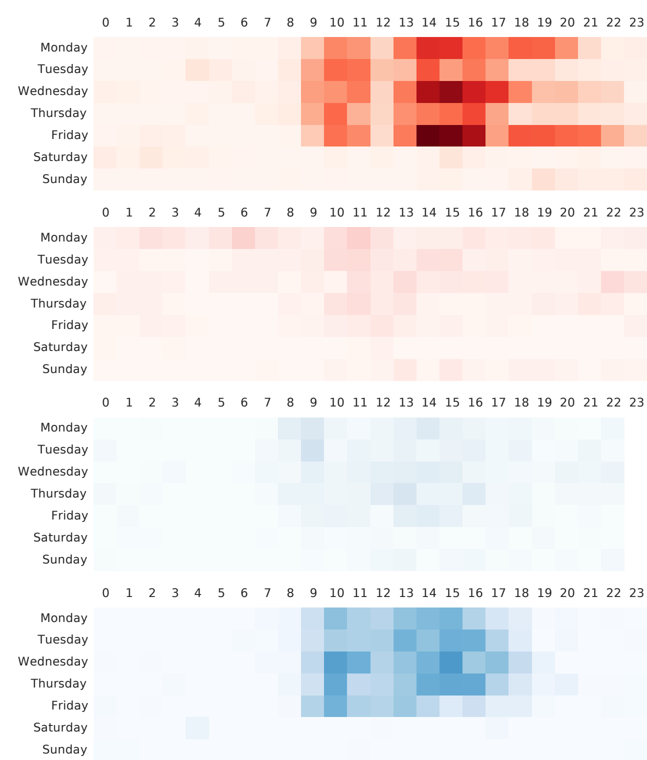

Fig. 8. Hours spent visiting Office Collaboration websites for 1st (top, red), 2nd, 3rd, and 4th (bottom, blue) quartiles

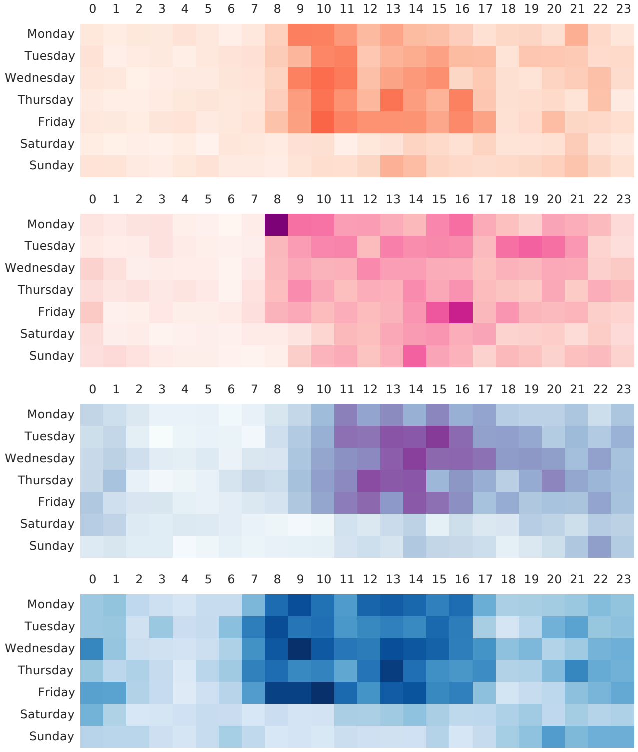

Fig. 9. Hours spent visiting Social Networks websites for 1st (top, red), 2nd, 3rd, and 4th (bottom, blue) quartiles

