Received: 16 February 2017 | Revised: 14 September 2017 | Accepted: 7 February 2018

DOI: 10.1111/gwao.12255

ORIGINAL ARTICLE

WILEY

# Gendered work culture in free/libre open source software development

Yu-Wei Lin[1] | Matthijs den Besten[2]

[1] Division of Communications, Media and Culture, University of Stirling, UK

[2] Montpellier Business School, Montpellier, France

**Correspondence**
Yu-Wei Lin, Division of Communications, Media and Culture, Pathfoot Building, University of Stirling, UK.
Email: yuwei.lin@stir.ac.uk

This article adopts a feminist perspective to examine masculine work culture in the development of free/libre open source software. The authors draw on a case study of the 'Heidi bug' discovered during the development of the Mozilla Firefox web browser to examine how 'gendered talk' was (en)-acted to facilitate 'bricolage' in an online work environment. Such gendered talks contain cultural references familiar to male developers. Though seemingly innocuous, such acts could be seen as a performance of gender that simply reflects the hegemonic heterosexual masculine culture manifested in an online virtual work space. The virtual work space therefore can be exclusive to those who shared the cultural references. Although it may not necessarily be ignorance or insensitivity of male developers, a more gender-balanced, women-friendly and inclusive workplace certainly would benefit from a more diverse environment. This article highlights the gendered aspect of software development through examining the language use and mainstream 'bricolage' practice, and establishes a compelling ground for enlarging the talent pool to include more women and integrating gender ethics (e.g., raising awareness of sensitive language and design approaches) into computer ethics education.

**KEYWORDS**

free/libre open source software, gendered talks, heterohegemonic masculinity, humour, online virtual work space, open innovation, work cultures

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.
© 2018 The Authors Gender, Work & Organization Published by John Wiley & Sons, Ltd

## 1 | INTRODUCTION

How different genders behave and communicate at work shape work cultures. Scholars have looked into how work cultures can be gendered (e.g., Hall, Hockey, & Robinson, 2007; Kelan, 2008; McDowell, 1997, 2009; Valentine, Jackson, & Mayblin, 2014; Warren, 2016). For example, Hall et al. (2007) and Valentine et al. (2004) investigate the embodied experience of female and male fire brigades. McDowell (1997) looks into various ways of doing gender in the services sector. Warren (2016) studies the masculinities of male workers in the context of surfboard-making. They all found that masculinities and femininities are constructed through labour processes which are deeply emotional and embodied.

Online virtual workplaces where people collaborate and communicate are no exception to such gendered work cultures. One can observe such gendering processes in the development of free/libre open source software (FLOSS). FLOSS such as the Mozilla Firefox browser, is the result of many incremental innovations contributed by people located in different places around the world. Contributors communicate mainly online via email, on the Internet forums, mailing lists, Internet Relay Chat (IRC) or on social media. Some of these channels are formal, used by organizations to communicate work-related information, and others informal; some private, others public. While the open characteristics of FLOSS, which include the freedoms to study, change, distribute and redistribute source code, are warmly embraced, it has been argued that there are numerous barriers stopping people from participating. Gender issues have been highlighted as one. For example, the number of female contributors is extremely low, lower than the average in the information and communications technology (ICT) sector (see Ghosh, Glott, Krieger, & Robles, 2002; Nafus, Leach, & Krieger, 2006a). Gendered or even sexist language has been observed on the communication channels mentioned above. While the 'open' nature of FLOSS has generated a highly valued online collaboration culture and network effect (Benkler, 2006), such male-dominated networks have produced a culture predicated on male sociability and stabilized as masculinist space (Lin, 2005; Nafus et al., 2006a; Nafus, Leach, & Krieger, 2006b; Reagle, 2013; Valentine et al., 2014).

Barriers for women to take part in FLOSS development have been discussed (see, e.g., Lin, 2005). For example, in terms of work culture and work–life balance, it is difficult for many women to juggle different responsibilities in life while doing a job that requires long hours of commitment to coding. Many women also do not have the skills and knowledge of some home-baked solutions circulated only in a closed male-centric geeky free/open source software community; most female developers seem to gain their knowledge from the formal school curriculum (Lin, 2005). Additionally, many have reported experiences of discriminating, abusive, sexist language used online and/or offline (e.g., the Read The Fxxx Manual (RTFM) culture), the lack of 'mentors' and role models, and the male-dominated competitive meritocratic culture. The term 'brogrammer' was also coined (Fores, 2012; Hicks, 2013; Kumar, 2014) to label a rather misogynist behaviour that male hackers sometimes exhibit when they socialize. All of these contribute to a women-unfriendly environment.

Numerous actions have been taken by larger FLOSS communities and ICT companies to improve the gender gap. For example, the largest GNU/Linux distribution Debian has created the Debian Woman project[1] to encourage the participation of more female members and provide mentoring. Other FLOSS projects such as GNOME, KDE and Mozilla have also initiated similar groups, for example, GNOME's Outreachy project[2] and the WoMoz project.[3] Multinational IT companies such as Google that use, develop and share FLOSS have also sponsored activities and opportunities to bridge the gender gap, for example, Google-Sponsored Women Techmakers[4] and the 'Made with Code' initiative.[5] Also, non-profit companies such as Code First: Girls[6] delivered training to young women across the UK to increase the number of women in the tech industry.

Despite these efforts and the consensus of the imbalanced gender distribution in the field, the questions about how such gendered environments emerge and shape software development processes, experiences of developers and consequently the quality of work remain largely unanswered. The above-mentioned initiatives tend to focus on skill training that is visible on the surface of this lack of gender diversity, rather than changing (sexist) behaviours that are based on deeper but often invisible cultural differences. More thorough academic debates and scholarly research that aims to identify hidden causes for the gender imbalance will therefore strengthen the urgency of the need for diversifying the talent pool.

Following this line, this article will examine the gendered work cultures in FLOSS development through understanding how communications between FLOSS developers take place in an online space. Social meanings and societal values are embedded in languages. 'Language is the most intense and stubborn fortress of sexist assumptions' (Sontag, 1973, p. 186). Analysing languages therefore can deconstruct the linguistic devices created to facilitate gendered cultures in a certain place (online and offline) and helps understand how a virtual masculinist work space is produced and stabilized through the repetition of certain communication behaviours (Valentine et al., 2014). Through analysing the communications on public mailing lists (the language used specifically), this article helps understand how developers behave and communicate, and how their communications between developers reflect laddish culture and reproduce it in virtual workplaces.

