

Interview-Transkript Jann-Intveen (WMA)

I: Interviewer

P: Protokollant

B: befragte Person

P [00:00:03] Es hat funktioniert.

I [00:00:03] Ja. Mega. Herr Bente hatte uns schon ein bisschen erzählt - von DiveKit geht - allerdings so eine recht schnelle Einführung. Da du wahrscheinlich der einzige Stakeholder bist, der technisch was zu den Hintergründen sagen kann - erfahren wir ja dann noch - haben wir bei dir auch einen großen Teil an technischen Fragen genommen – wo du auch gerne die Tiefe gehen darfst.

B [00:00:25] Ja.

I [00:00:27] Ja, aber fangen wir mal ganz allgemein an: Wie bist du denn zu diesem Projekt gekommen?

B [00:00:33] Also wie gesagt, ich arbeite ja bei Herrn Bente als WMA. Und dann hatten wir irgendwann die Anforderung - und das war auch schon vor Corona - dass wir die Leute halt mehr zum Coden bringen wollen. Also das war das erste. Und dann ging es erstmal darum, dass die Leute nicht einfach nur irgendwelche Sachen lokal coden sollen - und dann sag ich mal irgendwo Dateien hochladen – sondern es ging auch darum, dass man mal so eine Plattform hat, wo die Leute auch Git, zum Beispiel, benutzen können. Also dass man so eine Umgebung schafft, die für die Leute eben relativ realitätsnah ist und nicht so realitätsfern, also dass sie eine richtige IDE benutzen - wie gesagt Git nutzen und so. Das war unsere Anforderung. Und dann haben wir uns überlegt, wie wir das so auch ein bisschen automatisieren könnten, weil das Problem ist, halt bei GitLab zum Beispiel oder das ist ja auch bei GitHub ebenso bei so Repository Verwaltungsplattform. Da ist ja das Problem, dass es nicht so ausgelegt hat, rein für die Lehre. Also wenn halt jeder sein eigenes Repository haben will, dann müssen die irgendwie angelegt werden zum Beispiel. Also solche Anforderung hatten wir eben und haben uns dann dafür überlegt, weil wir da jetzt nichts gefunden haben, was es schon so als Lösung gibt. Wir haben uns da überlegt, dass wir uns da mal einen Tooling zusammenbauen, das halt angepasst ist an Anforderungen aus der Informatik-Lehre. Und da kam dann halt noch relativ spontan mal so irgendwie so eine fancy Idee auf. „Ja, man könnte ja sogar Individualisierung mit reinbringen.“ Und ja, dann hat sich das mit der Zeit entwickelt. So kam ich eben dazu.

I [00:02:12] Welche Funktion hast du dann eingenommen? Reine Entwickler oder alles?

B [00:02:19] Ja, alles ist ein bisschen grob, aber hauptsächlich Entwickler. Aber ich war halt auch mal beteiligt mit... Wir haben uns über Überlegungen angestellt. So was könnten wir brauchen? Wie könnten wir das auch umsetzen? Und da war ich ja mit beteiligt. Da war dann halt auch noch zwei andere beteiligt. Und dann halt hauptsächlich in der Entwicklung. Also das heißt, ich habe mir hauptsächlich überlegt, wie man das jetzt implementiert oder wie man halt die Anforderung, die wir haben, konkret halt abbildet, indem wir halt eine Software dahinterstehen haben. Ja.

I [00:02:55] Welche Rollen waren das konkret, die da jetzt beteiligt waren?

B [00:02:59] Ich kann ja auch mal Namen nennen, die müsste dann nur leider wahrscheinlich später anonymisieren, weiß ich jetzt nicht. Einmal halt mit dem Fabian - Krampe. Der hat halt vor allem sehr viel inhaltliche Kompetenz. Also der ist bei Softwaretechnik sehr inhaltlich dabei. Ist halt auch ein WMA von Herr Bente und natürlich Herr Bente. Also da kamen dann halt die hauptsächlichen Anforderung her und natürlich auch das letzte Wort. Sowas brauchen wir tatsächlich, sowas brauchen wir tatsächlich nicht - und was wäre denn cool oder sowas? Ja.

I [00:03:32] Wurdet ihr da mit einbezogen in den Prozess? Was er sagen wollte, das finde ich gut - das finde ich eher schlecht?

B [00:03:36] Das auf jeden Fall, Ja. Also das. Wir sind ja dann halt so ein Dreiergespann gewesen, was sich oft halt Dinge, sag ich mal, überlegt hat, wo noch nicht so viel Lösungen halt existieren und haben halt auch geschaut, was wir konkret brauchen. Erst mal für uns – was für uns halt relevant ist. Und auch was, bezogen halt auch auf die Lehre, die wir halt zu machen oder bzw. die Herr Bente halt hauptsächlich dann so macht und wie er die halt ausführen will. Aber klar konnten wir auch Ideen einbringen, wenn es dann natürlich sinnvoll war, dann sind die tatsächlich auch mit übernommen worden. Wenn man natürlich Dinge sagt, die jetzt vielleicht nicht so ideal sind, dann wird halt drüber diskutiert und halt schon mal auch verworfen.

I [00:04:28] Gut. Gerade eben hast du schon gesagt, wer ungefähr alles bei der Nutzung des DiveKits vorhanden ist, also Studenten, Professoren wahrscheinlich, noch mehr?

B [00:04:39] Ja halt noch die Betreuer natürlich, hier wissenschaftliche Mitarbeiter, Hilfskräfte, eben solche Leute, die halt die Professoren unterstützen. Ja, das sind eigentlich so alle Gruppen, die da beteiligt. Ich weiß nicht, ob wir jetzt auch wissen möchtest, inwiefern die beteiligt sind, oder nur, ob die Gruppen beteiligt sind.

I [00:04:58] Ja gerne. Gerne auch „inwiefern“.

B [00:05:01] Also der Professor oder halt wir als WMA. Das kann man so erstmal auf eine Ebene stellen, die erstellen halt erstmal Aufgaben. So. Diese Aufgaben werden dann auch verteilt, übers DiveKit, und das macht natürlich auch das nutzt dann eben auch die, die Professoren und die die WMAs, unterstützende Betreuer. So und von dem DiveKit an sich sehen die Studenten halt eher weniger, die sehen halt erstmal nur ihr Repo im GitLab bei uns jetzt konkret und bearbeiten dann da eben die Aufgaben drin. So und das heißt von diesem Hintergrund „technische Sachen“ kriegen, die weniger mit. Sie sehen die gesagt nur ihr Repo und so ein Repo würden sie theoretisch auch sehen, wenn ich ihnen manuell anlege. Das wär halt nur eine Heidenarbeit, aber von diesen hinteren Prozessen kriegen die halt tatsächlich nicht so viel mit.

I [00:05:54] Gibt's bei wissenschaftlichen Mitarbeitern und Professoren so eine Rollenverteilung oder ist das quasi eine Rolle? Also haben beide dieselben Rechte?

[00:06:04] Dieselben Rechte, ja, also wir sind halt. So auf dem GitLab sind wir halt eben alle admin, das heißt da kann eigentlich jeder machen, was er will - quasi. Das basiert natürlich alles auf Vertrauen. Dann kann man natürlich auch Sachen

rückgängig machen, wenn jetzt Unfug passiert ist. Aber hauptsächlich die Rechteverteilung ist schon sehr gleich. Es geht natürlich mehr dann so in der Richtung Richtung „Wer hat jetzt das Sagen“, „Sowas wird jetzt gemacht“ und so ist natürlich der Professor ganz oben. Aber jetzt die reine technische Rechteverteilung da sind eigentlich Professoren und wir sind alle Mitarbeiter so auf einer Ebene.

I [00:06:39] Alles klar.

B [00:06:40] Weil sonst wird das halt auch schwierig, Dinge zu machen, zu agieren, wenn da andauernd Rechte fehlen.

I [00:06:46] Ja, auf jeden Fall. Gut. Welche Ziele sollen denn konkret verfolgt werden, durch das DiveKit?

B [00:06:56] Also es hat sich – richtet sich vor allem nachdem, was wir für Verantwortung hatten, als wir angefangen hatten mit dem Tooling, erstens mal die Leute sollen mehr coden. Die Leute sollen mit Werkzeugen hantieren, die auch nicht realitätsfern sind, sondern die man auch wirklich nutzt. Das heißt, eine vernünftige IDE. Und nicht gerade sag ich mal mit Notepad coden. Das gibt's tatsächlich auch. Dann halt, wie gesagt mit Git, also da haben wir uns halt für GitLab entschieden. Aber gut, das ist jetzt ein anderer Schuh. Das Problem ist aber oft im Studium, vor allem wenn man sehr viele Studenten hat. Dann ist manchmal so ein bisschen jeder für sich. Also dann hat man nicht so ein Wir-Gefühl, glaub ich auch oft. Und deswegen die Leute, die kopieren halt man schon. Also die haben manchmal nicht so eine Mega Motivation, deswegen muss man die halt manchmal auch dazu zwingen oder die halt so extrinsisch motivieren, damit die halt auch mal was machen und lernen. Die brauchen manchmal noch, weil da sind ja auch oft jüngere Leute, die sind noch nicht ganz so ganz hinter dem her was, also was Motivation angeht. So und das heißt sie kopieren oft. Vor allem wenn man Gruppenarbeiten hat, dann macht oft einer was und die anderen machen vielleicht in anderen Kursen was und das ist ja eigentlich nicht die Idee dahinter, oder manchmal manche, die, wenn es jetzt keine Gruppenarbeiten gibt, da lädt jetzt einer die Lösung hoch und alle anderen kopieren sich das.

I [00:08:15] Ja

B [00:08:16] So. Das es das die andere Anforderung, dass wir eben wollen, also Gruppenarbeit sich nicht schlecht, Wir wollen nur, dass die Leute auch alle irgendwie bisschen beteiligt sind, so deswegen machen wir das häufig so, dass wir zwar sagen „Ja, ihr könnt keine Gruppenarbeit machen, aber es muss halt jeder einzeln abgeben. Und zwar nicht in Form von mündlichen Prüfungen, weil die sind halt eben manchmal schwer, also in der mündlichen Prüfung ist es schwer zu sagen „So, du bist jetzt durchgefallen“, weil oft hat er halt dann schon ein paar Dinge gesagt, da kann man jetzt nicht sagen, der hat jetzt keine Ahnung gehabt. Aber man merkt hat schon, dass der eigentlich nur irgendwas nachplappert. Vielleicht auch was seine Teamkameraden ihm vorhergesagt hat oder was andere gesagt haben. So, deswegen machen wir das ebenso: „Jeder muss einzeln abgeben und halt auch eine Reihe von Tests, dann automatische Tests, erfüllen“. Und so hat man dieses Ziel erfüllt, dass jeder was machen muss. Jeder muss die bei sich grün haben. Und damit man eben nicht kopieren kann, um die grün zu haben, haben wir halt auch Individualisierung mit drin. Also dass wir halt einen bestimmten Grad individualisieren, dass halt nicht einfach was kopiert werden kann. Und sofort ist alles grün, aber halt auch nicht so stark

individualisieren, dass man sagen kann, „So, jetzt ist keine Gruppenarbeit mehr möglich, weil der hat jetzt auf einmal der programmiert was in der Autowaschanlage und der andere programmiert jetzt einen Bäckerladen, keine Ahnung. Also das hat schon ein bisschen angeglichen, sodass man sich halt unterhalten kann, aber man sich dann damit beschäftigen muss. Das sind auf jeden Fall so die größten Ziele.

