	Questions on the underlying project				
	Do you remember any of the contributors of this project? Please name those that you remember below	What were the two largest packages of this monorepo (not the npm packages, but the onesthe monorepo is made out of)?	Which of the projects "sub"-packages were used most throughout the other subpackages?	Any npm packages you've revealed in this project, that you can think of? Did you take alook at their respective websites?	Have you taken a look at any of the files that were potentially "legacy" files? If yes, didyou agree on them being "refactor-worthy" or do you think they were fine as-is?
P1	Nope, not a single one. I stumbled uppon one with only 3 or 4 letters and I can't remember it, either :D Well, I'm bad with names in general but I also didn't really care when I did this because I wasn't aware this was a thing? Since I have no emotional attachment to this project and didn't really know what I was going for other than just checking out whats actually going on here, I didn't waste any capacity in trying to remember names. Because contributors don't really help me to get started with a project I guess.	I think one with 8kb and 500ish lines and one with 4kb and 250ish lines. There was one described as pretty large that only had some 80 lines and I thought to myself 'well, that's not that big in my opinion' :D	No clue. If this doesn't mean npm packages I didn't really get the impression of the existance of 'sub' packages, whatever they are supposed to be anyway	There was this merkle one. Well, that's the one I remember because of the politician :D Did have a quick look, yes, but didn't really check it out. I only check out npm packages when I actually have to work with them, not randomly only because they are used in a project. Mayyyybe that's an indicator I should change my approach towards projects haha	Not exactly sure what you mean with 'legacy' file so I don't know how to answer this.
P2					
P3	Ich erinnere mich nur an axic. Ich bin im Namen merken aber generell sehr schlecht. Ich habe mir aber ein paar Profile angeguckt.	packages/client\npackages/block	packages/common\n(Bin mir nicht sicher, klingt für mich aber am logischsten)	fs-extra: Kannte ich schon vorher. Habe aber nochmal auf der npm-Seite nachgeschaut, ob es auch wirklich das ist, was ich kenne.\ncreate-hash: Habe auf der npm-Seite nachgeschaut, welche Hash-Algorithmen unterstützt werden, bin aus der Seite aber auf die Schnelle nicht sehr schlau geworden.	Nein, ich habe mir keine Datei angeguckt. Anhang der Größe und Zeilenmenge hätte ich dem \"refactor-worthy\" aber in den meisten Fällen vermutlich zugestimmt.
P4	ein Kollege mit nem Bart, daran erinnere ich mich :)	packages/ts\npackages/common ??\nblockchain weiß ich noch, das war aber eins der kleinsten	puh, gute Frage		Feedback: Ich glaube, dass wenn man tiefer in dem Thema drin ist, das wirklich mega sinnvoll sein kann! Gefällt mir gut!
P5					
P6	vpulium I think was one of the first ones, the other ones I do not really remember	Should be block and client	Think client and common, but many of the connections were overlapping, hard to see	took a look at chalk, but I am not really into the npm ecosystem so could not take much from it	yes, seemed fair for the file I looked at - although line/file size alone does not tell the whole story
P7					
P8	s1na\ndanjm\n	ethblock\n	maybe util?	fs (does that even count as npm package?)	just came across some files, that were bigger than usual and the hint to refactor them to keep them readable. \n\nSome code examples listed below looked to be refactor-worth. For example commented out console logs or very long config values mapped to string in inline calculations.
P9	axic war glaube ich einer	block denke ich, aber nicht so genau hingeschaut, müsste ich nochmal sehen	block und client müssten das auch gewesen sein, aber waren insgesamt viele packages verbunden miteinander	fs-extra, damit hab ich schon gearbeitet, hab aber nicht auf den link gklickt	hab mir die files nicht genau angeschaut, aber grundsätzlich find ichs vertretbar, dass man bei über 1000 zeilen etwas refactoren sollte

Questions on the

Would you say, that you have discovered something interesting about the underlyingproject from going through this interactive visualization? If yes, what? If no, what wasmissing or what would you have liked to see?	What was your overall experience going through this visualization? What did you like ordid not like?	How did you experience the companion (lower right)? Was it helpful, annoying or did younot really interact with it at all?	Could you imagine yourself using some kind of interactive visualization or somethingsimilar on different projects to learn about them? If so, on which projects would you want totry it out? If not, what would you prefer instead to make yourself familiar with new projects?	How do you usually approach the onboarding onto new projects? To whom do you talk to,what applications do you use, how do you navigate through code?