## 2 | BUG REPORTING, DEBUGGING AND BRICOLAGE

The most effective way of understanding how a gendered online work environment is created is to look into everyday narratives and common work practices. Bug reporting and bug fixing are essential tasks in software development, but they are more than signalling software malfunctions and fixing them. Bug reporting and bug fixing involve a process of identifying, raising, discussing and resolving problems. It is a practice that requires both cognitive and communicative skills, and a process that is highly socio-technical, subjective as well as interpersonal (den Besten & Dalle, 2014).

In FLOSS development, identification and resolution of problems is commonly carried out with the help of so-called 'bug tracking systems'. As the name suggests, these systems allow for recording, reporting, tracking, triaging and resolving ('patching') of bugs. Reporting is an activity that engages a lot of people, ranging from aspiring developers to computer-literate end-users (Crowston & Howison, 2006). The sentence 'given enough eyeballs, all bugs are shallow' (Raymond, 1998) highlights that an open bug-reporting environment is seen as central to the success of FLOSS. Yet, as Villa (2003) points out, it takes time and effort to deal with all the reported bugs. To come up with a solution to the problem that a bug report identifies is not a straightforward process. Although there are defined steps to take, communications, decision-making and problem-solving can take place in many forms and in many directions.

A bug needs to be confirmed first, then verified (if the problem reported can be replicated), rated (evaluating the importance and difficulty of the problem at hand) and finally, people with the appropriate roles, responsibilities and expertise will be appointed to work on the problem. In the debugging process, in order to solve the problem, all parties with an interest in its resolution need to identify tools and resources available from different materials and sources. This is what Villa (2003) calls 'triage' or what Cunha (2005) terms 'bricolage'.[7] These FLOSS developers are '*bricoleurs*', 'someone who works with his hands and uses devious means compared to those of a craftsman' (Levis-Strauss 1962, p. 11), using 'instruments he finds at his disposition around him (Barnard, 1996, p.192). … finds at his disposition around him, those which are already there, which had not been especially conceived with an eye to the operation for which they are to be used and to which one tries by trial and error to adapt them, not hesitating to change them whenever it appears necessary, or to try several of them at once, even if their form and their origin are heterogenous—and so forth' (Derrida 1978, page 360). In short, bricolage can been seen as 'using or adapting material which is at hand … improvising, putting together found or second-hand items to create a look or ensemble' (Barnard, 1996, p.192).

But all these activities, ranging from identifying, classifying something as a bug, to choosing a tool to perform a task, could be political. As Borgman (2003) notes, 'Multiple types of problems exist, as do multiple types of knowledge that may contribute to solving them' (p. 99). It is this cognitive and problem-identifying and problem-solving process where one may observe different gender cultures reflected. Where to borrow ideas and concepts and tools from, and what strategies to use to gather resources are also profoundly rooted in someone' s cultural backgrounds, influenced by someone's social circles. As Derrida (1978) argues, 'bricolage is the critical language itself' (Derrida 1978, p. 360). Bricolage, from this perspective, can be political and gendered.

## 3 | METHODOLOGY AND METHODS

This article uses the development of the Mozilla Firefox web browser as a case study to demonstrate the gendered bricolage process. To study how bug reporting and bug fixing is performed and played out, and what (strategic) communications are involved to facilitate the work, we conducted a case study of the 'Heidi bug' drawing on the narratives in the bug tracking system Bugzilla. We collected all the narratives in relation to the Heidi bug and conducted an analysis on the narratives. We started by looking at how questions are raised and phrased on the bug tracking system, a common tool that has long been employed by developers to communicate with one another in an open and distributed environment to manage the process of bug reporting, triage and patching in the development and maintenance of software projects. An online forum discussion on the [WoMoz] mailing list was instigated to understand how such gendered language is perceived by female developers.

Unlike proprietary software development, problem-solving in FLOSS development is transparent. A formal (and sometimes also informal) record of the problem-solving process is kept, archived and, mostly, publicly accessible. For instance, most FLOSS projects make use of bug tracking systems to keep a record of problems identified and features requested with respect to their projects and of the efforts that have been made to resolve them. As we already indicated, the process of identifying, reporting, discussing and resolving of problems resembles a process of bricolage, and we study the communication and logs on the Bugzilla bug tracking system to investigate the gender politics embedded in this process. The threads on the Bugzilla system offer a detailed account of problem-solving activities with documented time, dates and narratives. They exemplify how programmers perceive, classify and frame a problem and how they come up with solutions to it. Through narrative analysis, we identify signs of sexist language in the bug description or in the subsequent discussion and look for indicators of the effect that the heterohegemonic pose (Frank, 1987) has had on the success or failure of the process of bricolage that accompanied the bug.

We then shared our preliminary finding from the analysis on the [WoMoz] mailing list (a mailing list for female Mozilla developers) in 2013 to understand how this gendered event was perceived by the members of the Women and Mozilla Project[8] who were more sympathetic towards gender ethics. After circulating a message on the [WoMoz] mailing list, we received responses from six developers who started a discussion around heterohegemonic jokes, language and the brogramming culture. This online discussion with the members on the [WoMoz] mailing list revealed that people have different opinions towards what is considered as sexist, but men and women like different types of jokes.

## 4 | BUGZILLA

The Mozilla foundation, which drives the Firefox browser development, uses Bugzilla as their bug tracking system to organize information flows in a process of bricolage. In Bugzilla, bug reports are assigned a status, which indicates the progress that has been made with the resolution of the bug. Bugs reported by outsiders will have the status 'unconfirmed' and only insiders with the so-called 'can-confirm privilege' can turn this status into 'new'. Next, someone will be asked to take ownership of the bug. If this person accepts, the bug turns from 'new' into 'assigned'. After that the bug may or may not be 'fixed' and once it has been fixed the solution will be verified before the bug report is closed (cf. Masmoudi, den Besten, de Loupy, & Dalle, 2009).

It is important to note that not all bugs can be fixed. Sometimes, a bug is orphaned (remains unclassified, unanswered, unresolved) or abandoned. One of the reasons for abandoning a bug is that it reports a known issue on which people are already working. In that case, it will be declared a 'duplicate'. Alternatively, it may be decided that the problem reported is not a real problem and in that case the bug will be closed with a line saying 'works for me'. Apart from changing the status of bugs, another important activity of the people participating in the bug resolution process on Mozilla's Bugzilla is to map out the mostly technical dependencies among bugs. A bug is said to 'depend on' another bug if the resolution of the former is only possible after the issue identified in the latter has been resolved.