I [00:09:50] Gut. Gibt es bestimmte Dinge, wo ihr gesagt habt „Das wollen wir auf jeden Fall nicht damit erreichen“, also was vermieden werden soll?

B [00:09:59] Ja, also was man, wo man aufpassen muss, ist da ist halt viel Automatisierung dahinter, müssen wir auch, weil das könnte man jetzt eben schon mal gesagt nicht alles manuell eben. Da wäre Erstellen der ganzen Repos und Individualisierung, manuell sowieso so ziemlich nicht machbar. Es heißt ja viel Automatisierung ist dabei. Was wir aber überhaupt nicht erreichen wollen, ist, dass sich jetzt ein Lehrender denkt „Okay, ich chill jetzt mal und mein Tool macht alles und ich mache auch jetzt nichts an Betreuungen und das wird schon irgendwie so“. Also die Idee ist halt, dass das Tool einen unterstützt in der Lehre und dass man halt solche Betreuung auch an anderen Stellen effizient einsetzen kann, da wo sie wirklich nötig ist. Dass man zum Beispiel nicht Zeit damit vergeudet zu schauen, ob jemand eine Aufgabe korrekt hat. Besser wäre es doch, wenn jemand die Aufgabe nicht korrekt hat, weil das die Tests eben irgendwie äußern und man dem dann hilft, vielleicht die Aufgabe irgendwie hinzubekommen, zu bestehen oder auch einfach erklärt, was da vielleicht nicht richtig ist, also was die Tests anmeckern. Wenn jemand z. B. super im coden ist, sofort die ganzen Tests grün hat, dann braucht er wahrscheinlich da erst einmal keine Betreuung oder zumindest auf einer anderen Ebene. Also vielleicht kann man mit dem dann über Dinge reden, die noch fancy wären oder vielleicht interessiert er sich noch für irgendwas. Dann hat man dafür eben Zeit, weil man die Zeit nicht woanders vergeudet. So, und das ist wie gesagt das Ziel. Was wir überhaupt nicht erreichen wollen, ist halt, dass das Gegenteil passiert, dass man sich so ein bisschen auf dem Tooling ausruht, sagt „So, jetzt hab ich eine Aufgabe, die mach ich jetzt zehn Jahre lang und die knall ich denen immer mit dem automatisierten Tool um die Ohren und es wird sofort getestet. Ich mache gar nichts mehr.“ Also das soll mir auf keinen Fall erreichen.

I [00:11:42] Alles klar. Dann kommen wir mal auf die Frage wie du glaubst, wie das System denn auf andere wirken könnte und zwar jetzt z. B. auf die Studierenden und deren Leistung.

B [00:12:00] Also das kann man so an dem anknüpfen, was ich gerade gesagt habe. Es kann natürlich, je nachdem, wie man es nutzt oder wenn man erst nur davon erzählt, was es so macht, kann es natürlich so wirken, dass die sagen „Ja Moment mal, ihr wollt uns gar nicht mehr wirklich hier betreuen. Wir werden nur automatisch abgefertigt“ oder so und da müssen wir den Studenten halt klarmachen, dass es eben nicht so sein soll. Und da müssen die auch einfach mal so einen Durchlauf mitmachen, den man dann eben auch so führt, dass – also die Leute kriegen halt sag ich mal so ein Git Repo, dann können sie darin rumbasteln, abgeben und wenn man die dann halt an den richtigen Stellen immer betreut oder Betreuungsleistungen eben zur Verfügung stellt, dann glaub ich nehmen ich das auch positiv war. Also die dürfen sich halt nicht so verloren fühlen und irgendwie so abgekapselt. Vor allem mit Corona ist das halt glaub ich oft so, wo man nicht auch mal einen Betreuer hat, der irgendwie rumgeht durch den Raum und sagt „Kommt ihr hier weiter oder so?“ Das geht ja jetzt

schlecht. Ich kann ja nicht jeden jetzt irgendwie ne Mail schreiben „Hier hast du ein Problem bei der Aufgabe?“ Es muss dann halt mehr von den Studenten ausgehen und dann müssen die eben auch dementsprechendes Feedback kriegen von den Betreuern. Also dass man da eben, wenn man sagt ich würde gerne die Tage mal Hilfe haben, dass man da eben den Studenten auch hilft und nicht sagt „Ja, hier macht die Tests grün“. So ungefähr. Also ich glaube, wenn man das halt so macht als Betreuer, dann wirkt es auch positiv auf die Leute.

I [00:13:30] Wissen die Studenten schon ungefähr, dass sowas kommt oder gab's das schon mal irgendwie so? Gespräche oder so Feedback?

B [00:13:40] Du meinst Feedback bezüglich auch wie das auf die wirkt?

I [00:13:43] Genau. Dass vielleicht gesagt wurde „Hey, wir wollen das Ganze ein bisschen automatisieren, indem wir uns so etwas vorgestellt haben.“ „Was denkt ihr darüber?“

[00:13:54] Ach so nein. Die Idee an sich kam nicht von Studenten ab sich, wenn du darauf hinauswolltest. Wir haben halt nur, nachdem wir jetzt mal den ersten Kurs gemacht haben, nach dem Prinzip halt mal ein bisschen Feedback eingeholt, wie Leute zu finden. Und da ist natürlich auch viel Feedback mit dabei. Generell, wie wir die Lehre da gestaltet hatten, weil wegen Corona mussten wir einiges anders machen. Und das war aber sehr positiv. Zum Beispiel haben wir ja auch Discord genommen als Kommunikations-Tool. Da kann man so eine Art virtuellen Klassenraum, sag ich mal, aufbauen. Die Leute reden untereinander, haben aber halt auch direkt die Betreuer direkt an der Hand. Das ist irgendwie so ganz anders als dieses distanzierte E-Mail-Kommunizieren. So und da ist halt viel Betreuung drüber passiert. Und jetzt muss ich gerade mal überlegen. Ja, und allgemein fanden die Leute es halt auch super, dass - bei manchen Praktikumsterminen bist du ja immer zu einer bestimmten Zeit an einem bestimmten Ort und dann musst du das da machen. Also Konzepte gibt es ja auch und die Leute waren halt auch cool, dass das so „on-demand“ ist. Also die kriegen halt die Aufgabe können die halt immer hochladen und haben halt auch sofort dieses Feedback. Also das hat den meisten dann sehr gefallen, dass... Sie laden ne Lösung hoch und müssen nicht irgendwie 3 Tage warten bis ein Betreuer jetzt sagt „Ja, richtig!“ oder „Falsch, Praktikum nicht verstanden!“, sondern dass man halt sofort das Feedback kriegt: „Ja, da haste jetzt die ganze Aufgabe richtig“, oder „Das hast du nicht richtig“. Und dann müssen die auch bei den anderen Sachen halt keine Zeit mehr verschwenden und da rum gucken, weil dann wissen sie halt, wenn der Test sagt, dass diese und diese Unit zum Beispiel oder Element ist jetzt korrekt programmiert oder zumindest funktional korrekt, dann - ja müssen die da nicht mehr reinschauen und können sich halt den Tests widmen, die sagen da fehlt noch was. Und das fand die Leute wirklich gut.

I [00:15:44] Gut finden, wenn man ein Student nicht so zufrieden ist. Wie kann der sich denn beschweren?

B [00:15:49] Ja, es kommt drauf an Beschwerden beschweren „womit“? Also wenn er jetzt z. B. sagt bei mir - dieser Test, der wird nicht grün, der ist irgendwie falsch oder kaputt oder was - dann sagt er natürlich meistens dann einem von uns Betreuern erst mal Bescheid. Häufig auch über Discord jetzt, wie gesagt, die können aber auch über Mail schreiben. Egal. So, und dann guck ich mir das Zeug erst einmal an. So hat er

erst mal recht, oder ist der Test vielleicht doch nicht kaputt? Und der Student hat irgendwie komische Vorstellungen von dem, was der Test machen sollte? Also so eine Art von Beschwerde. Dann gucken halt einfach nach, ob es stimmt und wenn es nicht stimmt, also der Test ist in Ordnung so was. Wenn er jetzt sowas sagt wie. „Ja, ich dachte ja, die Abgabe wäre jetzt zum Beispiel letzte Woche. Aber ich hab's nicht gesehen und das wurde nicht kommuniziert“ und sowas. Sowas ist halt schwierig, da bin ich dann weniger beteiligt. Das geht dann also wenn es um so organisatorische Beschwerden geht, sage ich mal, dann geht das halt an Herr Bente. Bei mir sind das eher Beschwerden wie „Ah, der Mist klappt nicht, oder?“ Ja, das funktioniert nicht. Dann hab ich da mehr so die Rolle, das zu fixen. Also falls das mit Beschwerden gemeint ist.

I [00:17:10] Ja, ich dachte, es gibt auch, also - wenn ich es richtig verstanden habe kommen die Studenten ja nicht direkt mit dem Tool in Verbindung?

B [00:17:16] Genau. Nicht direkt,

I [00:17:17] weil sie sonst wäre die Frage, ob es da so ein - ich sag mal so einen „Beschwerde-Knopf“ gibt oder sowas, dass man sich dann melden kann.

B [00:17:23] Ach so.

I [00:17:24] Was automatisiertes. Aber das läuft ja dann über Discord wahrscheinlich.

B [00:17:28] Ja genau. Oder halt E-Mail. Also wir wollen den Leuten ja nicht Discord aufzwingen.

I [00:17:34] Und wenn einer sagen würde, ich finde das Tool an sich nicht toll, ich würd lieber ich sag mal persönlich betreut werden, so wie man es kennt mit so einer normalen Abgabe, wo jemand drüber schaut.

B [00:17:45] Ja und ich glaube da kommt es dann auch mehr auf die Masse an. Also wenn jetzt sag ich mal 80 Leute sagen ja wir finden das gut und dann sagen wir mal 5 Leute ja, ich hätte das andere lieber. Manchmal kannst du nicht allen recht machen dann. Ich meine wahrscheinlich vor allem Herr Bente hat ja auch den Vergleich zu vorher und sieht wahrscheinlich mehr die Vorteile/Nachteile und das wird dann sicher auch eine Rolle spielen. Also nicht nur die Leute, die sich dann beschweren, wenn jetzt sage ich mal sagen 80 Prozent der Leute. Ja, das mit dem Automatisieren ist blöd, wir wollen alle jetzt lieber Gruppenarbeit und lieber mehr so und hier? Dann würden wir es wahrscheinlich auch wieder ändern. Aber, ... Ja, dann haben wir immer noch das Problem mit den Abgaben. Aber Hauptaspekt beim Studium ist ja auch, dass die Leute lernen. Wenn die Leute alle sagen das ist doof, so wie ihr das macht, dann lernen die halt auch nichts. Dann haben wir ja nur die Klausur, irgendwie und das ja nicht sind vom Studium. Aber - hatten wir tatsächlich noch nicht, sodass die Leute sagen, wir mögen dieses Konzept gar nicht mit den Tests und so. Außer, dass die Leute sowas sagen wie „wir mögen, Git nicht“ oder sowas. So ein Feedback. Da kann man halt nichts machen. Also zum Beispiel wenn bestimmte Tools nicht gemocht werden, die aber halt definitiv essenziell sind für einen Programmierer. Da bringt halt die Beschwerde nichts. Die Leute müssen halt Git können, da führt kein Weg dran vorbei.