Hmmm. When I think about it, it was actually interesting to get told about contributors and see their faces. Not that it would really help me getting into this project but yeah, gives it a more personal flavor. The information provided during each discovery was still a bit big to digest in a playfull way I think. More visualizations like images/icons rather than just plain text and a sample of the code and a link to the actual file. With icons I mean representation for e. g. (too) big files or an icon I learned earlier this represents a contributor. This is easier to process than reading 2 lines of text that tells me this is a contributor or whatever. The texts where pretty repetitive and I stopped reading after 'contributor' so I might've missed any	Was ok. I was a bit confused because I didn't know what to do when I reached the first blank slide. The helper on the bottom was quick to assist, though. When elements disappear, I would make them fade out rather than just getting invisible instantly. Felt a bit glitchy. You could make the impression of revealing areas better when the initial background would be dark and when you trigger the sonar, everything inside the expanding circle is bright. Maybe with some something visual in the background that gives some depth perception.	Didn't really interact with it but I think it came to help always at the right time. Good guy!	To be honest, I don't really know. It strongly depends on the urgency to get familiar with a project I guess. I mean, 'interactive visualization' can mean a lot. Exploring something like in this example can be frustrating if you really just want to get the gist of something because you stumble accross so many random things. Actually I think this kind of visualization isn't the right thing to learn about a coding project. At least not for me.	Well, hadn't had too much of these occations yet but so far I set it up first to get a feeling for what I'm dealing with (if possible) and than I dig deeper into it. I talk to those who set up or work on the project if possible. What exactly do you mean with what applications I use? If you mean for my IDE, it's mostly stuff that helps visualize things like pairs of brackets and key words or variables vs functions and so on.
Ich habe einen ungefähre Vorstellung davon bekommen, wie die Projekte untereinander zusammenhängen. Auch wenn ich es mir nicht merken konnte, war eine Visualisierung auf alle Fälle hilfreich.\nEs ist schön durch die Vorstellung der Contributors ein paar Namen kennenzulernen, denen man im weiteren Verlauf des Projekt vermutlich häufiger über den Weg laufen wird.\nMit der Vorstellung der großen \"refactor-worthy\" Dateien werden einem direkt mehrere Möglichkeiten eröffnet, in das Projekt einzusteigen.\n\nMir fehlte ein wenig das fachliche über das Projekt. Was tut es? Wofür ist es gut? Nur anhand des Codes ist das schwierig zu verstehen	Im ersten Moment wusste ich nicht, wie ich starten sollte. Durch die Hilfen habe ich aber sehr schnell verstanden, wie der Ablauf gedacht ist. Das Suchen der verschiedenen Punkte hat im ersten Moment Spaß gemacht, wurde anschließend aber ein bisschen nervig (vor allem weil einige Punkte vor einem abgehauen sind).	Er war auf alle Fälle hilfreich um zu verstehen, wie das Onboarding zu bedienen ist. Leider sagt er nichts, wenn man auf ihn klickt. Eine Wiederholung der Hilfe oder irgendein Feedback wären nett gewesen.	Um sich einen ganz groben ersten Überblick über das Projekt zu verschaffen, würde ich so eine Visualisierung durchaus gut finden.	Wenn ich an einem neuen Projekt anfange, arbeite ich mich von außen immer weiter rein. So verstehe ich auch den Ablauf des Codes anhand von konkreten Interaktionen mit dem Programm. Austausch mit erfahrenen Entwicklern finde ich dabei sehr wichtig. Auch das durchgucken alter Issues und deren Lösungen finde ich immer sehr hilfreich.