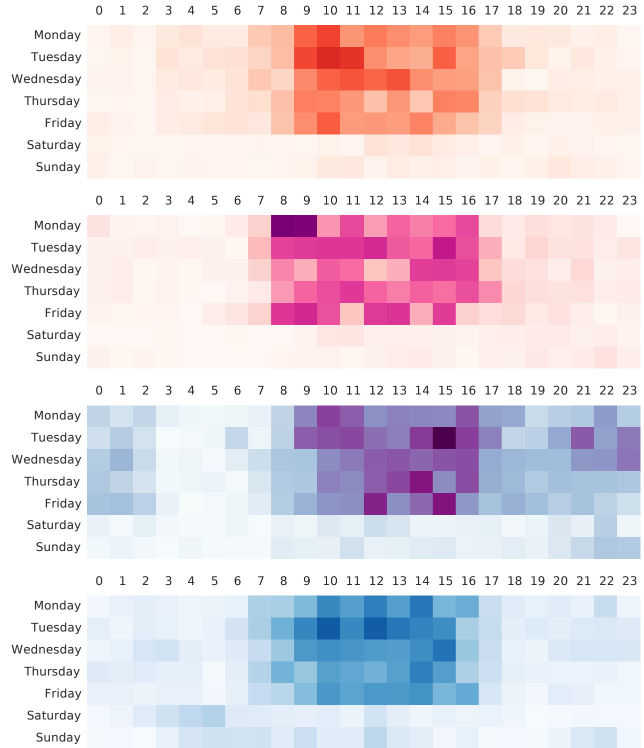

Fig. 10. Hours spent visiting Webmail Clients websites for 1st (top, red), 2nd, 3rd, and 4th (bottom, blue) quartiles

and 3rd quartiles. We hypothesize that these groups aren't necessarily using the sites to socialize, but to collaborate with teammates via messaging.

The "Webmail Clients" category also maintains a slight positive correlation at 0.38 overall. Figure 10 shows mail usage is mostly consistent across all quartiles, with some non-work hour usage by the 3rd quartile. Again, the 1st and 3rd quartiles are able to maintain focus while communicating via webmail.

We also see a medium positive correlation (0.4631) with the "Office Tools" category. Figure 11 shows that the 1st and 3rd quartiles in particular have a high usage of office tools, such as Microsoft Office 365 and Google Drive.

Figures 13 and 12 show no clear distinctions between the first three quartiles for "Music & Videos" and "Blogs, News & Reading". However, it does appear the lowest focus quartile does visit sites in the "Music & Videos" category more often in non-work hours than other groups.

Categories with negative correlation include Gaming-related sites and Managment Tools, such as time reporting and billing tools.

2) Predicting Focus: We also investigate whether SVM can predict a developer's focus quartile based on web browsing activity alone. The SVM was able to correctly predict a developer's quartile a meager 26% of the time when all hours with web usage are considered. If we remove the hours where

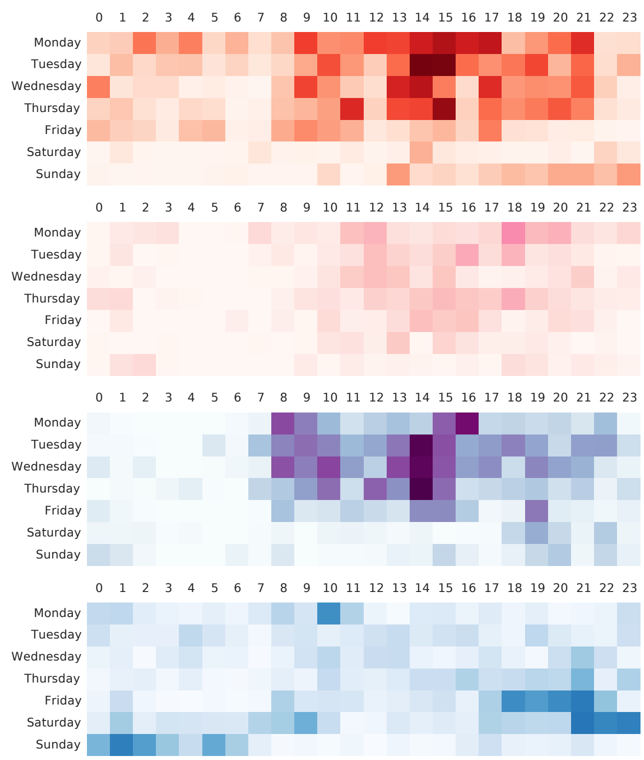

Fig. 11. Hours spent visiting Office Tools websites for 1st (top, red), 2nd, 3rd, and 4th (bottom, blue) quartiles

