

Questions on the research project									
<p>What you say, that you have discovered something interesting about the underlying project, how going through this interactive visualization? If yes, what? If no, what was missing to what would you have liked to see?</p> <p>Heroin: When I look about it, it was actually interesting to get lost 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 playful way I think. More visualizations like image/icon 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 lists where pretty cluttered.</p>	<p>What was your overall experience going through this visualization? What did you like add/d not like?</p> <p>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 scaner, everything inside the expanding circle is bright. Maybe with some something visual in the background that gives some depth perception.</p>	<p>How did you experience the companion (lower right)? Was it helpful, annoying or did you not really interact with it at all?</p> <p>Didn't really interact with it but I think it came to help always at the right time. Good guy!</p>	<p>Could you imagine yourself using some kind of interactive visualization or something similar on different projects to learn about them? If so, on which projects would you want try it out? If not, what would you prefer instead to make yourself familiar with new projects?</p> <p>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 this in this example can be frustrating if you really just want to get the gist of something because you stumble across 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.</p>	<p>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?</p> <p>Well, hadn't had too much of these occasions yet but so far I set it up first to get a feeling for what I'm dealing with (if possible) and then 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.</p>	<p>Do you work on Open Source Projects? If you are a contributor of one, how do you try to make it easy for new contributors to work on these projects? If not, how do you approach collaborating for yourself?</p> <p>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.</p>	<p>Do you have any additional ideas on how playful elements or game mechanics could be used within the onboarding phase of software development projects?</p> <p>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 contributors 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.</p>	<p>What is your general stance on using games/game mechanics or playful elements within software development? Do you like a difference of using such mechanics in open source software vs. in a work environment?</p> <p>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 fun. Exploring something in a playful way also needs to be done if 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 :)</p>	<p>Do you have any additional ideas on how playful elements or game mechanics could be used within the onboarding phase of software development projects? Any elements from games yourself that you think could be reused when making yourself familiar with new projects?</p> <p>As I mentioned earlier, exploring something also needs time and is not fun if you are under pressure 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 not necessarily 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 guidance 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</p>	<p>Anything else you want to mention?</p> <p>Nepel! Good luck with your study my guy! :)</p>
<p>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. Wie ich schon durch die Vorstellung der Contributors ein paar Namen kennenzulernen, deren man im weiteren Verlauf des Projekt vermutlich häufiger über den Weg laufen wird. InMitt der Visualisierung der großen 'Tractor-worship' Dateien werden einem direkt mehrere Möglichkeiten eröffnet, in das Projekt einzusteigen. InMitt der Karte finde ich wenig das technische über das Projekt. Was ist es? Wofür ist es out? Nur anhand des Codes ist das scheitern zu verstehen.</p>	<p>In ersten Moment wusste ich nicht, wie ich starten sollte. Durch die Hilfen habe ich aber sehr schnell verstanden, wie der Ablauf gestaltet 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).</p>	<p>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.</p>	<p>Um sich einen ganz groben ersten Überblick über das Projekt zu verschaffen, würde ich so eine Visualisierung durchaus gut finden.</p>	<p>Wenn ich an einem neuen Projekt anfang, arbeite ich mich von außen immer weiter rein. So verstehe ich auch den Ablauf des Codes anhand von konkreten Interaktionen mit dem Programm. Auslausch mit erfahrenen Entwicklern finde ich dabei sehr wichtig. Und das durchsuchen aller Issues und deren Lösungen finde ich immer sehr hilfreich.</p>	<p>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.</p>	<p>Ich finde 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 nachdenken zu müssen.</p>	<p>Ich bevorzuge doch mehr den direkten Kontakt zu Kollegen/Contributors und das selbständige Entdecken des Codes. Für den allerersten Überblick ist eventuell einer fachliche 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.</p>	<p>Eine spielerische Vorstellung des Projekts aus fachlicher Sicht mit Beispielen finde ich hilfreich. Vielleicht mit einem Art Wettbewerb wenn ein gewissen Workflow schneller durchgeschaltet hat.</p>	