Inversely, a bug is said to 'block' another bug if the resolution of the latter is only possible after the issue identified in the former has been resolved. Together, the dependencies among bugs form a bug report network (Sandusky, Gasser, & Ripoche, 2004). Hence, by identifying dependencies, participants contributing to the discussion forum associated with a bug work together to resolve this bug in a network. Once verified and confirmed, the bug becomes a shared concern for the people working collectively and collaboratively on it.

Since software systems are complexly networked and connected, related bugs are all relevant and as such included in this network. Besides, bugs are classified according to the priority that should be given to them in relation to other bugs. In particular, that can be nominated to be added to a list of bugs to be resolved before a certain milestone is reached (typically a future release of the software). In short, the log of past actions and comments about 'bugs' and 'issues' allows us to trace strategic communication and bricolage that take place.

## 5 | THE HEIDI BUG

'*Heidi*' is the alias of a bug reported by Christopher Aillon in 2002.[9] Typically bugs that are reported are assigned a unique identification number (in this case 121084). Aliases are sometimes provided to allow people to easily distinguish and locate the bug report. In this case '*Heidi*' refers to the German model Heidi Klum, whose association with the bug is to illustrate the problem of images failing to be reloaded on certain occasions. The bug description contains presuppositions that are indicative of the adoption of a heterohegemonic pose.

Here, we provide a chronological description of the event to demonstrate how a process of bricolage was performed. We draw on the bug report data of the *Heidi bug* and compare it to the processes of bug resolution in two other related bugs, which address similar issues.

The first is *bug 98890*, a bug reported by Guillaume Filion several months before Aillon reported his. Even though it was reported earlier, this bug was eventually classified as a duplicate of the Heidi bug (rather than vice versa). The second is *bug 93015*. This bug is related to the Heidi bug via *bug 83774*. It is worth noting that the people working on *bug 93015* had already resolved many issues described in the Heidi bug, and that Jim Dunn, who was then assigned the bug, made the connection with bug 83774 and managed to recruit Boris Zbarsky to devote his attention to the remaining issues and fix the bug.

Figure 1 is a section of the bug report network around the Heidi bug, which provides an overview of the complex communications between developers through a bug tracking system. Each bug report is associated with a forum for asynchronous communication among developers. Dependency relations are noted in the bug reports as soon as they have been identified.

Figure 2 is a partial bug report network that shows the dependency between the Heidi bug (report number 121084 in bold) and other related bugs we studied in this article. The dates of reporting are indicated in parentheses below the bug identifiers for each single report. Some bug reports are mentioned as 'collections' and the dates of

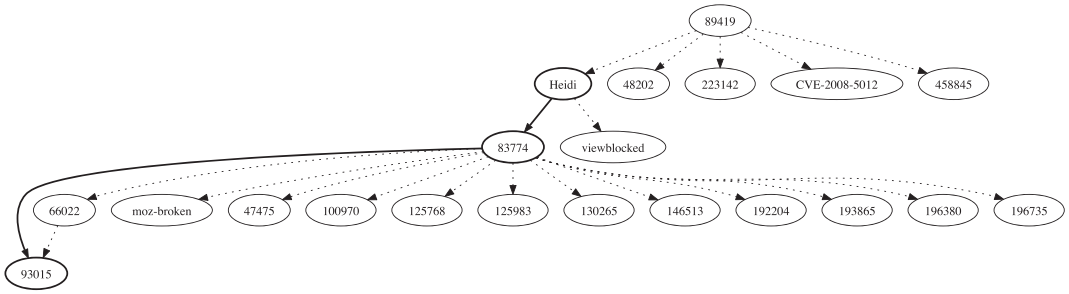


**FIGURE 1** A partial bug report network of bugs that are technically related to the Heidi bug. The solid lines identify links between bugs mentioned in the main text.

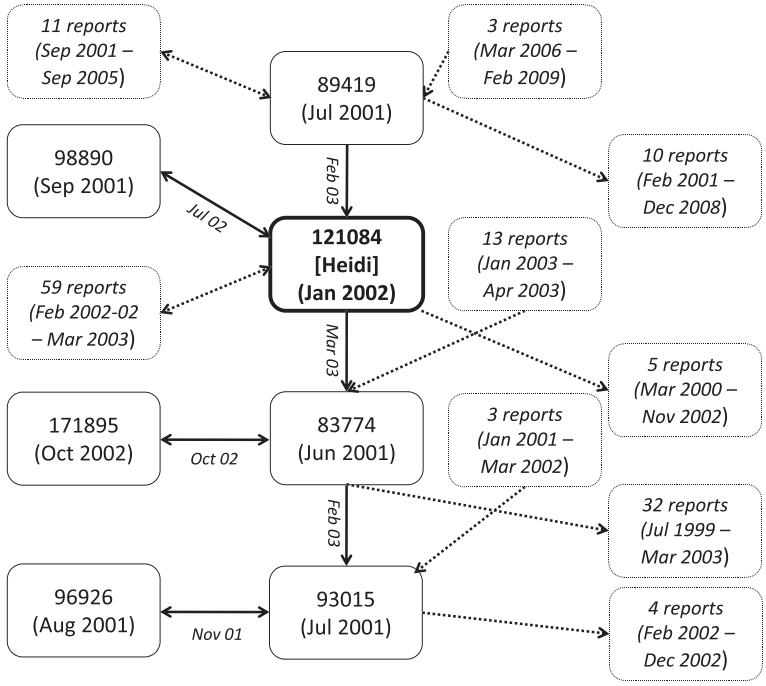


**FIGURE 2** A collapsed view of the full bug report network associated with the Heidi bug, together with the dates when individual reports were created and when dependencies between the bugs were established

reporting are shown as a range. Double-headed arrows are used to show duplication of bug reports and single-headed arrows are used to show dependency. The Heidi bug has 60 duplicates, and it directly depends on the resolution of the report (89419) and blocks six other reports.