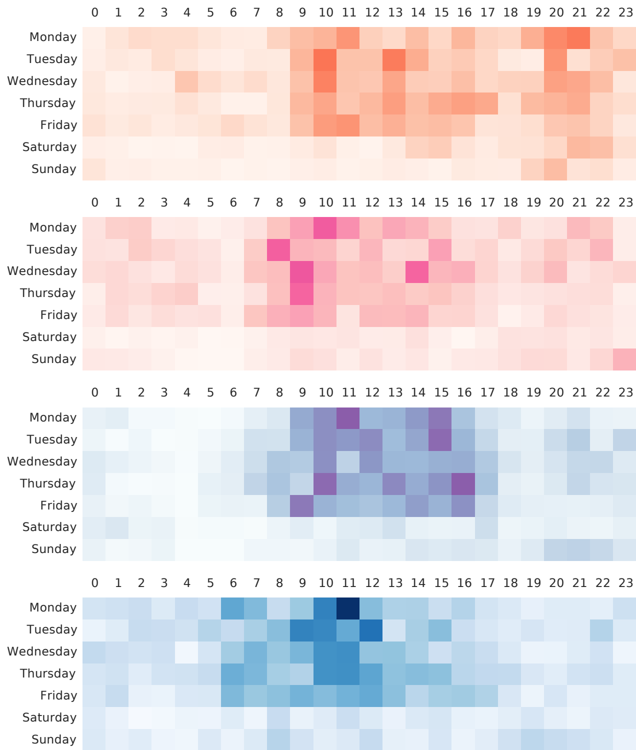

Fig. 12. Hours spent visiting Blogs, News & Reading web sites for 1st (top, red), 2nd, 3rd, and 4th (bottom, blue) quartiles

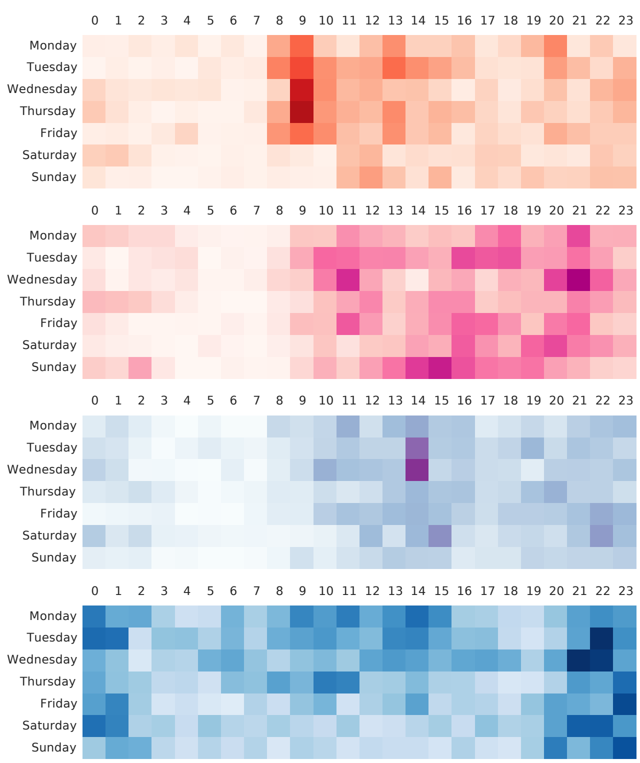

Fig. 13. Hours spent visiting Music & Videos websites for 1st (top, red), 2nd, 3rd, and 4th (bottom, blue) quartiles

web usage exists, but there is no IDE activity, we can boost prediction to 36%. However, as there are only 4 quartiles SVM can classify vectors into and thus random guessing yields 25% accuracy, we conclude that focus quartile can not be predicted based upon web usage alone.

F. Key Findings & Recommendations

Our key findings derived from Section III-E are that highly-focused developers use the web differently, but web usage alone is not a predictor of focus quartile. Further, web usage can affect focus, but has different affects across focus quartiles.

Our recommendations are as follows. First, as a developer, do not hesitate to search for your problem on the web using a search engine or a Q&A site such as StackOverflow. Finding the related API documentation or answer to a problem quickly allows you to concentrate on the task on a higher level [1].