I [00:19:01] Ja, auf jeden Fall. Das stimmt. Die nächste Frage wäre jetzt, ob. Ja, wie die Aufgaben bei den Studierenden ankommen, aber Ich denke das hatten wir. Die Mitarbeiter und Dozenten, die sich nicht mit diesem System auskennen, aber trotzdem gerne nutzen würden, weil - bringt ja seine Vorteile mit sich - glaubst du, die würden diesen Mehraufwand in Kauf nehmen? Also das sich anzueignen, wie das System funktioniert?

B [00:19:34] Ich glaube prinzipiell erstmal ja. Das hängt aber auch ein bisschen von der Weiterentwicklung ab. Das heißt, momentan haben wir auch so ein paar Hürden drin. Halt was die Bedienung angeht, sag ich mal. Von Leuten, die nicht mit dem Tooling so vertraut sind. Die das halt nicht so aufgezogen, sag ich - aufgezogen haben. Wie sag ich mal, als wir drei dann. Die werden sicherlich ihre Probleme haben und die können wir halt auch jetzt vielleicht noch gar nicht identifizieren oder wissen, weil das muss man dann einfach praktisch mal ausprobieren und dann Leute nach Feedback fragen. Natürlich auch. So und da sind wir halt bald auf dem Stand, dass das vielleicht dann auch mal geht. Also das Ding ist halt - bis jetzt kann man den Leuten ja nichts „ein Viertel gares“ vorsetzen. Also da kriegt man halt kein gutes Feedback, weil sowieso an allen Ecken was verbessert werden müsste. So, das heißt, wenn man das so ein bisschen schafft, dass das Tool halt besser nutzbar ist von anderen, dann, glaube ich - warte jetzt habe ich die eigentliche Frage vergessen.

I [00:20:29] Ob der Mehraufwand-

B [00:20:31] Achso ja, genau. Genau. Wenn wir, wenn man das Tooling hat und so ein bisschen bedienbar macht, dann glaub ich auf jeden Fall. Also ich hab ja auch drei Experteninterviews geführt im Rahmen der Masterarbeit und da hatte ich eben auch eine ähnliche Frage drin. Da hab ich auch gefragt, ob die Leute den Mehraufwand haben. Also ich hab jetzt nicht mega viele Leute interviewt, sondern tatsächlich vier. Aber die war auf jeden Fall alle - Zumindest klangen sie sehr bereit, das zu tun. Und die sagten eben auch, dass man vielleicht nicht unbedingt immer mehr Aufwand hat, denn oft auch bei Klausuren ist das so, die muss man oft neu stellen, weil Aufgaben sind bekannt. Also ältere. Und dann würden die Leute halt nur noch irgendwie die auswendig lernen, weil die irgendwie kursieren oder auch genauso bei Praktikumsaufgaben. Wenn die halt irgendwann bekannt sind, dann würden diese Leute einfach die Lösung kopieren und geben die ab. Fertig. So, und wenn man dann mehr Aufwand reinsteckt und da vielleicht auch relativ stark individualisiert vor allen Klausuren, weil da will man ja sowieso keine Teamarbeit, da kann man vielleicht auch eine Klausur mal öfters guten Gewissens nutzen. Also manche nutzen die natürlich auch so noch, ohne dass da irgendwas anders ist und dann schreiben die Leute halt einfach, wenn sie die alte Klausur kennen, da so hin, was sie da wissen. So, aber vielleicht kann man so halt auch Aufwand sparen, dass man vielleicht nicht jedes Mal eine neue Klausur erstellen muss, weil man eben vorher so viel Mehraufwand betrieben hat, um die zu individualisieren. Also man muss ja auch ganz klar sagen, der Haupt-Mehraufwand kommt vom Individualisieren, das einfache automatische Verteilen, das ist jetzt ja - eine normale Aufgabe muss ich ja auch so stellen und dann kann ich die jetzt halt auf Git verteilen. Das eigentlich nur so ein Plattformwandel. Ich könnte ja auch genauso gut wie zum Download bereitstellen, dann machen die die lokal. Und durch das Tooling können die dann halt eben auch Erfahrung mit Git sammeln. Mit GitLab und sowas und automatischen Tests. So. Und da hab ich ja erst mal nicht wirklich Mehraufwand. Natürlich muss ich die Tests auch schreiben, aber das müsste ich wahrscheinlich auch ohne das Tooling, wenn man so einen Testing-Ansatz

haben will und das nicht irgendwie selber prüfen will. Ja. Also ich glaube für den Fall, dass Leute, den Mehraufwand betreiben würden - wenn es sich lohnt und wenn es halt in Maßen bleibt - auch nicht frustrierend wird.

I [00:22:40] Ja, auf jeden Fall. Ist das denn geplant? Schon, dass das Fakultät übergreifend z. B. eingesetzt wird?

B [00:22:49] Wäre natürlich schön. Also vor allem Herr Bente ist da jetzt momentan auch verstärkt unterwegs. Dass man das ein bisschen bekannter macht. Also wir haben es in unserer Fakultät schon öfters anderen angeboten. Eine anderes Labor hat es auch mal ausprobiert, aber halt nun auch ein bisschen, ein bisschen weniger. Also die haben das tatsächlich glaube ich jetzt in ihrem Praktikum genutzt, aber halt weniger individualisiert, sondern mehr das automatisch verteilen genutzt. Also das die Leute auch wie gesagt auf GitLab kommen, dass die halt automatische Tests haben und sowas und halt mal so ein richtiges Projekt haben mit einer POM sag ich mal als Maven und nicht irgendwie so son. So ein 08/15 Projekt halt, wo man halt irgendwie dann 2 Punkt Java Files und wo hochlädt oder so. Und das haben tatsächlich dann auch noch andere an unserer Fakultät genutzt. Und wie gesagt, Herr Bente dabei, dass man ein bisschen bekannter zu machen und da planen wir irgendwann glaub ich auch mal so ein Paper zu schreiben, dass es irgendwie ein bisschen breitere Kreise erreicht und bis dahin auf jeden Fall mal schauen, dass man das halt auch den Leuten dann zugänglich macht. Also momentan ist es tatsächlich nur unserer Fakultät zugänglich ist das jetzt momentan unserem GitLab und Git Lab ist ja jetzt zu uns. Unsere Instanz sitzt nur aus der TH erreichbar. Und ja, vielleicht dann nochmal auf GitHub stellen. Da ham wir auf jeden Fall auch schon Organisationen erstellt, aber da ist halt noch nichts drin. Das kommt dann noch.

I [00:24:18] Alles klar. Gut. Dann. Welche Richtlinien musstest du einhalten, also z. B. bezüglich Datenschutzes oder Datensicherheit?

B [00:24:32] Ja. ... Es wird ja getestet und es sollte jetzt nicht so sein, dass Studenten gegenseitig gucken können "Was hat der jetzt alles grün? Was hat die alles rot" und solche Sachen. Das ist der Hauptaspekt. Und die Repositories die dann auf GitLab erstellt werden, da soll z. B. auch niemand in den Code von wem anders gucken können, es sei denn ist jetzt natürlich Gruppenarbeit. Also explizit, dass drei Leute an einem Projekt arbeiten würden, das haben wir eigentlich weniger. Deshalb hat jeder sein eigenes Projekt. Da ist die Anforderung, dass andere da nicht reingucken können. Das heißt, das muss privat sein, mit dementsprechenden Rechten. Es gibt auf GitLab verschiedene rechte Stufen. Es gibt auch Projekte, da kann dann jeder einfach reinschauen und das können wir definitiv nicht haben. Und dann darf natürlich auch nicht irgendwie der Name im Repo oder so stehen. Also da haben wir tatsächlich dann so eine UUID im Repo, die dann jemanden zugeordnet wird. ... Eigentlich können dann nur wir Betreuer sehen, die Admin auf der Plattform sind, von wegen „wem gehört dieses Repo?“. Also wir sehen dann die Member und das ist Repo überhaupt da ist. Jeder andere sieht dann immer nur sein eigenes [Repo] wo er auch Rechte hat. Obwohl im Ordner sag ich jetzt mal 100 oder 150 Sachen drin sind. Sehen die anderen einfach nicht. Also viel mehr Datenschutz ist da eigentlich nicht zu beachten. [Es] hat jeder sein eigenes Projekt, eigene Rechte und es ist dann schon sehr getrennt. Das handelt dann auch hauptsächlich GitLab. Also da haben wir gar nicht so viel mit am Hut.

I [00:26:14] [Es] wäre nur die Frage, wenn das dann größer wird und dann auch fakultäts-übergreifend, wie das dann gemanagt wird.

B [00:26:21] Ja, das kommt immer drauf an. ... Das [Tooling] kann man auch erweitern, aber momentan ist das auf GitLab basierend, ... weil man kann einfach bei GitLab Dinge in Gruppen ordnen, das heißt für bestimmte Meilensteine im Praktikum kann man immer Gruppen erstellen. Dann hat man auch eine private Instanz, das geht bei GitHub auch nicht. Das heißt da ist auf jeden Fall der Datenschutz schon gut, weil wir eben das nicht in fremde Hände geben müssen. Wobei ich nicht glaube, dass GitHub Datenschutz schlampig umgeht, aber wenn man die Selbstkosten-Distanz hat, dann muss jeder für sich schauen, dass man diesen Datenschutzrichtlinien entspricht. Also wahrscheinlich würde eine andere Hochschule, die unser Tool nutzt, nicht z. B. unseren GitLab nutzen. Die wollen vielleicht einen eigenen haben. Das Tooling ist jetzt z. B. auch etwas, was man lokal ausführt und was dann auf GitLab, die Sachen per API erzeugt. Und das heißt, wir kommen auch gar nicht an die Namen ran, die andere Hochschulen eintragen. Das ist keine Website oder so was, was dann bei uns gehostet wird und erzeugt dann auf deren GitLab irgendwas. ... Also ich sage z. B. im Tool [diese drei] deren Namen eingetragen wurden, die kriegen jetzt ein Repository und das ist nur lokal. Das kriegt niemand anders zu Gesicht. Deswegen, dass es auch da sehr getrennt. Ja.

I [00:27:53] Gut, ich würde sagen, steigen wir mal mehr in das System ein. Was kannst du denn zur Systemstruktur von DiveKit sagen? Wie ist das ganze aufgebaut?