## 6 | GENDERED TALK AS A STRATEGY FOR BRICOLAGE

Let us compare two bug reports: one by Christopher Aillon and one by Guillaume Filion. The summary that Christopher Aillon initially gave for his problem was 'Heidi Klum doesn't like being compared to the cache'. In contrast, Guillaume Filion chose to describe the same issue as 'The URL of the image is showed instead of the image.' While Filion's summary was more informative, Aillon's had a humorous twist.

Judging by the names (ID of contributors), among the 69 people participating in the discussion on the Heidi bug, only one was female.

Among the 11 people discussing Filion's bug report, no one appeared to be female. Note that Aillon gradually added informative content to the bug summary after he engaged fellow developers. He first reported the bug in January 2002. In April 2002 he added the explanation 'Images requested twice -> "The image cannot be displayed, because it contains errors." message.' In June, the joke about Heidi was removed from the summary and replaced by the alias 'Heidi' for the bug.

It appears that Aillon with his Heidi bug managed to gain more attention than Filion with his earlier description of the same problem. Part of the reason that Aillon succeeded might be because he made the problem look more interesting. At the same time, from the Heidi bug history, it was clear that Aillon and others working on the bug went into great lengths to recruit people for his problem (discussing it on IRC, adding people to the CC-list, establishing dependencies), at any rate more than what Filion did. Moreover, Aillon was also proactive. For a long time, the bug failed to

get full attention from core developers. In March 2002, Aillon complained about the lack of attention the issue received:

> *<sl8r> matti: i'm saying that at least for me this is a highly visible bug. i can't believe that nobody else is seeing it. there are no dupes etc.*
>
> *<sl8r> caillon: yeah … and i can't nominate it for anything, i'm not 'sufficiently empowered' … *sigh**

After this complaint, David Hallowell noted a month later that the problem also appeared on Windows and after that people started to identify duplicates. All this activity convinced Stuart Parmenter to change the target milestone from the assignment 'future' that he had given in February to the more concrete 'mozilla1.1alpha' in April. In July, Jason Bassford, who had been working on Filion's bug, noticed the Heidi bug and pointed out that it was a duplicate of Filion's. Christopher Aillon, however, did not appreciate this action:

> *Dude, don't dupe this bug to the other just because it has a lower number. This one has more people on it that care about the bug. It's been triaged already and people know how to find this bug and not the other. If the other is the same as this, then dupe that one to this one. Re-opening.*

The fact that Filion had even less success in getting attention could be due to the more neutral description that he chose, but just as well it could be because fewer efforts were made to make the bug known to Mozilla developers. However, this episode exemplifies how strategic communication (humour in this case) is brought into a bricolage process to solve a problem and reflects certain male-centred values implicitly. After identifying a bug, the reporter, instead of providing a plain description of the fault, provided an anecdote containing a heterohegemonic presupposition in his pitch of the problem to attract more attention. The light-hearted tone gave the bug report much currency to flow in the busy and complex bug tracking system. However, while successfully securing the attention (resources) from fellow developers on this particular bug, the content of this bug report reflects the heterohegemonic value that is prevailing in mainstream society. For example, the use of 'Dude', and a heterosexual male gaze on a highly sexualized and objectified female celebrity model, and the rather patronizing tone throughout the message. The gendering process started when the bug reporter assumed fellow developers would share his worldview or background belief (assuming that the blonde female model is a symbol of beauty and a pleasant image on a computer monitor). This presupposition was verified (thereby exacerbating the already gendered situation) when much-needed resources (attention from fellow developers) were allocated. It was confirmed that the presupposition was mutually known by the reporter and the addressee; both parties considered such an utterance appropriate in the context. The well-received bug report reaffirms that such a brogrammer's view and culture is accepted and shared.

There had been other signs of heterohegemonic culture. It turns out that the source code to which the patches were applied in bugs 83774 and 93105 was called 'libpr0n'. A Mozilla forum gave the following explanation for this name:

> *The main goal of the library is to render pornographic images in an efficient way. Plus, the name 'imglib2' is boring.*[10]

As developers are aware that the Bugzilla bug reports are public, the casual use of gendered cultural references might be indicators of bigger underlying issues. In this particular case of the so-called 'Heidi bug', there had been both positive and negative effects on the capacity of the bug to attract attention and resources. The way through which the 'Heidi bug' and other bugs closely related to it were fixed seems to suggest that the 'small and casual talk' used to 'frame' a question is not trivial at all. In fact, the adoption of a heterosexual masculine pose in a specific context might help bricolage. But, the very same strategy may not be replicable in other contexts (as it has not been seen elsewhere). Some fellow developers remarked that the gendered attitude and behaviours are not welcome for resolving a bug (as seen in the discussion in the next section).

## 7 | RESPONSES FROM THE MOZILLA COMMUNITY

As a way of reflecting and reciprocating, the researchers shared the preliminary analysis on the [WoMoz] mailing list in autumn 2013. [WoMoz] is a sub-community of Mozilla established to encourage women to participate in the development of the FLOSS Mozilla Firefox browser. Discussing this case study on [WoMoz] therefore could enact a much-needed reflexivity and reciprocity into online research (Morrow, Hawkins, & Kern, 2015). The researchers decide to report the remarks from the community verbatim and at length to reflect the kind of dynamics and conversation that people would have in a virtual workplace.

Few members had heard of the so-called 'Heidi bug' (or the 'Heidi Klum bug'). There were short responses describing the bug as '"old and quite simple" — (i.e. technically uninteresting, and a bug reported 10 years ago cannot be regulated by the community guidelines[11] where sexist languages are not allowed)'.

One thought the 'Heidi bug' is humorous but problematic:

> *An informative title can be easy for people to write off 'oh I see from the summary this doesn't matter to me' whereas a humorous title can get people to read more of the bug, and once they're in the bug, they can't help but try and solve the problem.*
>
> *I honestly can't think of any other examples like this where the bug is humorous but it's a real issue. The other ones I can think of are humorous entirely.* (MC1)

Another said:

> *The reporter of the bug gave an URL to an adult site to demo the bug. It's disputable, given that the bug was apparently very frequent. There was audience to the bug and it was solved. Could not say if the sexist jokes helped or not.*
>
> *The expected results made me laugh though it can be considered as sexist. But it's not really that bad (at least he offered lunch), it could have gone worse (especially in the comments).* (FA1.)