Second, we find that highly focused developers tend to avoid social networking sites. However, they also correlate highly with focus for most groups. Hence, we recommend that social media should be something visited when you have a particular information need, such as asking a question about the task at hand. We also notice that highly-focused developers rely on office collaboration tools, such as company-wide chats. These collaboration tools, along with social media, should be used wisely for information needs.

Finally, developers should consider avoiding spending time pursing entertainment-related sites, such as Music & Videos or Blog, News & Reading sites. While listening to music while working may help some concentrate [12], we advise to not spend time needlessly switching or browsing for music.

IV. Related Work

Brandt et al. [1] present the first work in exploring how developers are able simultaneously forage the web for information, learn that information, and write code. They find that developer's rely on online resources for various reasons, including learning new information, clarifying already known information, and to remind themselves of details otherwise forgotten.

Black et al. [13] conduct a survey of developers to collect information on their social media usage. Storey et al. [3] present paper positioning social media use by developers as a increasingly important aspect of software development. Parnin and Treude [14] investigate how developers document and communicate usage of the jQuery API. They find that the majority (88%) of the API is covered by blog posts.

Stylos et al. [15] present Mica, a web searching tool finding API documentation. Hoffmann et al. [16] amalgamate Java JAR files with API documentation and tutorials to create a novel approach for searching. Gottipati et al. [17] investigate how developers search online forums to find answers to technical problems and propose a semantic search engine to assist developers in quickly finding solutions.

Vasilescu et al. [18] find the association between StackOverflow and GitHub productivity for over 45 thousand developers. They find that StackOverflow usage increases the rate developers commit code to GitHub. Vasilescu et al. [2] show that there is an increasing reliance on StackOverflow and a decline in usage of mailing lists. Dabbish et al. [19] present results of interviews with users of GitHub on how they perceive other developers based on site activity.

V. Conclusion and Future Work

In this study we collected and analysed the web usage and focus levels of 168 developers. Our results suggest that web usage is a significant factor in whether a developer is considered to have high focus, but we are unable to predict whether a developer will be focused based on web usage alone. We also find that web usage in general can help developers maintain focus.

Future work includes expanding this study outside of public Codealike users. In particular, we would like to extend the study to include ABB developers. While this study focuses on public Codealike users and we have a sense of the nature of work that each user is doing, whether they are developing as a hobby, open source, or as an employee in industry, we must rely on users self-identifying correctly.

Additional future work involves an in-depth study of the affects of content-control filters. We would like to compare the web usage and focus of developers that have complete access to the web to developers that are working behind a web content-control filter, such as Microsoft Forefront[3] or Websense[4].

References

[1] J. Brandt, P. J. Guo, J. Lewenstein, M. Dontcheva, and S. R. Klemmer, "Two studies of opportunistic programming: interleaving web foraging, learning, and writing code," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2009, pp. 1589–1598.

[2] B. Vasilescu, A. Serebrenik, P. Devanbu, and V. Filkov, "How social q&a sites are changing knowledge sharing in open source software communities," in *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, ser. CSCW '14. New York, NY, USA: ACM, 2014, pp. 342–354.

[3] M.-A. Storey, C. Treude, A. van Deursen, and L.-T. Cheng, "The impact of social media on software engineering practices and tools," in *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research*, ser. FoSER '10. New York, NY, USA: ACM, 2010, pp. 359–364.

[4] A. Bacchelli, "Mining challenge 2013: Stack overflow," in *The 10th Working Conference on Mining Software Repositories*, 2013, p. to appear.

[5] G. Gousios, "The ghtorrent dataset and tool suite," in *Proceedings of the 10th Working Conference on Mining Software Repositories*, ser. MSR'13, 2013, pp. 233–236.

[6] A. T. Ying, "Mining challenge 2015: Comparing and combining different information sources on the stack overflow data set," in *The 12th Working Conference on Mining Software Repositories*, 2015.

[7] C. Parnin and S. Rugaber, "Resumption strategies for interrupted programming tasks," *Software Quality Journal*, vol. 19, no. 1, pp. 5–34, 2011.