Liked the connections and the overview map, helped me see at a glance what is in the project\nmissed actual textual content of the project besides the previews, maybe include more of the contents of the underyling project	Liked the little worm character, felt smooth\noverview as said before was good, would have liked more connection to the source	Helpful as far as indicating what I have done goes\nbesides that not much interaction with it, could not click it besides messages	The overview could help yes, would like to see it on projects of another programming language\nespecially those that maybe are not as cleanly divided into sub projects\nwould have also liked to see more of the structure of the single sub projects\noverall would probably still go at new projects in the IDE itself, seems more efficient at least for the source code	First, I try to get access to everything that I need (git, deployment server, ...) and then set up the project locally and try to build it, if there is at any point problems I try to contact the person who wrote the code otherwise the other team members already in the project\nBest case there is documentation for the initial setup, but that is often missing\nI also try to talk to project management to get to know the organizational structure\nNavigation in code I do with IntelliJ and go-to-definition most of the time
I was surprised that even a major project like ethereumjs has the same problems like many other projects like commented out code or console logs or very big util.ts files, ...\nSome basic information like GitHub stars would be cool at the beginning	within the first module, I needed some time and the help box to understand, that I was supposed to click somewhere to reveal the information.\n\nAt the beginning I had some problems catching the fast moving dots, but with some practise they where kind of sliding into my mouse position and stop when hovered. That was fun at first, but at the end it was not that much fun anymore to catch the dots, because it was that easy.\n\nThe first visualisation of all modules looked a bit chaotic. At first I thought, \"that should help me get a better overview?!\".	It came up when I was the first time in a sub module and didn't know what to do. There is was very helpful. \nI didn't interacted with it active.\n	I don't think that I would prefer such a game/interactive visualisation in contrast to just going throw the filestructure, because the game seemed to be a bit random. I don't know if the modules I checked out are the most important ones or not.	usually it's a coworker introducing someone new to the project. When I'm checking out a new github project, I just roam around and search for some modules and files whose name seem important to me.\n\nAt first I just go through the file structure in GitHub and when I wanna dig deeper I open the project in Intellj.\n
fand die übersicht gut, dass man alle packages auf einem platz sieht\nhätte mir mehr info zu den packages gewünscht, zusätzlich zu namen und größe, vielleicht direkt in der übersicht	fand die infos je package ganz gut, dass man die direkt die personen sieht mit dem was sie als letztes gemacht haben\ndas aufdecken war aber bisschen eintönig, da hätte mir was zusätzliches zu machen gefallen	sah ganz nett aus, fand die kleine animation mit dem auge ganz gut, aber viel interaktion gab es nicht	ja fände ich ganz gut bei neuen projekten grade bei größeren, wo es sonst schwierig is einen überblick zu bekommen\nlinux-kernel würde mir einfallen, also bei großen ganz bekannten projekte, bei denen ich aber nicht unbedingt viel mit der programmiersprache gemacht habe	klone mir das projekt als erstes und öffne es dann in meiner ide, da gehe ich dann vom main file ausgehend in die unterfiles, je nachdem was ich machen muss\nwenn mehrere leute am projekt arbeiten, versuche ich in einer session zusammen die arbeit am projekt anzufangen um direkt einen ansprechpartner für fragen zu haben

research project				
Do you work on Open Source Projects? If you are a contributor of one, how do you try tomake it easy for new collaborators to work on these projects? If not, how do you approach collaborating for yourself?	Do you have any additional ideas on how playful elements or game mechanics could beused within the onboarding phase of software development projects?	What is your general stance on using games/game mechanics or playful elements withinsoftware development? Do you see a difference of using such mechanics in open sourcesoftware vs. in a work environment?	Do you have any additional ideas on how playful elements or game mechanics could be usedwithin the onboarding phase of software development projects? Any elements from games youplay that you think could be reused when making yourself familiar with new projects?	Anything else you want to mention?