B [00:28:03] Ja, also es ist schon ein bisschen schwierig einzuordnen. DiveKit erstmal nur ein Tool was [ein] bestimmtes Testkonzept erlaubt. Z. B. finde ich GitLab gehört nicht zu DiveKit, weil man könnte, wenn man wollte auch etwas GitLab ähnliches nutzen und da hat das gleiche Testkonzept drauf aufsetzen. DiveKit wird genutzt, um es da zu erzeugen auf dieser Plattform. ... Hauptsächlich haben die Leute z. B. mit GitLab zu tun. Da liegen die Repos eben drauf. Ich führ DiveKit lokal aus und erzeuge dann darauf Repositories. Das heißt alles was git-basiert ist, sollte eigentlich dann auch [da]mit funktionieren. Also ich könnte es vielleicht auch ans Laufen bringen mit z. B. GitHub und Jenkins daneben. ... Ich weiß, da es zwei getrennte Systeme [sind], es ist noch ein bisschen schwieriger. Aber z. B. GitLab hat CI direkt mit integriert. Das macht es dann ein bisschen einfacher. Also da sind eigentlich nur diese zwei Haupttools integriert, also GitLab und das DiveKit. Und dann gibt's natürlich noch einige Dinge, die relevant sind fürs Testkonzept. Aber ich weiß jetzt nicht, ob du das jetzt mit System interner Struktur sehen würdest.

I [00:29:23] Das ganze Konzept ist auch noch Teil des DiveKits, oder?

B [00:29:28] Ja, da sind wir uns glaub ich noch nicht so ganz [einig]. Das ist so ein bisschen Auslegungssache. Das DiveKit unterstützt ein Testkonzept, aber ist es nicht direkt. Also bei dem Testkonzept werden auch Tests ausgeführt. Und Tests, die haben wir ziemlich häufig in einer Library hinterlegt, dass man die auch mal wiederverwenden kann. Diese Test Library finde ich gehört nicht zum Tool DiveKit direkt, weil die programmiersprachenabhängig ist. Es geht nicht Tests zu schreiben, die programmiersprachenunabhängig sind. Das DiveKit an sich ist aber programmiersprachenunabhängig. Deswegen versuche ich das definitionsmäßig zu trennen. Das DiveKit ermöglicht dieses Erzeugen von den Repos und dass man auch diese Tests automatisch ausführt, aber das ist wieder ein anderes Paar Schuhe. Das

unterstützt es, dass es geht, aber würde theoretisch auch ohne dieses Tool DiveKit gehen. ... Also ich könnte auch manuell ein Repo auf GitLab [anlegen] und dann ... referenzier ich in einem Spring Projekt oder was auch immer, auf meine Test Library und führt ein paar Tests aus. Das könnte ich auch ohne großes Tooling machen. Das DiveKit ist mehrheitlich Unterstützung dahinter, dass man das eben automatisch alles hinbekommt und nichts manuell machen muss. So sehe ich das von den Abgrenzungen her. Für das Testkonzept, ansonsten sind auch die Komponenten innerhalb GitLab relevant, was die einzelnen Repos wären. ... Wir unterscheiden immer zwischen Code Repo, das ist das wo die Studenten ihre Aufgabe sehen und auch bearbeiten. Dann ein Test Repo, das hat ein paar andere Aufgaben. Das versteckt vor den Studenten ein paar Dateien, führt die Tests aus und erzeugt am Ende eine Testseite, auf der die Studenten Ergebnisse sehen. Das gehört natürlich auch alles eigentlich zum Paket dazu. Und das DiveKit ermöglicht, dass man das dann auch automatisch erzeugt.

I [00:31:43] Okay. Gibt es noch mehr Tools, die du bei der Entwicklung genutzt hast, wie du kannst oder die für die Weiterentwicklung geplant sind?

B [00:31:51] Ja, Wenn das in diese Richtung geht. Wir haben verschiedene kleinere Automatisierungstools entwickelt, die auf irgendeiner Anforderung [aus der Lehre] entsprungen ist, aufgesetzt wurde. Z. B. haben wir auch Klausuren über GitLab dann gemacht, vor allem wegen Corona. Die müssen auch irgendwie korrigiert bewertet werden und so weiter. Wir wollten auch, dass die Leute in den Klausuren viel mehr coden. Und es ist schwierig, Programmcode zu bewerten, auch im Sinne davon, dass es schwierig ist, dass mit Tests zu machen. Also wir können nicht sagen, „wenn der Test grünes gibt es 20 Punkte, wenn er nicht grün ist, gibt es null“. Das ist ja kein richtiges Bewertungsschema. Das heißt, wir haben ein sehr detailliertes Bewertungsschema, wo dann viele Dinge aufgeführt sind. Diese Kriterien, bringen dann tatsächlich immer auch nur einen halben Punkt oder einen. So war wie z. B., ist die Annotation gesetzt, wenn sie natürlich nötig ist oder Alternativen sind implementiert. Ich kann bei Java jetzt ein equals- oder hash Code als Annotation über eine Klasse schreiben, wenn man Lombok nutzt. Oder ich mache z. B. eine equals Methode ohne Hash Code Methode in den Code selbst rein. Es wären zwei Möglichkeiten und die wären dann ein Kriterium. ... Es gibt ein Punkt, wenn jetzt jemand equals implementiert hat und das Problem ist, dann hat man am Ende eine sehr große Liste an Tabellen. Also sagt man z. B. Aufgabe 1, eine Tabelle oder 3 und dann hat jede Tabelle Kriterien und das muss man am Ende natürlich zusammenzählen. Das manuell zu machen wäre kein Spaß und auch unnötig. Dafür haben wir z. B. ein Tool geschrieben, was dann, die einzelnen Repos der Studenten, wo dann die Bewertung liegt, die Punkte der zusammenliest, pro Aufgabe zählt und am Ende einen Report ausgibt, von wegen „der und der Student hat von der Aufgabe 1 20 von 30 Punkt, Aufgabe 2 so und so viel und insgesamt so viele Punkte.“ Also dass wäre z. B. so ein Tooling haben, was sich da ganz gut interagiert, was aber nicht in das eigentliche DiveKit Tool integriert werden sollte, weil das was ganz anderes ist. Dann müsste man da irgendwie eine intelligente Lösung finden, wie man mehrere kleinere Tools oder mehr so Features oder Funktionen zusammen integriert. Aber das ist noch was anderes.

Wir haben wie gesagt mehrere dieser kleinere Tools entwickelt, die jetzt gerade irgendeine Anforderung, welche jetzt gerade da war und ... schnell implementiert. ... Die haben sich dann so ein bisschen eingefügt in unsere Infrastruktur. Da das nochmal ansteht, fällt mir gerade ein, wir haben noch ein Tool. Z. B. müssen wir manchmal auch Dinge manuell korrigieren. Es ist nicht alles automatisch prüfbar. Da hatten wir ein Tool, das in die Repos der Studenten schaut und guckt, wann sie das letzte Mal bspw. committed haben. Dann kriegen wir eine priorisierte Liste und [entscheiden] wer als nächstes korrigiert wird. Das ist z. B. auch so ein kleines Tool, was wir mal geschrieben hatten. Das existiert dann eher am Rande ... aber das es braucht man nicht unbedingt, um das Testkonzept an sich ans Laufen zu bringen.

I [00:35:29] Dann. Bestehende Technologie, die angewandt und eingebunden werden soll. Noch zukünftig.

B [00:35:39] Das weiß ich gar nicht, ob wir da groß was künftig anbinden. ... Ich glaube, es geht erst mal darum, die bestehen Technologien, die integriert sind, nochmal rund zu machen, also die Nutzung von diesen. Ich wüsste gar nicht, was man da noch für Technologien bräuchte. Ich hätte z. B., das ist jetzt in ... TypeScript geschrieben, das Programm. Ich wollte es eigentlich erst in Java geschrieben haben, aber aus diversen Gründen hat sich da Typskript viel besser geeignet. Und dann die andere Technologie wäre GitLab. Das heißt, man hat eigentlich Typskript und GitLab. ... Je nach Programmiersprache natürlich muss man ein paar Dinge anpassen. Also jetzt nicht am DiveKit an sich, sondern mehr z. B. die Tests. Diese müssen dann anders gemacht werden. Die sind sowieso von Aufgabe zu Aufgabe teilweise unterschiedlich. Das heißt, wenn jetzt jemand sagt, „Wir wollen jetzt hier mit aber C# Praktika abbilden“ dann bräuchte man als neue Technologie natürlich in die Richtung Dinge. Aber das DiveKit an sich, was eigentlich nur die Repos verteilt, individualisiert. Obwohl, da fällt mir etwas anderes ein. ... Wir brauchen bessere Sprachunterstützung beim DiveKit. Also wenn wir Aufgabentexte schreiben und vor allem Herr Bente macht ja auch viel mit DDD. Da schreibt man mal gerne ausführliche Aufgabentexte, die man auch braucht. So, und wenn man die jetzt individualisieren will, dann wird das sehr schwierig. Wir individualisieren ja über Platzhaltervariablen. Das heißt auch immer so mit Dollar, also „Dollar Variablennamen Dollar“ und wenn die im Text sehr verteilt sind und je nachdem wie die heißt, wäre das sehr blöd bei der Aufgabenerstellung. Das heißt da muss man darauf achten, dass man die Dinge einfach hält, damit man ... noch durchblickt als Aufgabeersteller. Was da cool wäre, wir wären so Technologien, die die Sprache unterstützen, irgendwie miteinbringen, die vielleicht auch Grammatik Unterstützung bringen. Vor allem im Deutschen ist das sehr schwierig. individualisierte Texte zu erstellen. [Zum einen] sind da Artikel, die ändern sich und irgendwelche Fälle und was weiß ich nicht alles was die deutsche Sprache da so zu bieten hat. Da wäre Toolunterstützung nice oder Technologien, die das vereinfachen. Da sind wir aber noch relativ am Anfang und da kamen [mir] jetzt keine konkrete Technologie im Sinn, die das hat sich so umgesetzt würde, wir es gerne hätten.

I [00:38:22] Gut. Kann die Anwendung der Plattform unabhängig genutzt werden?