[8] M. Csikszentmihalyi, *Flow: The psychology of optimal experience*. HarperPerennial New York, 1991, vol. 41.

[9] S. T. Iqbal and B. P. Bailey, "Understanding and developing models for detecting and differentiating breakpoints during interactive tasks," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '07. New York, NY, USA: ACM, 2007, pp. 697–706.

[10] V. Basili, G. Caldiera, and H. Rombach, "The goal question metric approach," *Encyclopedia of software engineering*, vol. 2, no. 1994, pp. 528–532, 1994.

[11] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[12] Y.-N. Shih, R.-H. Huang, and H.-s. Chiang, "Correlation between work concentration level and background music: A pilot study," *Work*, vol. 33, no. 3, p. 329, 2009.

[13] S. Black, R. Harrison, and M. Baldwin, "A survey of social media use in software systems development," in *Proceedings of the 1st Workshop on Web 2.0 for Software Engineering*. ACM, 2010, pp. 1–5.

[14] C. Parnin and C. Treude, "Measuring api documentation on the web," in *Proceedings of the 2Nd International Workshop on Web 2.0 for Software Engineering*, ser. Web2SE '11. New York, NY, USA: ACM, 2011, pp. 25–30.

[15] J. Stylos, B. Myers *et al.*, "Mica: A web-search tool for finding api components and examples," in *Visual Languages and Human-Centric Computing, 2006. VL/HCC 2006. IEEE Symposium on*. IEEE, 2006, pp. 195–202.

[16] R. Hoffmann, J. Fogarty, and D. S. Weld, "Assieme: finding and leveraging implicit references in a web search interface for programmers," in *Proceedings of the 20th annual ACM symposium on User interface software and technology*. ACM, 2007, pp. 13–22.

[17] S. Gottipati, D. Lo, and J. Jiang, "Finding relevant answers in software forums," in *Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 323–332.

[18] B. Vasilescu, V. Filkov, and A. Serebrenik, "Stackoverflow and github: Associations between software development and crowdsourced knowledge," in *Social Computing (SocialCom), 2013 International Conference on*, Sept 2013, pp. 188–195.

[19] L. Dabbish, C. Stuart, J. Tsay, and J. Herbsleb, "Social coding in github: Transparency and collaboration in an open software repository," in *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, ser. CSCW '12. New York, NY, USA: ACM, 2012, pp. 1277–1286.

Appendix

In this appendix, we include hours spent heatmaps for categories not discussed in the paper.

[3] http://microsoft.com/forefront/

[4] http://www.websense.com/

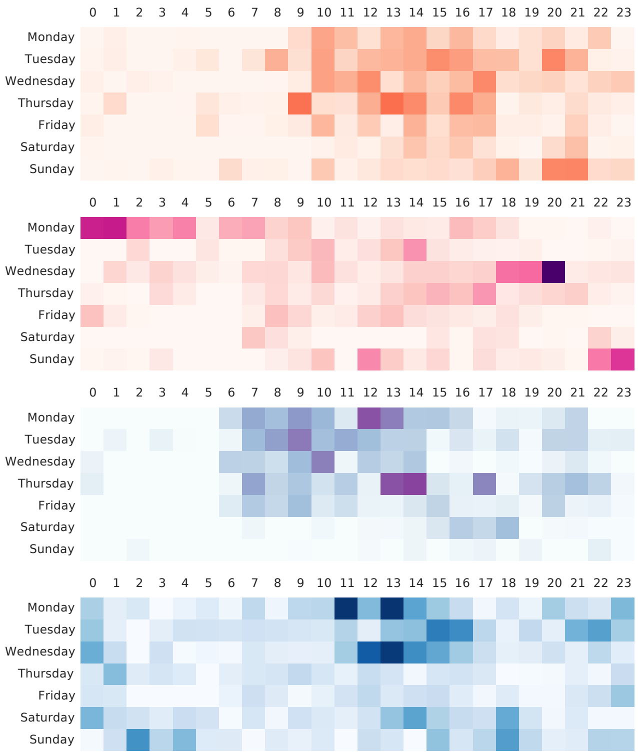

Fig. 14. Hours spent visiting Courses & Education websites for 1st (top, red), 2nd, 3rd, and 4th (bottom, blue) quartiles