Another more detailed account was provided:

> *I can't say in my years developing code that I have come across other examples to give you, but I'll give you my thoughts on this, what I have, and I hope it helps.*
>
> *While I do notice that the 'representative bug' is not the oldest one[1], as is convention (but this convention is by no means an invariant of bugzilla), but the 'funny' one about Heidi. It is possible that the offensive nature[2] of the test image & humorous text made it more memorable than the original bug.*
>
> *It is possible the Heidi bug was more memorable than the original bug because it could be viewed as a computer science in-joke. Computer graphics/image issues often need images of figures, especially figure& face for test cases (dynamic range, multiple types of curved lines, etc). While I personally had no love of seeing Playboy centerfolds used casually in my university graphics classes[3], it is something that is still baked into the field's history, so 'Heidi Klum as a modern day centerfold' may be a sly reference/homage to the use of centerfold images for testing dating back to the 1960s.*
>
> *The closest I have encountered personally might be not gender but sexual orientation based[4], and there was no particularly special reaction when we triaged the bug.*
>
> *I also notice that the Heidi bug has 59 duplicates, which would make it a highly visible problem for developers. In my experience, often reported, highly visible, and thus memorable bugs do acquire names. However, the name is usually associated with a real person, often the bug finder or the bug writer. Examples such as 'the mjrosenb memorial bug' or 'the katz bug' come to mind. One could argue that the bug takes on the gender of the person associated with the bug, but that might be a bit of a stretch?*

> *I suspect that may be a topic where there is more unconscious bias than overt bias in play, and you may need statistics on a large data set to prove it, but I'm afraid I do not know how one would extract that kind of nuance from bugzilla.*
>
> *I suppose we could file a graphics issue using a saucy image of a man for a test case and another with a gender-neutral-meme and compare the results. Would that make a good case study? We just turned on apzc on metrofx, so I've got graphics issues to spare right now.*
>
> *[1] first bug filed was https://bugzilla.mozilla.org/show_bug.cgi?id=98890*
>
> *[2] bug reporter notes offensive image https://bugzilla.mozilla.org/show_bug.cgi?id=121084#c2. As of this writing, the original image link 404s so I can't tell how much resemblance it bears to the iconic test images like [3]*
>
> *[3] http://en.wikipedia.org/wiki/Lenna*
>
> [4] https://bugzilla.mozilla.org/show_bug.cgi?id=903953#c2 *(AMN1)*

However, some disagreed that this way of framing the bug is 'sexist' and inappropriate:

> *I know we all have different feelings here, but I am uncomfortable with calling a straight male who likes to look at women 'sexist'. I know I sure like looking at men (well and women too, we're gorgeous ladies!) I'm totally comfortable with calling it inappropriate in x,y,z situations because it *does* offend some people.* (MC2)
>
> *To follow up on an earlier comment (not quite in thread, apologies) I'm not saying it's not okay for men to find women attractive — I find it objectifying and inappropriate in this *context* though. That's just me. Anyway. It's a really interesting conversation.* (LS1)

It also seems that the Mozilla community has developed quite a smooth working relationship so developers know each other. So a more experienced respondent, instead of accusing their male counterparts, rationalized:

> *I actually know Chris and would be happy to ask him if he remembers how he ended up choosing this example. One thing that occurred to me is that we test against the top 50 or 100 or whatever popular sites, and many of these are porn. So if we see a bug that refers to a porn site it could simply be that it was part of the set of test sites and that's why the error was caught there. If that is happening and that is the case this is a chance for us to give advice on whether or not we think the bug should be reproduced on a non-porn site, and then filed that way.* (MC2)
>
> *Here is the comment in the bug that explains why the older bug was duped to the newer bug — https://bugzilla.mozilla.org/show_bug.cgi?id=121084#c81*
>
> *This happens, you can see the dupe wasn't found for a long time, and the relevant people had seen the newer bug but not the older bug. Think that a developer notices something, files a bug, pings his/her developer friends to help out on it — that bug has obviously got instant attention and work started on it.* (MC3)
>
> *I agree with Majken (now that she explained why the image could have come from an adult site …). I laughed at the idea of Chris expecting to have dinner with Heidi and so on. It's just the kind of thing I would say, totally quirky humor …*
>
> *There was no further comment on the kind of picture or Heidi herself in the comments.*
>
> *Do you know that, in french, 'bug' and 'problem' are male words whereas 'solution' is a female word?*

> *E.g.: 'J'ai un bug!' 'J'ai trouvé la solution.'*
>
> *Think of that the next time you file a bug :)* (FA1)

But this discussion has raised some attention on developers' bug-reporting behaviours:

> *I agree with the point of this, is this something that happens more than once? How often, is it the same people?* (MC2)
>
> *The 'Heidi' bug looks to me as though the original reporter of the bug just didn't bother to find a non-adult example since he felt comfortable posting adult content (and his accompanying 'humorous' STR) in a Mozilla bug report. This isn't directly sexism but it's certainly an example of privilege and comfort of being the straight guy in a primary straight guy space. So if the intent of bringing up this thread was to discuss whether bugs like this might alienate potential community members in an open source project I would say that (besides it being an old and resolved issue) yes, it might be and it's very likely to come up (see the bug above) on occasion in a project.*
>
> *I like how Liz dealt with the bug above, a recent example, in that she called it out for what it was and since there was no _real_ issue, the bug has remained closed. We don't monitor bugzilla for stuff like this, or at least not in any official capacity, but don't think that there's not unintentional leveraging of privilege and power imbalances playing out in our bug tracker sometimes that might accidentally scare off or set the wrong atmosphere for a potential contributor.*
>
> *Good things to think about when filing our own bugs, and when helping others file theirs.* (LB1).

## 8 | GENDERED LANGUAGE AND ITS HIDDEN MEANING

Valentine et al. (2014) emphatically argue that:

> *the public expression of the most blatant forms of gender prejudice, sexism persists and is manifest in subtle ways. As a consequence, it can be difficult to name and challenge with the effect that patriarchy as a power structure which systematically (re)produces gender inequalities is obscured by its ordinariness.* (p. 401)

In an environment where distributed collaboration takes place, it is even more challenging to detect invisible sexism hidden in technical languages, and to manage the relationships between actors from different backgrounds. To facilitate online collaboration, some languages have been selectively or strategically used to motivate or enact participation. However, some of these languages also reflect a hegemonic masculine culture that dominates in the computing field. Seemingly innocuous jokes actually reflect a male-centric position and may lead to invisible sexism (Bemiller & Schneider, 2010; Shifman & Lemish, 2010a, 2010b).