B [00:38:41] Genau das ist ja gerade das, was ich eben schon so ein bisschen angedeutet habe. Das DiveKit an sich auf jeden Fall, also [zumindest] plattformunabhängig im Sinne von programmiersprachenunabhängig. Aber ich könnte es jetzt z. B. nicht einfach auf GitHub loslassen. Das ist natürlich schwierig, weil da eine ganz andere API hinter ist. Momentan geht es nur lokal, das heißt, theoretisch

könnte ich mir auch jetzt lokal 20 verschiedene Varianten von Aufgaben erzeugen und die dann z. B. auf Ilias hochladen. Das könnte ich auch machen. Aber richtig plattformunabhängig im Sinne von das wird direkt auf die dementsprechende Plattform gepackt, also z. B. Ordner in Ilias oder Repos auf GitLab oder Repos auf GitHub, dass das geht leider nicht. Wie gesagt, da muss man immer wieder eine dementsprechende API implementieren. Das kann man natürlich tun, da ist nur die Frage, wie viel Aufwand es ist und ob es überhaupt übertragbar ist, z. B. GitLab hat ja Repositories GitHub auch Ilias jetzt nicht. Das ist da nochmal grundlegend anders, wie einfach die Dinge dort gesehen werden. [Bei] Dateien und Ordner, da weiß ich jetzt nicht, wie gut es darauf übertragbar ist. Aber alles, was irgendwie so Pipelines mit drin hat, Repositories mit git oder vielleicht auch etwas anderes als git, ... dass sollte dann schon relativ einfach übertragbar sein auf andere Plattformen. Es hat aber programmiersprachenunabhängig. Ich kann jetzt auf GitLab auch C#, ... Scala, Kotlin, was weiß ich alles machen. Kotlin wird jetzt schon unterstützt, weil auch die ganzen Tests und so eh ... für Java [ist] und das kann man einfach so in Kotlin dann übernehmen.

I [00:40:38] Fallen dir noch weitere Probleme ein, außer dass man jetzt überall die Schnittstelle bräuchte, wo man es integrieren will? Gäbe es noch andere Probleme, Überschneidungen, vielleicht irgendwas Technisches noch.

B [00:40:49] Aber wie gesagt, ein großer technischer Punkt ist vor allem die CI-Unterstützung, also die Pipeline-Unterstützung, weil die Tests, automatisch ausgeführt werden. Das macht eine Pipeline automatisch und generiert die Testseite. Das heißt [bei] einer Plattform, die keine Pipeline hat, wird es schwierig. Also z. B. wie Ilias, da wüsste ich nicht, wie Ilias mir meine Tests ausführen soll, automatisch. Ob man da Plug-Ins oder so selber schreiben müsste und das damit überhaupt klappt? Keine Ahnung. Z. B. GitHub hat ja auch keine integrierte Pipeline, wie GitLab das hat. Dann müsste man wie gesagt Jenkins nehmen und das wird sicherlich schwerer sein, da da die Schnittstelle implementieren als für Git Lab, denn GitLab hat es ja schon integriert. Das heißt, man müsste dann irgendwie die zwei über das Tool [integrieren]. Also dass man sagt „Okay, ich lad jetzt ein Repo auf GitHub, erstell jetzt, aber gleichzeitig ein Projekt in Jenkins“, da muss das ja irgendwo im Webhook verlinkt werden und was da alles Technisches anfällt. Das habe ich GitLab jetzt nicht. Das wäre auf jeden Fall die Herausforderung, dass wenn man auf solche Plattform geht, die nicht so dieses ganze Funktionsfeld haben, wie GitLab. Das man das darauf ebenso einfach übertragen kann, das glaub ich mich nicht. Selbst wird es mit Schwierigkeit verbunden ist nicht unmöglich.

I [00:42:09] Gut. Wir hatten jetzt mehrmals über die Funktionalität vom DiveKit gesprochen. Kannst du das nochmal ... alles was dir dazu einfällt, wo du sagst das gehört auf jeden Fall zum DiveKit, [zusammenfassen]. Das hast du ja gerade eben auch schon abgegrenzt, [dass] mit dem Test Konzept.

B [00:42:31] Genau, also ich kann es ja nochmal so vielleicht ein bisschen strukturierter darlegen.

I [00:42:36] Das wäre super.

B [00:42:37] Also einmal gibt's ein bestimmtes Testkonzept. Das Testkonzept ist bestimmten Anforderungen entsprungen. ... Wie gesagt, es gibt einen Code Repo,

Test Repo und dann die Testseite. Im Code-Repo laden die Leute ihre Lösung hoch. Das Test Repo testet, führt das aus und die Testseite stellt es eben dar. Die Anforderung von den ich jetzt gesprochen hatte, sind z. B. Studenten dürfen nicht einfach ihre Tests manipulieren und so Dinge. Das heißt, wenn ich jetzt Test in ihrem Repo habe, die automatisch ... auf der Testseite aufgeführt werden, dann können die in den Tests auch einfach irgendwie den Code raus löschen und die [Tests] werden immer grün. Das wäre natürlich eine Art von Pfusch. Das ist eben das Testkonzept, das es gibt. Dieses Testkonzept wäre manuell zu erstellen ist kaum möglich. Ich müsste für jeden Studenten hingehen, müsste die Code Repo manuell erstellen des Test Repo erstellen und müsste die zwei auch verlinken, weil wenn ich als Student aufs Code Repo etwas pushe oder committe, dann kriegt ein anderes Repo davon erstmal nichts mit. Das heißt es meldet quasi dem anderen Repo „So auf mich wurde etwas gepusht, du darfst jetzt die Tests ausführen“ und so Sachen. Also das heißt, das geht manuell sehr schlecht, ... das wäre auch sehr fehleranfällig, weil schon wenn eine Verlinkung fehlt, dann committed und der Studenten sagt „meine Testseite aktualisiert sich nicht“. Solche Probleme wären total nervig immer wieder zu fixen. [Es müssten dann] irgendwo irgendwelche Umgebungsvariablen ... gesetzt [werden], sodass man da jetzt machen [kann]. Wie gesagt das manuell zu machen ist blöd, deswegen gibt es DiveKit. Das macht das für einen. Und im DiveKit sagt man dann z. B. so Sachen wie „Welche Aufgabe willst du überhaupt jetzt vervielfachen?“. Man hat ein Repository, wo die Aufgabe, die Aufgabenstellung und die Tests usw. alles drin sind und das gibt man eben an. Dann sagt man, das nutzt man jetzt mal als Ursprung für alle Repos, die jetzt die Studenten kriegen. Dann muss ich sagen „wer bekommt denn alles so ein Repo“. Das heißt, dann gebe ich dann eine Liste von Namen an bzw. jetzt in unserem Fall sind das CampusIDs. Könnte jetzt aber auch hier nach anderem System auch irgendwelche anderen Namen sein, vielleicht, vollständigen Namen oder Emails keine Ahnung. Das ist der Identifier für die [Studierenden]. Die gebe ich dann da an. Danach konfiguriere ich natürlich noch so ein paar andere Sachen. ... Also wir gehen jetzt davon aus, die Aufgabe schon fertig. Dann führe ich eben das DiveKit aus und das erzeugt dann automatisch auf GitLab für jeden Studenten ein Repository, wo er mitarbeiten kann. So, das wäre so also der Prozess, wenn das Repo schon steht. Wenn es noch nicht steht, also wenn die Aufgabe noch nicht fertig ist, muss ich die als Lehrender erstmal irgendwie konstruieren. Da will ich ja aber auch schon testen, was das DiveKit z. B. an Individualisierung aus den Aufgaben machen würde. Und da kann ich ja nicht hingehen und das immer direkt auf GitLab hochladen, auch nicht zehnmal und was auch immer. Ich würde ihn ja komplett zumüllen, das heißt solche man kann es auch erstmal lokal ausführen lassen, dass man sagt, „Erzeug mir mal so 10 Varianten von meiner Aufgabe, die ich jetzt eben bisher erstellt habe. Und ich guck mir das mal an, was jetzt dabei rauskommt.“ Wie gesagt, das wird dann lokal, es erstellt dann 10 Ordner und dann kann man eben den Ordner rein browsen und gucken „welche Dateien sind jetzt hier? Wie wurden die jetzt benannt?“. Je nach Individualisierung nennen sich auch Dateien um. Ich kann ja z. B. wenn ich jetzt sage ich habe einen Baum-abstrakt, dann kann ich sagen „ich habe jetzt eine Birke ... oder eine Tanne“. Keine Ahnung. Und dann kann man je nachdem auch schon Dateien umbenennen, wenn man den Studenten schon mal leere Dateien geben will, wo man sagt „ihr schreibt jetzt hier den Code rein“, dann muss bei dem einen das Birke heißen, bei dem anderen Tanne. Da kann ich eben schauen lokal, ob es tatsächlich auch so passiert, wie ich das erwarte. Also habe ich jetzt tatsächlich eine Datei Tanne, Baum oder was auch immer. Da kann ich eben schauen, erstens, wie die Individualisierung rauskommt und was da so passiert ist. Aber auch bestimmte andere Funktionen, die das DiveKit ausführt, kann ich auch schauen bspw. was daraus wird. Zum Beispiel

sollen bestimmte Dateien für die Studenten gar nicht zugänglich sein. Die kommen dann nur in Test Repo rein. Da kann man sie dann auch nachgucken als Lehrender. Da hat man ja Zugriff drauf, aber nicht ins Code Repo. Das wäre jetzt aber blöd, wenn man bei der Aufgabenerstellung [immer] zwei Repos erstellen müsste. Also ich sage, das kriegen alle die Studenten, das kommt jetzt das eine Repo. Das kriegen die Lehrenden, alles in das Repo. Das heißt, es wird alles in einem Repo gemacht und dann kann man eben über so bestimmte Modifikationen Dateinamen sagen, wer das jetzt kriegen soll. Also ob die Studenten kriegen oder beide oder nur die Lehrenden. Solche Dinge muss man natürlich auch erst mal lokal testen und dafür ist es gut, wenn man das DiveKit mal lokal ausführt. Und wenn man tatsächlich in den Dateien mal rum browses und schaut sieht das so aus, wie ich das jetzt erwarte. Es heißt, das macht man auch sehr oft bei der Aufgabenerstellung, wo ich dann auch für die Individualisierung Variablen nutze. Dann spreche ich von meinem Dollar Baum Dollar in meinem Aufgaben Text und da will ich natürlich auch sehen wieder auf dem Text aussieht, wenn ich das mal ersetze durch je nachdem was da gewürfelt wird z. B. eine Birke, das führe ich dann aus und gucke mir das lokal an. Wenn die Aufgabe dann am Ende steht und der Meilenstein auch veröffentlicht werden soll, dann kann ich das auf GitLab loslassen das Tooling.

I [00:48:18] Aktuell ist das nur für Praktika angedacht, richtig, also für Klausuren nicht?