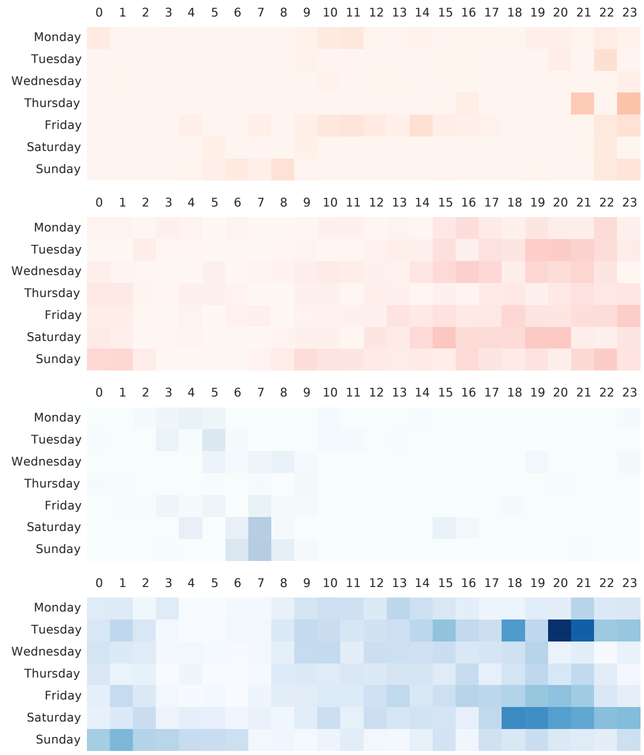

Fig. 16. Hours spent visiting Gaming websites for 1st (top, red), 2nd, 3rd, and 4th (bottom, blue) quartiles

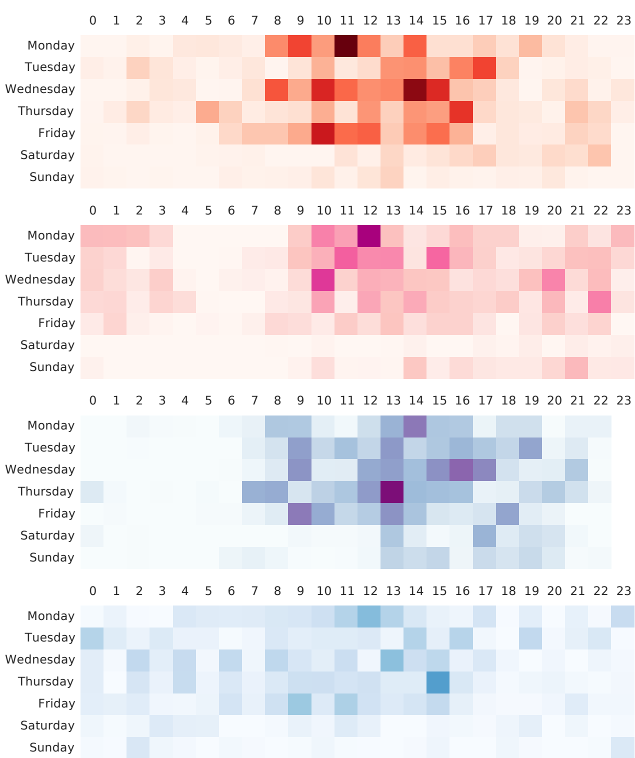

Fig. 15. Hours spent visiting Developer Tools websites for 1st (top, red), 2nd, 3rd, and 4th (bottom, blue) quartiles

