Thanks again to the reviewers for their comments. We have addressed the comments as explained below, with the original reviewers’ comments **in bold** and our response *in italics.*

**Reviewer #1: The revised paper paper has improved and the points I raised were  
largely addressed or at least discussed in the paper. The code  
duplication study, while it still leaves me with a feeling of unease,  
now clarifies that the worst case is examined. All graphs now include  
labeled axis. The analysis in the discussion & future work section  
is definitely an improvement.  
  
One thing I do still wonder about the absolute numbers; the authors  
may deem it irrelevant, but does the absolute number of casts go down  
as generics are added? Or only the cast density?**

*The raw number of casts often does go down when generics are introduced, but not always. For example, if generics are introduced as new code is added, the number of casts do not go down. As we argue in Section 6.1, we believe that measuring density is a better way to answer the research question. If the reader is truly interested, she can inspect our data set, which we will link to once the camera ready version of the article is ready.*

**And I'm happy to see  
an explanation and some discussion of Halstead in the revision, but is  
Halstead really better than normalization by SLOC? Formatting can't be  
the sole argument?**

*It is the sole argument – when formatting is different, Halstead will be more fair measure than SLOC. If the reader does not believe that formatting is a problem, that’s fine, but Halstead is not worse measure than SLOC. At a practical level, it was convenient for us to measure Halstead instead of SLOC, since we had to parse the code along the way anyway for our generics analysis.*

**Reviewer #2: Overall, I am quite satisfied with this revision of the paper, which in my mind could be accepted as is. I don't see any further improvement that are needed.  
  
If I nitpick, I could say that since the generics in figure 5 have a density multiplied by 10, there could be a second axis on the right showing that.**

*That figure shows the normalized casts vs normalized parameterized types and the text does indicate that the parameterized types are multiplied by ten to make visualization easier.  The way the three graphs are, we didn’t see an way to add a second axis without making things more confusing. Instead, for this revision, we updated the caption to indicate the scaling.*

**Otherwise, I honestly don't have anything else to add, and I am fully satisfied with the paper.  
  
I did find a few typos along the way:  
  
- p 2, l 36: "In this paper, WE add the following new contributions"  
- p 8, l 32 - 41: reread the paragraph carefully, there are several typos there  
- p 13, l 22: "we breakdown at the use" (remove at)  
- p 23, l 43: "of a total OF 4360"  
- p 24, l 42: "seperately" -> "separately"  
- p 29, l 43: "an new" -> "a new"  
- p 30, l 7: "during during"  
- p 32, l 4: "to not have" -> "do not have"***Fixed.* **Reviewer #3: The paper has been further improved. There are still some minor problems  
(some that seem to have been in the previous version!)  
  
\* Page 8. Line overhangs on lines 5 & 10 (I assume these will be dealt  
with during the publishing phase anyway)  
  
\* Page 8. Line 33. "safer" should be "safety"?  
  
\* Page 13. Line 22. "breakdown" should be "break down"?  
  
\* Page 20. Paragraph beginning line 21. The second sentence reads a little  
oddly,  especially now that you are using "density".**

*Fixed.* **\* ibid. While reading the discussion at the end of this paragraph it occurs  
to me that one place where type casts are still required is implementation  
of equals(). Maybe the introduction of generic containers somehow  
required, or at least encouraged, a higher need for equals() in those  
systems?**

*We have now added a mention of the equals() issue in the paper.*

**\* Page 30. Line 28. "their" should be "there"?  
  
\* Page 31. Line 4. Something seems broken with the sentence beginning  
"Because annotation..."  
  
\* Several references have problems with capitalisation. The ones I noticed  
are 7, 9, 12, 26.**

*Fixed.*