B [00:48:23] Doch wie gesagt, wir haben das auch schon mal für eine Klausur benutzt. Also zweimal. Damit hat sich dann mehr individualisiert. Ziemlich viel. Und was da der Vorteil ist, die Klausuren wird individualisiert. Wie das mit den Aufgaben Pools auch in Ilias ist, das sind erst mal ganz viele verschiedene Aufgaben und jeder ... würfelt 10 Stück raus. Da ist schwierig, dass man da aufpassen muss, dass sie noch gerecht verteilt sind, also die Schwierigkeit oder auch der Umfang. Und da gibt es definitiv Mechanismen, natürlich, wie das auch umgangen wird. Also dass man da schaut, dass sie ähnliche Schwierigkeitsgrade hat. Das DiveKit ist noch ein bisschen anders. Man kann z. B. bestimmte Gruppen von Aufgaben definieren, dass man sagt, man kriegt jetzt aus der Gruppe eine Aufgabe, aus der Gruppe eine Aufgabe und dann kann man im Vorhinein schon schauen, dass das relativ ähnliche sind. Also der eine macht jetzt ... aus dem fachlichen Datenmodell ein logisches Datenmodell und der andere macht ... Use-Case Diagramm. Also da muss man schauen, dass es ungefähr gleich viel Arbeit ist, aber dann kann man das in kleineren Etappen definieren und hat nicht am Ende irgendein Problem, dass man in eine Komplexitäts- oder Zuordnungsproblematik reinläuft. Sodass man sagt, „Oh, jetzt ist das Problem wieder zu komplex. Also meine Aufgaben, die jetzt der Student kriegt, das muss das wieder raus und eine andere kleinere Aufgabe rein“ und so Sachen. Das Problem hat man dann nicht. Deswegen haben wir das natürlich auch schon für Klausuren genutzt, wo wir viel individualisiert haben und wo sowieso das Abschreiben schwierig ist, weil wenn man eine Lösung von jemand anders kriegt. Dann müsste man da so viele Begriffe ersetzen. Und wenn man dann vergisst, von dem Kopierten einen Begriff, den es nur in einem anderen Kontext gibt, zu ersetzen, dann fällt es auch auf, dass man kopiert hat. Also wenn ich jetzt auf einmal eine Birke in meinem Programm habe, obwohl ich eigentlich Tannen gewürfelt habe, dann ist natürlich sehr merkwürdig, dass das so passiert ist.

I [00:50:23] Auf jeden Fall. Da müsst ihr aber wahrscheinlich auch nochmal immer nachgucken, nach den automatischen Testabläufen, ob das auch so richtig korrekt ausgewertet wurde.

B [00:50:36] Ja, also häufig ist das so. Klar Testen wir lokal, ob die Individualisierung so funktioniert. Vor allem beim Praktikum ist das weniger problematisch. Da führt man das dann einfach mal auf GitLab aus und wenn sich keiner beschwert, dann ist gut. Wenn sich jemand beschwert, guckt man. Aber da ist es nicht so, dass ich mir jetzt 150 Repos angucke, ob das jetzt immer funktioniert hat. Wenn sich keiner beschwert, wird wohl nichts kaputt gegangen sein. Wir haben natürlich auch ein paar integrierte Mechanismen, die einen ein bisschen warnen, wenn etwas schief gegangen ist. Angenommen. In der Grundaufgabe stehen in der Aufgabenstellung, da sind überall Platzhalter drin, mit Dollar, die werden ersetzt und wenn es fertig ist, wird es hochgeladen. Wenn jetzt nämlich in dem Prozess irgendwo doch noch ein Dollar da ist oder eine Variable nicht ersetzt wurde, dann ist wahrscheinlich etwas schiefgegangen. Dann werden schon Warnungen ausgespuckt. Das hilft, dass man eben sowas verhindert. Also wenn man sich z. B. verschrieben hat, „da habe ich jetzt BaumM geschrieben anstatt Baum“. Dann wird er sagen „Moment in der Datei steht das und das“, da ist schon viel, was an Kontrolle nachträglicher wegfällt durch Mechanismen.

I [00:51:56] Ok, gut. Dann kam ja vorhin schon mal drauf zu sprechen wegen Datenschutz und Datensicherheit. Gibt es noch weitere personenbezogene oder sensible Daten im System?

B [00:52:13] Also in GitLab. Jetzt im Hauptsystem, da ist die E-Mail drin, die CampusID. Dann bei uns lokal auf dem Computer, ich meine, das hat man sowieso immer, wenn man Kurse managet, sind natürlich oft auch die Namen zugeordnet und einem Excel zu CampusIDs und sowas. Man muss ja am Ende auch eine Liste erstellen können. Z. B. Wer hat das Praktikum bestanden, wer nicht und solche Dinge. Das ist sowieso eher lokal. Ansonsten groß an Daten wüsste ich jetzt nicht, was es da gibt. Also in GitLab selbst, aber das ist ja auch immer jedem einzelnen zugewiesen. Also da kann ja nicht jeder jetzt schauen, wer ist denn so in diesem GitLab und sieht alle E-Mails? So ist das ja nicht.

I [00:52:56] Ja.

B [00:52:57] Aber wie gesagt, das ist das Einzige, was da an personenbezogenen Daten drin ist.

I [00:53:02] Alles klar.

B [00:53:10] Und halt je nachdem, ob jemand die Tests grün hat oder rot. Da hat man halt ohne Admin-Zugriff, aber auch gar keine Idee, welche UUID jetzt, oder welche Testseite wem zugeordnet wird. Das heißt, selbst wenn ich, die Tests sind ja dann public, die werden ja gehostet, dann von GitLab. Die sind public einsehbar, d.h. theoretisch. Wenn ich jetzt jemandem anders ein Link von meiner Seite schicke, dann kann der schon sehen, was da grün und was rot ist. Aber der weiß, wenn der den wieder weiterschickt, da steht halt einfach nur UUID in der URL drin. Das kann man halt nicht zuordnen zu irgendeiner Person, es sei denn man ist halt Admin auf GitLab.

I [00:53:47] Ja dann sind die Daten ja auch perfekt anonymisiert schon.

B [00:53:53] Ja.

I [00:53:54] Ok. Wir hatten auch gerade schon mal gesprochen, ob es Ideen für die Weiterentwicklung gibt. Da war jetzt z. B. ein Punkt wegen der Sprache.

B [00:54:07] Ja.

I [00:54:08] Fällt dir sonst noch was ein oder irgendetwas, was das beeinflussen könnte? Positiv wie negativ.

B [00:54:13] Also ich hatte so ein paar Sachen davon schon mal genannt, aber ich kann es ja mal in einen Kontext bringen. Also nochmal aufzählen. Das eine war zum Beispiel das mit der Sprache. Man braucht bessere Sprachen-Unterstützung, damit die Aufgabenerstellung besser funktioniert. Und da gehts nicht nur um so Sachen wie deutsche Grammatik. Also wenn ich mal ein oder eine oder ein Artikel ändern muss. Also nicht nur solche Sprachen-Unterstützung, sondern auch wir können z. B. mit dem Individualisierungsverfahren, die jetzt das Tool unterstützt auch Beziehungen individualisieren zwischen Objekten. Also Beziehungstypen. Das heißt manchmal kann man jetzt sagen, es herrscht zwischen dem und dem also Objekt A und Objekt B eine 1-zu-1-Beziehung manchmal eine 1-zu-n oder eine n-zu-m. Und das ist halt, wenn man die Aufgabenstellung erstellt, vor allem mit einer komplexen Aufgabenstellung, wenn man die nämlich sehr ausführlich ausformuliert, ist das sehr schwierig zu beschreiben. Also ich kann nicht auf einmal von einem Baum sprechen, der nur ein Blatt hat oder hat er einmal n? Also das geht vielleicht noch. Aber wenn es dann anfängt mit n-zu-m, jetzt gehört das Blatt zu mehreren Bäumen, das macht z. B. auch gar keinen Sinn mehr. Das wird sehr schwierig, das in der Aufgabenstellung zu verpacken. Je nach Aufgabenkontext auch. Und da wäre es natürlich auch interessant, ob es da nicht Möglichkeiten gibt, wie man so eine Beziehungs-Individualisierung, also dass man sagt man individualisiert Beziehungstypen. Dass wir das nicht auch besser in Aufgaben-Text integrieren kann oder da vielleicht auch so optionale Dinge definiert wie ja, aber z. B. diese und diese Variante macht gar keinen Sinn und ich weiß nicht wie sowas aussehen wird. Das wäre mal interessant zu wissen, ob das überhaupt möglich ist und wenn dann halt wie. Also was man z. B. momentan definieren kann ist, was für relevante Beziehungen es geben könnte. Also ich kann z. B. auch ausschließen, schon mit dem Tooling, dass eine bestimmte Beziehung dann auf n-zu-m geht. Das heißt, ich könnte sagen ja, die Baum Beziehung wird halt nur 1-zu-1 oder 1-zu-n. Aber es ist trotzdem schwer eine Aufgabenstellung zu beschreiben, weil mal würde ich zum Beispiel auch von einer Liste von Objekten spreche und mal nur von einem. Das heißt den Begriff Liste kann ich ja schon gar nicht verwenden in der Aufgabenstellung. Das sind halt so Probleme, in die man da läuft und da wäre Tool-Unterstützung cool, die das Problem so ein bisschen aufschlüsselt. Also das ist das eine. Also bei der Aufgabenerstellung auch vor allem beim Formulieren der Aufgaben-Texte, dass man da Unterstützung braucht. Das andere ist halt, was ich auch nochmal wichtiger finde ist, dass man das Tooling an sich bedienbarer macht für bestimmte andere Nutzergruppen, die jetzt nicht gerade, jetzt sag ich mal Fabian, Herr Bente und mich einschließt. Also dass man z. B. die Benutzerfreundlichkeit anpasst, so den Einstieg ins Tool, dass das schneller geht, irgendwie vielleicht ein paar Tutorials hat, vielleicht sogar ein Video oder auch das Tool an sich, dass das halt robuster ist, also das wenn mal ein Fehler auftritt, z. B. wenn ich das Programm nutze und es tritt ein Fehler auf, weil ich hab's ja entwickelt, ich weiß genau was muss ich jetzt machen. So mach ich mal eben da mal was anders und da mal was anders. Also jetzt nicht im Programmcode, aber in der Konfiguration vom Tool jetzt zum Beispiel. Jemand

anderes wäre da vielleicht überfordert. Der sieht diesen Fehler und denkt sich, „was mache ich jetzt? Keine Ahnung“. Und da ist glaube ich wichtig, dass man da die Bedienbarkeit von dem Tool soweit macht, dass auch andere sehen, oh, hier ist ein Fehler und genau wissen, wie muss ich jetzt darauf reagieren? Häufig sind so Fehler, wenn zum Beispiel mal das Internet ausfällt oder sowas. Oder GitLab geradezu viel Last hat und das Projekt konnte nicht erstellt werden und ist mitten in der Erstellung abgebrochen. Das heißt normalerweise, wenn man jetzt auf GitLab ein Projekt erstellt, dann muss man erstmal das Projekt an sich erstellen und dann macht man darauf ein Commit. Also das geht nicht in einem. Man kann nicht sagen, ich möchte jetzt als eine Transaktion ein Projekt mit dem und dem Inhalt erstellen. Wenn jetzt das Tool z. B. abschmiert, nachdem man das Projekt erstellt hat, aber noch keinen Commit drauf gemacht hat, hat man ein Problem. Und jemand anders, der halt nicht so vertraut mit dem Tooling ist, könnte da Probleme haben und sagen, was mache ich jetzt? Dann sage ich mal, lässt er das Tool nochmal laufen und dann meckert das Tool vielleicht auf einmal, moment mal, hier gibt es schon ein Projekt, da ist halt nur noch kein Commit drauf. Ob das alles so richtig ist. Und ich glaube, da muss man einfach noch so ein bisschen Robustheit schaffen, dass man immer das erwartete Verhalten vom Tooling zurückbekommt oder das immer das erwartete Verhalten geliefert wird, was man eben erwartet, als auch jemand, der jetzt nicht gerade einer der drei Entwickler ist. Also, dass man sagt so jetzt hat das nicht geklappt, dann gebe ich den Namen nochmal ein und dann erwarte ich halt auch, dass es jetzt da ist oder so. Das sind so die 2 größeren Baustellen. Das Leute auch schnell da, dass ich das geben kann. Hat das zum Beispiel eine gute Readme. Vielleicht mit Link auf ein YouTube-Video, wo ein Tutorial ist und dass die Leute direkt loslegen können. Das ist, glaube ich so die ideale Welt.

