	Comment	Answer
R1 (Lutz)	leave out the "best" and "worst" lines from the plot	*ok, reproduced plot*
	explain the term "latent"	*ok, added "shows the _latent_ interactions (i.e. there but not visible unless you specifically extract them) among software development teams"*
	include a short legend directly in the graph	*Unfortunately, I cannot reproduce the graph. It was provided by the company anonymizing team names etc. for a prior publication ([1]). It "was" a screenshot from a prototype tool that only worked in the intranet with online access to company repositories. The company has gone through re-organization and the contact person is not accessable anymore.*
	plot teams as squares, not circles	*Same as above*
	simplify wording a bit in "How to act" section	*ok, made another pass. Hope it reads simpler.*
	The summary at the end does not fit to the story told before: "Even though the accuracy of our predictor was perceived to be good, coming up with a useful presentation of the predictions for practitioners was the real challenge." This is not the point of your actual story. In the story, you analyzed a different (and much simpler-to-analyze!) aspect of the data in order to produce something useful. The success did not come from a better presentation, it came from asking a different (and much-simpler-to-answer) question.	*ok, now it reads: "The goal of software analytics is to propose actionable changes to the way the projects are run [2]. Practitioners need to be convinced about the benefits before even considering taking actions. Our experience revealed that asking the right questions that are deemed relevant by practitioners is a key factor to provide useful insights for them. As in our case, this might require simplifying complex questions and analyses to simpler ones that are more relevant to practitioners in terms of providing actionable insight.* *_Answers to relatively simpler questions sometimes lead to better insights for practitioners._ "*
R2 (Tao)	The authors state "Though the project was a huge success in terms of scholarly outcomes and performance measures, we had run into difficulties in communicating the results to the practitioners." The authors are suggested to remove "huge" because different readers may have different expectations on how "huge" is "huge".	*ok, removed "huge".*
	with over 100.000 lines of code. Please replace "." with ",".	*ok, replaced with ","*
	They indicated that such performance figures were not useful to have an impact in their daily work, and pointing out error-prone sections within files was regarded as stating the obvious. The authors should give more elaboration on why "error-prone sections within files was regarded as stating the obvious". For example, what are these "error-prone sections" and how the practitioners already knew that these sections are error-prone without looking at the results of the analytics tool.	*ok, now it reads: "Specifically, they mentioned that they had a pretty good idea of where the bugs will occur (confirming the accuracy of our predictions), but wanted to know how they can either avoid or fix them in a different and easier way than how they already do."*
	The wording of "error" shall be replaced. There are three related words in the testing literature: "fault", "error", and "failure", with distinct meanings. The authors shall dig out these meanings and pick the right wording.	*While I'm very well aware with the terminology, and this was clarified with the company from day#1, that is the notation they use internally in practice. Since our target auidence are practitioners, I'd prefer to stick with original company terminology, also to be consistent with our prioir publication in ESEM [1].*
	The existing title and advocated comparison of prediction vs. exploration (e.g., "insights") are not rigorous. Basically the authors tried their prediction approach but it didn't work in practice, and then the authors switched to focus on visualization of software project information, and it worked. Such observation doesn't lead to the conclusion "Insights Trump Predictions". This authors didn't provide insights on why their prediction work didn't work (simply saying that the results are expected is not insightful). The CRANE system (with some prediction features) was a successful tech adoption case. So it is not the case that there exists no single system/case where prediction works.	*ok, changed title to:"Simpler Questions Can Lead To Better Insights"*
Editor	I second Lutz's comment about the summary.	*ok, pls see R1 yellow highlight*
	I second Tao's comment that the title might not be a good fit. There are cases where prediction works; in your case the engineers were just looking for different insight than what prediction could provide. As Lutz said, your success came by asking a simpler question, that was more relevant to the team.	*ok, pls see R2 yellow highlight*