In light of the responses received from the Mozilla community, most of the respondents agreed that the title of the 'Heidi bug' report was humorous and was a form of strategic communication. While few considered it sexist, some did notice the cultural meaning hidden in the technical discussion and the subtle framing. The cultural reference employed in the 'Heidi bug' report explicitly reflects the brogrammers' interests, beliefs and aesthetic values. The attention received suggests a shared brogramming culture supporting such a value system. Had it been a woman reporting the same bug joking about problems of seeing half-naked men, would it be treated in the same way in a brogrammer community prevailing with heterosexual masculine culture, the kind of 'hegemonic form of masculinity prevalent in society, which determines who counts as a "real" man' (Connell, 1995; Frank, 1987; Kelan, 2008)?

Not only does the brogramming culture exist online, but also offline, for example, at FLOSS conferences, where male speakers outnumber female. While these tongue-in-cheek jokes told in a speech given by a male speaker can be considered as harmless, they reflect certain hegemonic masculine values that dominate in society. The speakers tell these jokes to amuse like-minded conference participants/delegates, but these jokes may not amuse women (or may offen them even). These can be seen as the kind of microaggressions or microinvalidations referred to by Mitchell (2016) in her study of lived experiences of women in geek communities. Those who do not share the sentiment are excluded from the ambient.

Our observation echoes Wikhamn and Knights (2013) who argue that hegemonic masculine discourses are reproduced rather than challenged by open innovation. In their research, they found that when masculine norms govern an open innovation setting, it is difficult to incorporate alternative (feminine) discourses when acting within a strong masculine hegemony (Wikhamn & Knights 2013.).

By no means is this article suggesting a censorship on humour from a heterosexual male perspective; nor is this article arguing that there are gender differences in language use in terms of dichotomies such as public vs. private and informational vs. affective (Talbot, 2010). Through focusing on the construction and performance of gender in discourse (Talbot 2010.), this article merely demonstrates that languages or presuppositions or anecdotes cited for strategic communication can reflect certain types of cultures and they dominate the stage/field/platform. The work cultures in a distributed online environment can still be hegemonic masculine, as noted in Wikhamn and Knights (2013), and we need to pay more attention to such patterns and languages used to ensure that different voices (women, LGBT (lesbian, gay, bisexual and transgender), disabled, other minorities) are not excluded or muted in these environments. It also demonstrates that software development, like other types of works, are not neutrally technical jobs, but full of gendered performance and gender politics (as argued in Kelan, 2008). This existing body of work challenges heterohegemonic masculinity, and is important for achieving more inclusive, tolerant and equitable workplaces. In a similar vein, we are emphasizing the need to diversify the languages so as to have more inclusive work cultures on the Internet. While feminist scholars have challenged gendered jokes and humours, it seems feminist ethics has not yet been fully integrated into the computer science or software engineering curriculum. This is an effort to raise awareness by discussing the discourse emerging from the everyday talk in distributed software environments.

## 9 | CONCLUSIONS: SOFTWARE DEVELOPMENT AS A GENDERED FORM OF WORK

Herring (1996) powerfully argues that:

> *'netiquette' (a combination of 'network' and 'etiquette') norms are both a moral and a political dimension, in that they are founded on systems of values and judgments which may vary according to different groups of users.* (p. 115)

Her research has found that in computer-mediated communication, 'it is typically the most powerful or dominant group whose values take on a normative status' (ibid.). She further submits that:

> *the gender differences in public discourse on the Internet are not randomly distributed across individuals, but rather display a systematic pattern of distribution with male users as a group tending toward more adversarial behavior and female users as a group tending toward more attenuated and supportive behaviors.* (p. 137)

Many other feminist scholars have also questioned whether digital materials or artefacts are imbued with traditional gender identities, norms, roles, symbols, meanings and proposed gender-sensitive design approaches (Bath, 2007; Kember, 2003; Rommes, Bath, & Maaß, 2012; Rommes, van Oost, & Oudshoorn, 2001). For example, in her work

on AI (artificial intelligence), Kember (2003) talks about predictable and stereotypical female forms (e.g., electronic assistants and robot nurses). In her study of the semantic web, Bath found modern information systems and computational artefacts often are designed with

> *binary assumptions about women and men [which] are not reflected [upon] or the (gender) politics of [a particular] domain is ignored. Thus, the existing structural-symbolic gender order is inscribed into computational artifacts and will be reproduced by [their] use.*[12]

This article looks into how seemingly technical activities such as bug reporting and bug fixing are actually gendered. As argued earlier, framing a problem (classifying something as 'a matter' or 'an issue') is a social process. In the context of bug resolution, it involves efforts to gain the attention and interest of people who might have the key to solving the problem identified. Consequently, one might interpret *brogramming* behaviour as an attempt by the 'bricoleur' to create a common ground and make the bug that needs resolving seem fun and appealing. Unfortunately, what looks appealing to some group will offend others and the repeated search for a common ground might engender a process in which the purported interests of the group that is appealed to become more and more homogenous as those who feel offended leave or refrain from entering.

Fores (2012) argues that it is common to observe misogynist behaviour among hackers when they socialize. This article strengthens this argument by looking more deeply into everyday hacking practices, and theorising the observed asymmetries in the field from a feminist perspective.' Strategic communication involved in performing bricolage that contains presuppositions of mainstream interests and beliefs reproduces heterohegemonic masculine cultures in society in the virtual workplaces, as also argued in Wikhamn and Knights (2013).

While the authors are aware that the analysis of a bug done in this article cannot be generalized to the whole software development process, our analysis on the communications in a publicly accessible bug tracking system does suggest that the software development process can be highly gendered. The gendering process started from the initial problematization and framing of a bug (i.e., cultural references cited) and the sense-making discussion processes even though the extent to which they play a role remains unclear. Brogramming languages and behaviours are often made hidden or invisible in the highly technical talks, such as the one examined in this article. But such seemingly pragmatic mindset of bricolage held by brogrammers in fact reflects the gender norms prevailing in society (e.g., sexualization and objectification of women's bodies and static gender roles).

This highlights the tensions and conflicting values in the FLOSS development: on the one hand, the culture of 'openness' in FLOSS welcomes valuable practices for flexibly managing different resources for problem framing, collective problem-solving (e.g., teamworking, networking and bricolage), but on the other hand, the prevailing 'brogramming culture' leads to undesirable 'gendered arrangements of values [that] perpetuate dominance even in cases where no intimidation is intended' Herring 1996., p. 137).