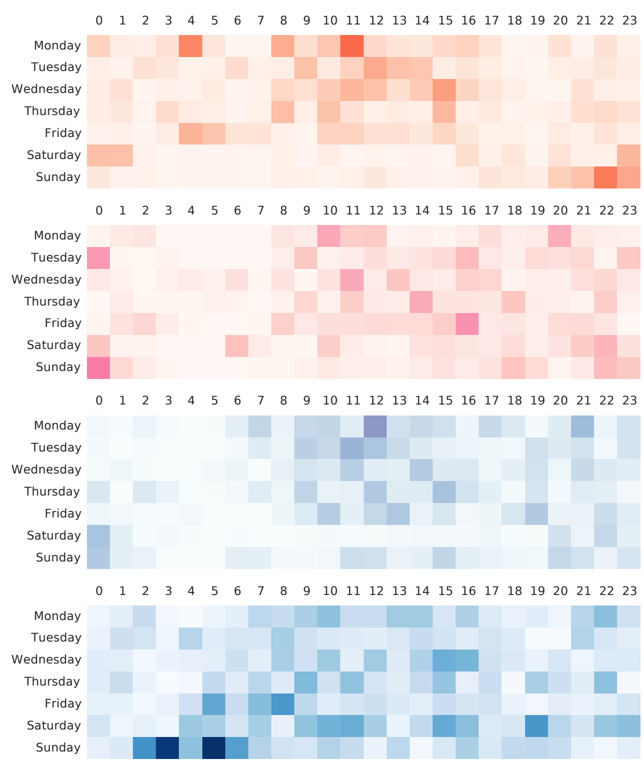

Fig. 17. Hours spent visiting Informational websites for 1st (top, red), 2nd, 3rd, and 4th (bottom, blue) quartiles

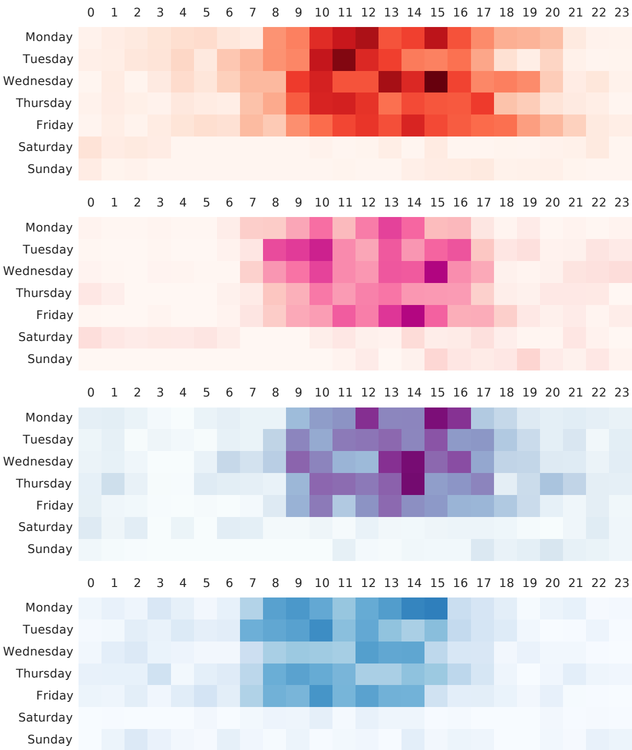

Fig. 18. Hours spent visiting Internal websites for 1st (top, red), 2nd, 3rd, and 4th (bottom, blue) quartiles