I [00:59:19] Aber konkrete Pläne gibt es da noch nicht, oder?

B [00:59:22] Also das ist, glaube ich, auf jeden Fall. Also diese Pläne, diese zwei sind schon relativ konkret. Also dass wir das mal gerne haben möchten. Ist halt die Frage bei dem ersten, das ich aufgezählt hatte, wie das mit der Machbarkeit aussieht. Beim zweiten ist die Machbarkeit auf jeden Fall gegeben. Also das ist auf jeden Fall, denke ich ein sehr konkreter Schritt, dass wir sagen, wir machen das mal robuster, damit das auch mal andere besser nutzen können. Das ist glaube ich schon sehr konkret dann. Jetzt sollte dieses eine Beispiel, was ich jetzt gebracht hab mit dem Projekt, was schon da ist und dann muss man damit klarkommen im Tool, das ist schon sehr konkret. Das muss auf jeden Fall, solche Dinge müssen eben behoben werden.

I [01:00:09] Gäbe es da noch andere Fächer, in denen man das DiveKit nutzen könnte, außer jetzt Programmierung?

B [01:00:15] Ja, also die Frage ist halt welche. Aber theoretisch ist das halt nicht komplett auf Programmierung ausgelegt. Es gibt halt, wie gesagt, ok das hatte ich nicht gesagt, drei verschiedene Arten der Individualisierung. Das heißt, einmal kann man immer (...) der Begriff mag jetzt vielleicht so ein bisschen objektorientiert klingen, aber man kann halt Objekte definieren. Und pro Objekt gibt's ja mehrere Variationen. Das heißt, ich kann einen abstrakten Baum definieren und mal ist das eine Birke, mal eine Tanne. Das wären dann Variationen. Das wäre eine Möglichkeit. Und das kann man vor allen Dingen auch auf viele andere Dinge übertragen. Z. B. gut, das hätte auch etwas mit Informatik zu tun, aber wenn ich z. B. Use Case Diagramme erstellen muss und da vielleicht verschiedene fachliche Anforderungen in irgendwelchen Anforderungs-Texten rein individualisiere, hat das ja erstmal nichts mit Programmieren

zu tun. Das geht natürlich so auch. Dann, was die Beziehung zwischen Objekten angeht, wenn man die individualisiert, ist natürlich schon sehr in Richtung Coding getrieben, aber man muss ja auch sagen, diese Konzepte des Programmierens, so sowas wie Beziehung, das leitet sich eigentlich aus der Realität ab. Also das ist jetzt nicht, nur Coding eigenes, was man da erfunden hat. Und das bildet ja schon so ein bisschen die Realität ab. Aber da bin ich mir nicht ganz sicher, wie man das z. B. in komplett anderen Fächern nutzen würde. z. B. in der Mathematik wüsste ich nicht, warum ich da jetzt eine 1-zu-1 oder 1-zu-n bräuchte, das wäre wahrscheinlich weniger sinnvoll. Und dann gibt es da noch die Logik Individualisierung, das wäre dann die dritte. Da kann man bestimmte Teile einer Aufgabe definieren, die optional reingeliefert werden, also jedem Studenten z. B. was anderes. Weil die ersten zwei Varianten Funktion über diese Platzhalter. Also da wird irgendetwas konkret ausgetauscht in Texten. Und diese Logik-Individualisierung ist mehr, der kriegt diese Datei, der kriegt diese Datei nicht. Z. B. Der kriegt die Aufgabe Use Case, der kriegt die Aufgabe Use Case nicht oder der kriegt die Aufgabe fachliches Datenmodell und der eben nicht. Sowas wird dann über diese dritte Art gemacht und vor allem diese dritte Art, die kann man auf alles übertragen. Das ist halt so ein bisschen, das hatte ich eben mal kurz angesprochen, in Richtung dieser Item Pools und Item Pools hat man ja auch fast überall. Das habe ich auch in der Mathematik. Das heißt, diese dritte Variante könnte man auch nutzen, indem man sagt, der kriegt die mathematische Aufgabe und der kriegt ja andere mathematische Aufgaben. Und vielleicht könnte man da diese erste Art der Individualisierung auch im Bereich Mathe nutzen, dass man sagt, der hat jetzt die Parameter für seine Mathematik-Aufgaben und der hat die Parameter. Auch so weiß ich nicht, wie das jetzt bei Bereichen wie Germanistik aussieht, weiß nicht, ob ich da überhaupt irgendetwas individualisieren kann, keine Ahnung.

I [01:03:05] War auch mehr so eine Idee, ob das vielleicht sogar noch fächerübergreifend genutzt werden könnte.

B [01:03:10] Ja gut, das könnte schon. Wie gesagt, das ist nicht rein auf die Programmierung aus.

I [01:03:19] Gut, ich denke, man muss ja auch nicht alle Tools nutzen, z. B. die Beziehungen, wenn man die nicht braucht (...)

B [01:03:28] Dann nutzt man die nicht genau.

I [01:03:31] Okay, gut. Kommen wir mal zu einer eher philosophischen Frage. Wenn du das DiveKit nochmal entwickeln könntest, würdest du etwas ändern? Und wenn ja, was?

B [01:03:46] Ich hätte direkt mit TypeScript angefangen. Tatsächlich hatten wir das schon in mehreren Kursen jetzt ausgeprobt, sag ich jetzt mal. Und im allerersten Kurs hatten wir erstmal nur die (Aufgaben?), da war nichts individualisiert. Da wollten wir nur Aufgaben auf GitLab haben mit ein paar Tests. Und da war wie gesagt das erste Problem, scheiße, wie kriegen wir denn jetzt 50 Repos auf GitLab? Das können wir nicht nochmal machen. Ja tatsächlich, das war auch jemand anders. Der hat halt so ein kleines JavaScript Skript geschrieben. Das war dann sage ich mal 30 Zeilen. Das hat einfach irgendwie eine for-Schleife da irgendwas halt geforkt tatsächlich noch. Und da halt ganz viele Repos erzeugt. Das war so dieser Ursprung in JavaScript. Dann habe ich halt das erstmal weiterentwickelt in JavaScript und es war einfach eine

Katastrophe, weil da viel Struktur reinzubringen (...) ach, ist halt irgendwie Käse. Dann dachte ich mir auch, ich probiere es mal mit Java. So und dann gings halt auch natürlich um die Library zu GitLab. Und das ist tatsächlich in Java hat die nicht alle Funktionalitäten tatsächlich. Also so Libraries die man findet, die halt die GitLab API irgendwie so wrappen. Da haben bestimmte Funktionalitäten gefehlt. Konkret war das glaube ich so, dieses Setzen von Umgebungsvariablen von Projekten damit halt das eine Projekt das andere Notification kann und so Sachen. So eine Art WebHooks. Das ging halt über die API die jetzt in Java zur Verfügung stand nicht. Und dann auch Handeln mit JSONs in Java ist natürlich auch super, aber nicht, wenn diese JSONs sehr dynamisch in der Gestaltung sind. Wir machen halt (...) ein Großteil der Individualisierung wird dort über JSONs gemacht. Das hat den Vorteil, dass man da sehr flexibel ist und relativ schnell auch. Also es ist natürlich ein bisschen technischer Ansatz, aber man kann halt in Form von JSONs sagen ich habe jetzt dieses abstrakte Objekt und das hat die und die Variation und dann kann ich dem z. B. beliebig viele Key-Value Pairs geben und der Key wäre dann immer der Variablennamen und Value wäre immer der Wert. Das heißt man ist sehr frei in der Gestaltung, auch welche Variablen man im Endeffekt haben würde, oder wie man das aufbauen möchte. Aber das dann mit Java auszulesen ist relativ schwierig oder relativ nervig und unübersichtlich, wenn man halt nicht eine konkrete Struktur hat, die man lädt. Also ich habe ja kein normales Java POJO und lad dann ein JSON und dann sind halt alle Daten drin und fertig. Sondern man muss dann halt ein bisschen abstrakter arbeiten mit den JSONs, mit irgendwelchen Nodes und keine Ahnung was. Und das ist z. B. in TypeScript viel einfacher, weil man da halt ein bisschen weniger auf die Typen guckt. Dann ist halt egal, was das ist. Dann kann ich das auch so abstrakt definieren, wie, ich habe einen Key und auf diesen Key folgt entweder ein String oder eine Number und eigentlich ist auch egal was drinsteht, so nach dem Motto. Also das ist halt Super gut. Ich hätte mir z. B. diesen Java Irrsinn da erspart. Groß anders gemacht? Weiß ich gar nicht, ob ich da groß was anders gemacht hätte. Manchmal kommen immer so Refactoring Aufgaben, die hätte ich mir ersparen können.

I [01:06:51] Sonst irgendetwas, womit du unzufrieden bist? Wo du sagst, das hätte nicht sein müssen?

B [01:06:57] Also ich hätte wahrscheinlich von Anfang an schon ein bisschen mehr auf Robustheit achten müssen, weil das ist mir dann immer so später bei der Nutzung aufgefallen. Natürlich habe ich da schon ein bisschen drauf geguckt. Also wenn ich jetzt z. B. 100 Projekte erzeuge und 1 klappt nicht, dann bricht es jetzt nicht ab und sagt ja, fang von neuem an. Aber trotzdem halt ein bisschen Robustheit mehr in den einzelnen Ecken des Toolings und dass man da halt das erwartete Verhalten hat, das man sich auch wünscht. Und dass so ein bisschen das Tooling das durchboxt, was man will. Da hätte ich wahrscheinlich von Anfang an ein bisschen mehr drauf achten können. Dann hätten wir jetzt wahrscheinlich ein paar Probleme nicht, aber ich glaube, da ist man im Nachhinein sowieso immer schlauer.

I [01:07:40] Ja, auf jeden Fall. Okay. Kommen wir zu den zwei letzten Fragen. Gibt es weitere Bereiche oder Personen, mit denen wir sprechen sollten, deiner Meinung nach?