I didn't contribute to an open source project so far but I try to stick to conventions and write code that's hopefully easy to understand with meaningful variable and function names, even though they might get a bit bigger.	Rather than just having the 'main quest' of getting to know the project, I'd provide some extra goals, like the first 5 questions about contributers I remember or not and so on so the explorer has something to aim for. Maybe these 'quests' can be aimed to point out core concepts of the project and guide the participant towards them.	I'm not sure. Gamification is great to learn things like coding itself, like exploring a story with command line operations. Depends on the project I guess. And it's about time. Exploring something in a playful way also needs to be done it each person's personal pace. I mean, I always think in the perspective of the gaming approach in this project. But I just see my thoughts to this will be better kept in the next input box :)	As I mentioned earlier, exploring something also needs time and is not fun if you are under preasure and you just want to get the gist of something. As you did it in your example, I want to have the headspace to branch out and check out parts of the project that are maybe just nice to know but necessarely helping me with actually working on the project. A better approach I could think of (in my opinion) is to make the participants make solving problems, like searching for an answer for a specific question. This means stronger guidence through parts of the project, smaller bit sizes. Maybe let the participant choose a goal and give him or her some sort of quest line with a bunch of sub goals to achieve. For example, letting them to	Nope! Good look with your study my guy! :)
Entweder mir fällt selber ein Fehler/Verbesserung auf und ich möchte es umsetzen oder ich sehe Issues anderer Benutzer und möchte diese Lösen. Anschließend versuche ich mich von oben nach unten durch den Code zu \"wühlen\" um irgendwann an der Stelle anzukommen, an der ich etwas ändern/verbessern möchte. Von da aus verzweige ich oft in die verschiedenen Bereiche des Codes, die an dieser Stelle verwendet werden. So lerne ich nach und nach die Funktionen kennen und wie sie verwendet werden.	Ich fände ein Beispiel-Issue spannend. Ein Fehlverhalten, das einem präsentiert wird und das man zu lösen hat. Das Beispiel-Issue sollte so gewählt sein, dass man an vielen wichtigen Stellen des Codes vorbeikommt. Am Ende sollte der Fehler aber recht einfach zu lösen sein ohne viele Dinge refactorn zu müssen.	Ich bevorzuge doch mehr den direkten Kontakt zu Kollegen/Contributors und das selbständige Entdecken des Codes. Für den allerersten Überblick und eventuell einer fachlichen Vorstellung der Software finde ich spielerische Elemente durchaus angebracht. Den Bedarf würde ich hier mehr bei den Open Source Projekten sehen, weil hier oft eine direkte und nahe Kommunikation mit anderen Entwicklern nur schwer möglich ist. Im Arbeitsumfeld sind Kollegen eher greifbar und bei Fragen ansprechbar.	Eine spielerische Vorstellung des Projekts aus fachlicher Sicht mit Beispielen fände ich hilfreich. Vielleicht mit einem Art Wettbewerb wer einen gewissen Workflow schneller durchgespielt hat.	
No, I am not working on open source projects, I just use them in my code, for that the first step is always the documentation, if there are problems I google or search through the issue list of the package	For the source code, would have liked an approach that is near to the code, maybe within the IDE itself and more textual or a little helper within the IDE acting playful\nFor the project organization structure maybe a clearer picture of who to talk to and communication with them as part of a game	I do not think it can be efficient in day-to-day work, but maybe for looking at open source projects out of interests\nAlso maybe for young developers or people starting with development could be helpful	Not really, maybe include more of a progress status or go all the way and create a 3D world from a project that you can go through, although that could be way too far from the project	
nope, not actively.	There could be some moving dots from module to module, depending on how often the module gets imported into another module. With that you would easy recognise which module is used a lot and which not that often.\n\nI think there is some activity missing, many games have this discover, fight enemies concept. This guide is good for discovering, but for fun some minigames like fighting the bugs or something similar could make it more interesting to play. \n\nThere could also be some videos of the commit history of the project like: \nhttps://www.youtube.com/watch?v=cNBtDstOTmA	In a work environment you are often applied to that project and there is someone who can introduce you. In open source project often someone new just joins randomly selfmotivated. Therefore I guess this playful automatic introduction brings more benefits for open source projects, than for work projects.	* \"The typical find x items of y\" feature (like the pigeons in gta4)\n* Some speaking character introducing with some basic information. \n* Some character to follow, like a rabbit or something that you need to follow from module to module. \n* Some area constraints, so that you discover a certain area, and after some tasks and knowladge gains you are allowed to discover a broader area.\n	nice design of the project.
ne, arbeite aktuell an keinen oss projekten	vielleicht könnte man die leute die am projekt arbeiten als npcs darstellen, die mit einem interagieren und so bisschen mehr fokus auf kommunikation legen\noder man macht ein multiplayer game aus der einarbeitung, das dann mehrere gleichzeitig spielen	fänd ich gut, grade als einarbeitung in projekte, wo man schwierig einen überblick bekommen kann\nist vielleicht nicht das schnellste, aber um sich spielerisch eine idee zu verschaffen eine gute idee vielleicht	wie beschrieben, multiplayer wäre eine idee	