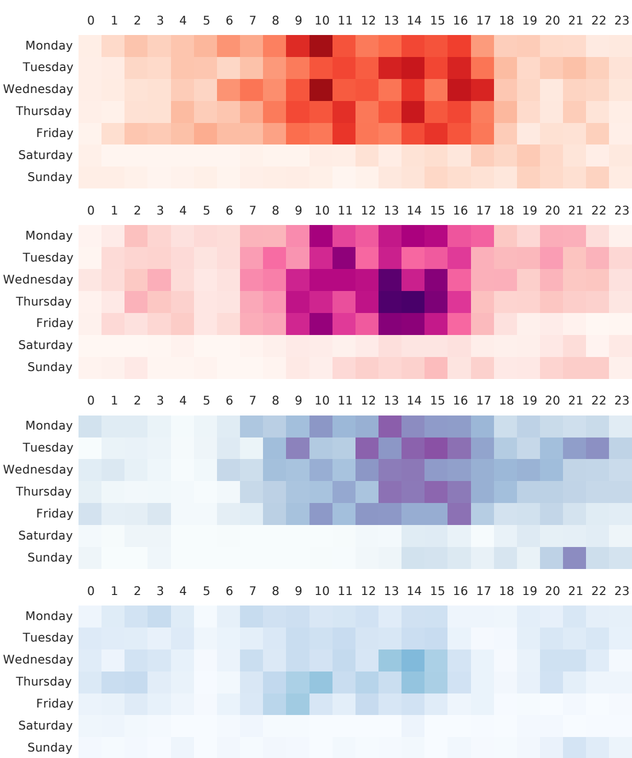

Fig. 19. Hours spent visiting Localhost websites for 1st (top, red), 2nd, 3rd, and 4th (bottom, blue) quartiles

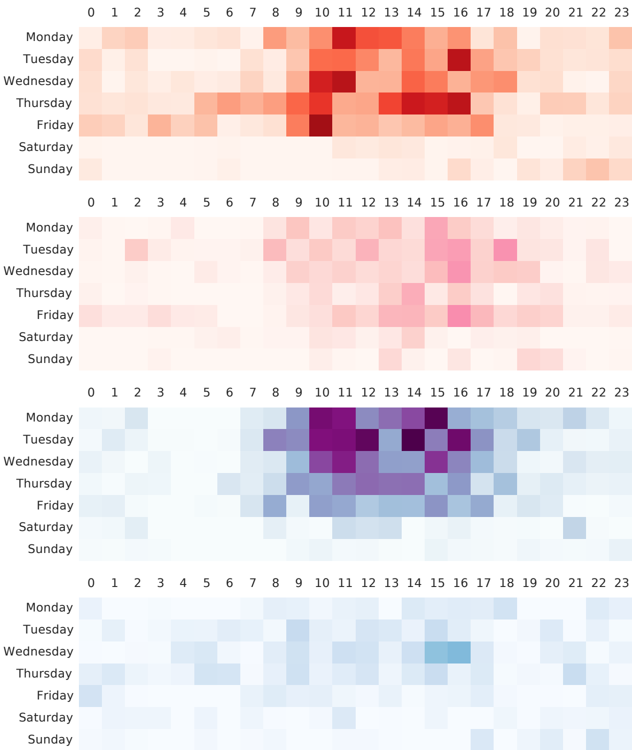

Fig. 20. Hours spent visiting Quantified Self websites for 1st (top, red), 2nd, 3rd, and 4th (bottom, blue) quartiles

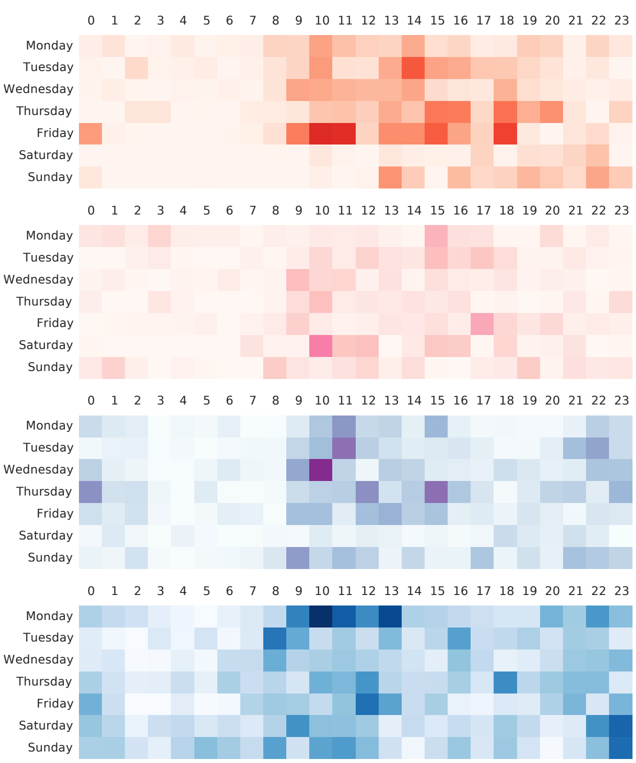

Fig. 21. Hours spent visiting Shopping websites for 1st (top, red), 2nd, 3rd, and 4th (bottom, blue) quartiles