<p>Liked the corrections and the overview map, helped me see at a glance what is in the project/confirmed actual textual content of the project besides the previews, maybe include more of the contents of the underlying project</p>	<p>Liked the little worm character, felt smooth/interview as said before was good, would have liked more connection to the source</p>	<p>Helpful as far as indicating what I have done goes/includes that not much interaction with it, could not click it besides messages</p>	<p>The overview could help yes, would like to see it on projects of another programming language/specifically those that maybe are not heavily divided into sub projects/would have also liked to see more of the structure of the single sub project/overall would probably still go to see new projects in the IDE itself, seems more efficient at least for the source code</p>	<p>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/Best case there is documentation for the initial setup, but that is often missing/also try to talk to project management to get to know the organizational structure/navigation in code I do with IntelliJ and go-to-definition most of the time</p>	<p>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</p>	<p>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/for the project organizational structure maybe a cleaner picture of who to talk to and communication with them as part of a game</p>	<p>I do not think it can be efficient in day-to-day work, but maybe for looking at open source projects out of interest/Also maybe for young developers or people starting with development could be helpful</p>	<p>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</p>	
<p>I was surprised that even a major project like elasticsearch has the same problems like many other projects like commented out code or console logs or very big util or files. In some basic information like GitHub stars would be cool at the beginning</p>	<p>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. InMitt the beginning I had some problems catching the first moving dots, but with some practice 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. InMitt the first visualization of all modules looked a bit chaotic. At first I thought, 'That should help me get a better overview?'</p>	<p>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. InI didn't interacted with it active. In</p>	<p>I don't think that I would prefer such a game/interactive visualization in contrast to just going throw the file structure, 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.</p>	<p>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. InMitt I first I just go through the file structure in GitHub and when I wanna dig deeper I open the project in IntelliJ</p>	<p>nope, not actively.</p>	<p>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. InI think there is some actively teasing, 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. InI think there could also be some videos of the current history of the project like: https://www.youtube.com/watch?v=dNB0x0D7mYA</p>	<p>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.</p>	<p>"I think the typical find x items of y" feature (like the pigeons in gltf/er) Some character to follow, like a rabbit or something that you need to follow from module to module. In" Some area constraints, so that you discover a certain area, and after some tasks and knowledge gains you are allowed to discover a broader area in</p>	<p>rice design of the project.</p>
<p>find die Übersicht gut, dass man alle packages auf einen platz sieht/verhält sie mehr info zu den packages gewünscht, zusätzlich zu namen und größe, vielleicht direkt in der übersicht</p>	<p>find die info ja package ganz gut, dass man die direkt die personen sieht mit dem was sie als infos gemacht haben/index aufdecken vor aber bisschen eintönig, da hätte mir was zusätzliches zu machen gefallen</p>	<p>sah ganz nett aus, fand die kleine animation mit dem auger ganz gut, aber viel interaktion gab es nicht</p>	<p>ja finde ich ganz gut bei neuen projekten grade bei größeren, wo es sonst schwierig ist einen überblick zu bekommen/minus-kernel würde mir einfallen, also bei großen ganz bekannten projekten, bei denen ich aber nicht unbedingt viel mit der programmiersprache gemacht habe</p>	<p>könnte mir das projekt als erstes und offene was dann in meiner idee, da gehe ich dann vom main file ausgehend in die unterteile, je nachdem was ich machen muss/lernen mehrere kette am projekt arbeiten, versuche ich in einer session zusammen die arbeit am projekt anzufangen um direkt einen ansprechpartner für fragen zu haben</p>	<p>ne, arbeite aktuell an keinen oss projekten</p>	<p>vielleicht könnte man die leute die am projekt arbeiten als npc darstellen, die mit einem interagieren und so bisschen mehr fokus auf kommunikation legen/oder man macht ein multiplayer game aus der einarbeitung, das dann mehrere gleichzeitig spielen</p>	<p>find ich gut, grade als einarbeitung in projekten, wo man schwierig einen überblick bekommen kann/ist vielleicht nicht das schnellste, aber um sich spielerisch eine idee zu verschaffen eine gute idee vielleicht</p>	<p>Wie beschrieben, multiplayer wäre eine idee</p>	