How can we inject the findings from our study back to the apparatuses of computer ethics? Our work contextualizes bricolage in a male-dominated decentralized online work environment such as the FLOSS communities. It bears a lesson for IT managers: the kind of bricolage observed in the FLOSS development is a useful skill and practice, but on the other hand, gender should be strengthened as an important area in computer ethics education to eliminate brogramming culture. Whilst computer ethics has been developed into an academic and practical discipline, we need more rigorous methods to integrate feminist ethics and gender-sensitive languages into computer ethics education (Adam, 2000; Adam & Ofori-Amanfo, 2000). That said, we need to continue to update the existing and invent new languages and vocabularies to delineate or develop a field of inquiry in relation to gender so that we can engage in intellectually sound discussion and make computer ethics education practically relevant to professionals working in information systems (Stahl, Eden, Jirotka, & Coeckelbergh, 2014). Some community-based efforts of creating community 'codes of conduct' (e.g., the Debian Code of Conduct[13]) or a list of harassment incidents (the Geek Feminism Wiki[14]), or a statement illustrating a shared community identity (e.g., the Debian Diversity Statement[15]) may also be a solution to documenting such undesired treatment. Lastly, conferences should increase female speakers or reject those sexist jokes that please male delegates more than female. Proactive actions must be taken by conference organizers or community coordinators in addition to some affirmative action on paper.

To overcome the methodological limitation of this article, future studies should include interviews with more developers about how they work, communicate and compose email messages when reporting bugs to understand online FLOSS work cultures and environments more.

## DECLARATION OF CONFLICTING INTEREST

The authors declared no potential conflicts of interest with respect to the authorship and/or publication of this article.

## ACKNOWLEDGEMENTS

The authors would like to thank the respondents who joined the discussion and provided their valuable opinions on the [WoMoz] mailing list, the anonymous reviewers and editors who provided helpful guidance for revising the manuscript, and friends in the Debian GNU/Linux community who shared their life stories as hackers and free/libre open source software developers.

## ENDNOTES

[1] to encourage the participation

[2] https://gnome.org/opw/

[3] http://www.womoz.org/

[4] https://www.womentechmakers.com/

[5] https://www.madewithcode.com/

[6] http://www.codefirstgirls.org.uk/

[7] The concept is subject to a lively debate (see, e.g., Baker & Nelson, 2005; Perkmann & Spicer, 2014).

[8] http://www.womoz.org/

[9] The Heidi bug can be found at https://bugzilla.mozilla.org/show_bug.cgi?id=Heidi. In order to locate bugs in Mozilla's Bugzilla bug tracker the URL of the bug is given by the bug's ID number or alias affixed to the string 'https://bugzilla.mozilla.org/show_bug.cgi?id=%E2%80%99.

[10] http://forums.mozillazine.org/viewtopic.php?t=16103 (last accessed 20 May 2013).

[11] The issue is the use of a sexist joke to draw attention to an actual technical issue — seems likely to violate some section of our community participation guidelines: https://www.mozilla.org/en-US/about/policies/participation/but this bug is more than 10 years old, so the policy was not applicable at the time.

[12] http://readwrite.com/2008/08/06/will_the_semantic_web_have_a_g

[13] https://www.debian.org/code_of_conduct

[14] http://geekfeminism.wikia.com/wiki/Online_harassment

[15] https://www.debian.org/intro/diversity

## ORCID

*Yu-Wei Lin* http://orcid.org/0000-0001-9798-5165

*Matthijs den Besten* http://orcid.org/0000-0002-4361-4278

## REFERENCES

Adam, A. (2000). Gender and computer ethics in the Internet age. The CPSR (Computer Professionals for Social Responsibility) Newsletter, 18(1). Retrieved from http://cpsr.org/prevsite/publications/newsletters/issues/2000/Winter2000/adam.html/

Adam, A., & Ofori-Amanfo, J. (2000). Does gender matter in computer ethics? *Ethics and Information Technology*, *2*, 37–47.

Baker, T., & Nelson, R. E. (2005). Creating something from nothing: Resource construction through entrepreneurial bricolage. *Administrative Science Quarterly*, *50*, 329–366.

Barnard, M. (1996). *Fashion as communication*. London, UK: Routledge.

Bath, C. (2007). 'Social' robots & 'emotional' software agents: Gendering processes and de-gendering strategies for 'technologies in the making'. In I. Zorn, S. Maaß, E. Rommes, C. Schirmer, & H. Schelhowe (Eds.), *Gender design IT: Construction and deconstruction of information society technology* (pp. 53–63). Wiesbaden, Germany: VS-Verlag.

Bemiller, M. L., & Schneider, R. Z. (2010). It's not just a joke. *Sociological Spectrum*, *30*, 459–479.

Benkler, Y. (2006). *The wealth of networks: How social production transforms markets and freedom*. New Haven, CT: Yale University Press.

den Besten, M., & Dalle, M. (2014, July). Coordination by reassignment in the Firefox community. In *Proceedings of ECIS 2014*. Tel: Aviv.

Borgman, C. L. (2003). Designing digital libraries for usability. In A. P. Bishop, N. A. van House, & B. P. Buttenfield (Eds.), *Digital library use: Social practice in design and evaluation*. (pp. 86–118). London, Cambridge, MA: MIT Press.

Connell, R. W. (1995). *Masculinities*. Cambridge, UK: Polity.

Crowston, K., & Howison, J. (2006). Hierarchy and centralization in free and open source software team communications. *Knowledge, Technology & Policy*, *18*, 65–85.

Cunha, M. P. E. (2005). *Bricolage in organizations (FEUNL Working Paper Series No. 474)*. Lisboa, Portugal: Universidade Nova de Lisboa, Faculdade de Economia.

Derrida, J. (1978). 'Structure, Sign, and Play in the Discourse of the Human Sciences'. In *Writing and Difference*. Trans. Alan Bass. London: Routledge & Kegan Paul Ltd..

Fores, B. (2012). *Brogrammers: When computer nerds become frat boys*. *Daily Caller*.

Frank, B. (1987). Hegemonic heterosexual masculinity. *Studies in Political Economy*, *24*, 159–170.

Ghosh, R. A., Glott, R., Krieger, B., & Robles, G. (2002). Free/libre open source software: Survey and study. Part 4: Survey of developers. Retrieved from http://www.flossproject.org/report/FLOSS_Final4.pdf

Hall, A., Hockey, J., & Robinson, V. (2007). Occupational culture and the embodiment of masculinity: Hairdressing, estate agency and firefighting. *Gender, Work and Organization*, *14*, 534–551.

Herring, S. (1996). Posting in a different voice: Gender and ethics in computer-mediated communication. In C. Ess (Ed.), *Philosophical perspectives on computer-mediated communication* (pp. 115–146). New York: State University of New York Press.

Hicks, M. (2013). De-brogramming the history of computing [Think Piece]. *Annals of the History of Computing, IEEE*, *35*(1), 87–88.

Kelan, E. K. (2008). Emotions in a rational profession: The gendering of skills in ICT work. *Gender, Work and Organization*, *15*, 49–71.

Kember, S. (2003). *Cyberfeminism and artificial life*. London, UK: Routledge.

Kumar, D. (2014). Disrupting the cultural capital of brogrammers. *ACM Inroads*, *5*(3), 28–29.

Levi-Strauss, C. (1962). La Pensée savage. (The Savage Mind). Translated from the French by George Weidenfield and Nicholson Ltd. University of Chicago Press.

Lin, Y.-W. (2005). Gender dimensions of FLOSS development. *Mute Magazine*, *2*(1). Retrieved from http://www.metamute.org/editorial/articles/gender-dimensions-floss-development

Masmoudi, H., Besten, M. den, de Loupy, C. de, & Dalle, J.-M. (2009). Peeling the Onion: The Words and Actions that Distinguish Core from Periphery in Firefox Bug Reports, and How They Interact Together. In K. Crowston, C. Boldyreff, B. Lundell, & A. I. Wasserman (Eds.), *Open Source Ecosystems: Diverse Communities Interacting*. (Vol. 299, pp. 284–297). Skövde, Sweden: IFIP. https://doi.org/10.1007/978-3-642-02032-2

McDowell, L. (1997). *Capital culture: Gender at work in the city*. Oxford, UK: Blackwell.

McDowell, L. (2009). *Working bodies: Interactive service employment and workplace identities*. Oxford, UK: Wiley-Blackwell.

Mitchell, L. G. (2016). *Exploring the impact of gender on 'online' and 'in person' lived experiences of women in geek communities*. University of Glasgow: Master of Research thesis.

Morrow, O., Hawkins, R., & Kern, L. (2015). Feminist research in online spaces. *Gender, Place & Culture: A Journal of Feminist Geography*, *22*, 526–543.

Nafus, D., Leach, J., & Krieger, B. (2006a). FLOSSPOLS report: Gender: Integrated report of findings. Retrieved from http://flosspols.org/deliverables/D16HTML/FLOSSPOLS-D16-Gender_Integrated_Report_of_Findings.htm

Nafus, D., Leach, J., & Krieger, B. (2006b). FLOSSPOLS report: Gender: Policy recommendations. Retrieved from http://flosspols.org/deliverables/D17HTML/FLOSSPOLS-D17-Gender_Policy_Recommendations.htm

Perkmann, M., & Spicer, A. (2014). How emerging organizations take form: The role of imprinting and values in organizational bricolage. *Organization Science*, *25*, 1785–1806.

Raymond, E. S. (1998). The cathedral and the bazaar. *First Monday*, *3*(3).

Reagle, J. (2013). 'Free as in sexist?' Free culture and the gender gap. *First Monday*, *18*.

Rommes, E., Bath, C., & Maaß, S. (2012). Methods for intervention. Gender analysis and feminist design of ICT. *Science, Technology & Human Values*, *37*, 653–662.

Rommes, E., van Oost, E., & Oudshoorn, N. (2001). Gender in the design of the digital city of Amsterdam. In E. Green, & A. Adam (Eds.), *Virtual gender. Technology, consumption and identity* (pp. 241–261). London, UK: Routledge.

Sandusky, R. J., Gasser, L., & Ripoche, G. (2004). 'Bug report networks: varieties, strategies, and impacts in a F/OSS development community'. In Proc. Of 1st Int'l workshop on Mining Software Repositories (pp. 80–84).

Shifman, L., & Lemish, D. (2010a). Between feminism and fun(ny)mism: Analysing gender in popular internet humour. *Information, Communication & Society*, *13*, 870–891.

Shifman, L., & Lemish, D. (2010b). 'Mars and Venus' in virtual space: Post-feminist humor and the internet. *Critical Studies in Media Communication*, *28*, 253–273.

Sontag, S. (1973). The Third World of women. *Partisan Review*, *40*, 186–199.

Stahl, B. C., Eden, G., Jirotka, M., & Coeckelbergh, M. (2014). From computer ethics to responsible research and innovation in ICT: The transition of reference discourses informing ethics-related research in information systems. *Information & Management*, *51*, 810–818.

Tablot, M. (2010). *Language and gender* (2nd ed.). Cambridge, UK: Polity Press.

Valentine, G., Jackson, L., & Mayblin, L. (2014). Ways of seeing: Sexism the forgotten prejudice? *Gender, Place & Culture: A Journal of Feminist Geography*, *21*, 401–414. https://doi.org/10.1080/0966369X.2014.913007

Villa, L. (2003). Large free software projects and Bugzilla: Lessons from GNOME project QA. In Proceedings of the Linux Symposium, *Ottawa*. Canada.

Warren, A. (2016). Crafting masculinities: Gender, culture and emotion at work in the surfboard industry. *Gender, Place & Culture: A Journal of Feminist Geography*, *23*, 36–54.

Wikhamn, B. R., & Knights, D. (2013). Open innovation, gender and the infiltration of masculine discourses. *International Journal of Gender and Entrepreneurship*, *5*, 275–297.

**Yu-Wei Lin** is a sociologist whose work largely centres on socio-technical dynamics in digital innovation systems. Her PhD research explored computer hacker culture and its implications in the development of information and communication technologies. Her current research is extended from this body of work and touches more broadly on open data, crowd sourcing, citizen science.

**Matthijs den Besten** is assistant professor at Montpellier Business School. His research interests include entrepreneurship and the study of online collaboration.

**How to cite this article:** Lin Y-W, den Besten M. Gendered work culture in free/libre open source software development. *Gender Work Organ*. 2018;1–15. https://doi.org/10.1111/gwao.12255