B [01:07:59] Das ist jetzt etwas schwierig, weil mit Herrn Bente spricht ihr die ganze Zeit. Beim Fabian ist super, weil er halt inhaltlich viel mehr Durchblick hat als ich jetzt z. B. und auch schon viel länger dabei ist und er auch viel eher sehen kann, was das Tooling so für Auswirkungen hat auf die Lehre an sich. Also der weiß halt z. B. was

war vorher, wo viel mehr personennah betreut wurde. Was ist jetzt, wo viel mehr gecodet wird. Auch auf GitLab hochladen, zu viel mehr Plattform Unterstützung und so dabei ist. der sieht halt da diesen Übergang viel mehr, weil ich war da vorher nicht so beteiligt. Ich kam halt erst rein mit dem Tooling. Vorher war ich da nicht so eingespannt und halt nur auf Studenten Sicht natürlich, aber jetzt nicht auf lehrenden Sicht. Und da hat Fabian wahrscheinlich einige interessante Anmerkungen. Ansonsten wüsste ich jetzt nicht, mit wem ihr da groß sprechen könntet.

I [01:08:59] Der Fabian Krampe ist das, ne?

B [01:09:01] Ja genau.

I [01:09:03] Ja, mit dem Herr Bente dürfen wir leider nicht reden, der hat sich selber ausgeschlossen.

B [01:09:08] Ihr redet ja sowieso mit keinem mehr.

I [01:09:12] Ja, genau. Weißt du, wie lange das denn schon geplant, oder in der Entwicklung ist, das DiveKit?

B [01:09:20] Wie gesagt, da war ich ja dabei, quasi wo das dann so geplant wurde. Wie lange das ist? vielleicht anderthalb Jahre. Obwohl, das war eigentlich so ziemlich am Anfang von Corona im Januar. Also müsste es eigentlich so ein Jahr, drei Monate oder so sein. Also da war Corona noch gerade so, was ist das da, was da im Osten so passiert? Das wird mich wahrscheinlich nicht betreffen. So Gedanken waren da und da hatten wir halt schon überlegt, wie können wir die Leute mehr zum coden bringen und auch mit Individualisierung und so fancy Sachen. Da waren am Anfang noch so Gedanken beteiligt, wie man könnte ja immer Adjektive vor irgendwelche random Nomen schmeißen, also sowas wie großer Affe, kleiner Wal oder so. Und das in Code reinbringen. Also da waren so ein paar sehr basic Ideen, waren da noch präsent. Und das hat sich dann halt so entwickelt. Das ist seit ungefähr einem Jahr 3 Monaten.

I [01:10:20] Gut. Sonstige Anmerkungen oder Hinweise, was du ergänzen kannst noch?

B [01:10:28] Nee, glaub nicht. Mir fällt nichts ein.

I [01:10:33] Mir fällt jetzt. Ich habe mir soweit hier eigentlich auch alles abgehakt und Fragen zwischendrin reingeworfen. P hast du noch was?

P [01:10:40] Dann grätsche ich dir mal mit Bonusfragen rein.

I [01:10:42] Jawohl.

P [01:10:43] Wie ist das Ganze bisher programmiert? Läuft das einfach als quasi (...) Ich kenne jetzt tatsächlich GitLab nicht, ich kenne nur GitHub. Hast du das quasi so als Workflow oder so heißt es ja glaub ich in GitHub nur in GitLab programmiert in TypeScript oder warum ist das TypeScript gewesen? Oder läuft das ganze einfach auf irgendeinem Node Server?

B [01:11:05] Ja also das ist halt. Ja, es wird halt lokal bei meinem Rechner jetzt zum Beispiel, wenn ich das ausführe wird es lokal bei mir ausgeführt. Einfach mit Node und

dann halt. Also klar vorher von TypeScript auf JavaScript kompiliert und dann halt einfach mit Node zum Node-Server halt ausgeführt. Genau. Und, also es wird gar nicht vom GitLab Server ausgeführt und ja. Und in TypeScript (...).

P [01:11:34] Ja, ne, ich wollte ja nur (unv.) rhetorische Frage, also das Ganze läuft dann einfach über WebHooks, dass der lokale Server das dann anspricht.

B [01:11:44] Nein, der lokale Server, also das ist nicht im Sinne vom Server. Man kann mit Node ja auch einfach Programme, sag ich mal, ausführen. Dann ist es auch relativ schnell zu Ende. WebHook bräuchte ich ja eigentlich dann, wenn ich von GitLab irgendwas wissen will. Und eigentlich will das DiveKit von GitLab gar nichts wissen. Es wird halt über die API angesprochen, sowas wie z. B. ich erstelle jetzt ein Projekt und mache jetzt da zum Beispiel ein commit drauf. Da kriege ich ja immer über die Anfrage selbst der API direkt schon irgendwie einen return Wert, ist ja irgendwie ganz normal http anfragen. So, und da kriege ich eigentlich alles mit, was ich wissen will, wie hat der Commit funktioniert? Oder hat das Projekt erstellen funktioniert und so Dinge. Wenn die Projekte einmal auf GitLab erstellt sind und das Tool dann, sage ich mal, nach 20 Minuten irgendwie abschaltet, weil es jetzt alle erstellt hat, sagt irgendwie finished, wie lange es gebraucht hat und dann schaltet sich auch schon ab. Also da ist dann auch kein Node Server oder so mehr an.

P [01:12:39] ah, okay. Ich habe bisher so ein bisschen überlegt, wo so der Schnitt zwischen dem DiveKit und dem GitLab ist. Aber dann ist das DiveKit ja wirklich nur einmal kurz, bau mir die Ordner und dann ist es für den Rest des Semesters quasi aus.

B [01:12:53] Genau. Und das ist das Coole, dass wenn es einmal funktioniert hat und darauf läuft, dann braucht man das DiveKit auch gar nicht mehr. Also es sei denn man will natürlich neue Projekte haben, aber das würde dann auch, das ist nichts, was die ganze Zeit läuft im Hintergrund, sondern die Idee ist halt, dass man nicht irgendwelche Daten Synchronisation auch andauernd hat. Wie du schon sagst, da brauch ich vielleicht WebHooks oder was auch immer ich bräuchte zwischen GitLab und DiveKit und was auch immer, wo es dann halt Probleme geben kann, sondern wenn das einmal halt erstellt hat, dann gehts halt aus und dann hat halt GitLab, die Verantwortung und natürlich die Pipelines die da definiert sind. Dann hat man alle Sachen, die man braucht dort.

P [01:13:32] Alles klar. Das war so die größte Frage, ich scroll mal eben durch, ob hier noch irgendwelche Sachen waren. Das meiste haben wir dann halt immer so zwischendurch mal ohnehin beantworten können. Ich glaube es sieht gut aus.

I [01:13:52] Mir ist jetzt auch nichts anderes eingefallen.

B [01:13:55] Und wie gesagt, das hatte ich jetzt glaube ich vergessen zu sagen. TypeScript war die Wahl, erstens wegen dieser einfachen und tollen Behandlung von JSONs und zu dynamisch aufgebauten JSONs und zweitens die gute Unterstützung von APIs zu GitLab oder wahrscheinlich auch zu anderen Plattform. Die sind in sowas wie JavaScript natürlich super gut vorhanden, als im Vergleich zu anderen Programmiersprachen.

P [01:14:26] Ja klar, erinnere ich mich. Wir haben versucht irgendwas mit Java einfach nur ganz normale HTML GETs mit Java zu machen, das ist schon super exotisch da.

B [01:14:35] Ja, aber Java muss man sich oft richtig rein fuchsen, um einfache Dinge zu machen. Und bei JavaScript oder TypeScript ist es oft dann sehr einfach, weil das das Hauptgebiet ist von den Sprachen.

P [01:14:48] Ja klar.

B [01:14:48] Irgendwas mit HTTP und Web-Servern und hast du nicht gesehen.

P [01:14:56] Gut. Ich glaube, das wäre es dann auch von mir gewesen tatsächlich.

I [01:14:59] Warte mal. Mir fällt noch was ein.

P [01:15:01] Perfekt.

I [01:15:02] Wie nutze ich das DiveKit? Also ich glaube der Herr Bente meinte das ist ein Plugin in IntelliJ?

P [01:15:09] Stimmt, der Herr Bente hat erzählt, dass es da noch oder zumindest nicht. Das ist jetzt nur ein Programm gewesen, meintest du und Herr Bente meinte, da wäre noch so ein Plugin für IntelliJ in Arbeit, das dann irgendwie so oder zumindest geplant, dass man das als Texteditor benutzen könnte, wo man dann Sachen einfügt?

B [01:15:26] Ja, das ist halt. Ich glaube, es ist ein Informatik Projekt von einem anderen Studenten. Der macht tatsächlich ein Projekt. Das Problem momentan ist, man definiert bestimmte Eigenschaften von bestimmten Objekten in einer Config und dann verwürfelt der Dinge und der sagt jetzt z. B. Du hast das und das gewürfelt und generiert da auf der Basis Variablen und die Variablen-Namen ergeben sich dann quasi aus der Config. Die stehen nicht von Anfang an so im JSON, weil angenommen dieses Baum Beispiel jetzt, ich habe ein Oberbegriff Baum, dann habe ich mal eine Birke und mal irgendwie eine Tanne. Dann habe ich vielleicht die Eigenschaft von Blätter, oder was für eine Art von Blätter das ist. Dann habe ich bei beiden Objektvariation. Also einmal bei der Birke und einmal bei der Tanne habe ich so eine Blätter-Eigenschaft. Die Variable wird am Ende aber nicht Blätter heißen, sondern die Variable wird dann heißen Baum-Blätter, weil Baum dieser Überbegriff ist. Was ich eigentlich damit sagen will ist, dass diese Variablennamen sich erst über die Zeit ergeben. Also während das Tool einmal ausgeführt ist. Das kann man natürlich dann auch einmal das DiveKit ausführen, um zu sagen, generier mir mal zumindest die Variablennamen, damit ich weiß, was ich in meiner Aufgabenstellung nutzen muss, was für Variablennamen. Also das ich dann in der Aufgabenstellung nicht irgendwie mache Dollar Blätter Dollar, sondern Dollar Baum Blätter Dollar z. B. jetzt. Und da müsste man halt die Variablennamen kennen, um relativ schnell die Aufgabe erstellen zu können oder zumindest immer mal wieder im DiveKit gucken oder sich irgendwo abspeichern und immer wieder dahin schauen. Keine Ahnung. Und was da in Arbeit ist, ist ein Plugin, was Code-Vorschläge hat bezogen auf Variablen, die es schon gibt. Das heißt da würde dann, im Hintergrund läuft dann so ein Language Server halt. Ich bin da nur so halb, höre ich da immer mit was da so läuft. Dieser Language Server wird dann auch diese Variablen generieren immer wieder, wenn man die Configs bearbeitet und dann kann man während man eine Aufgabe erstellt in der IDE hat man dann halt Code Vorschläge oder halt die Vorschläge der Variablen kann man dann nutzen. Wenn ich dann z. B. nur b oder Dollar b eintippe, dann weiß der schon, dass

es die Variable Baum Blätter geben wird. So und das erleichtert halt schon die Aufgabenerstellung, im großen Maße. Das ist das Projekt.

P [01:18:00] Alles klar.

I [01:18:06] Gut, ich glaube das wärs dann.

B [01:18:10] Ja, das wisst ihr wahrscheinlich besser als ich.

I [01:18:15] Wenn P nichts mehr hat, dann. Ich habe zumindest nichts mehr.

P [01:18:22] Geht mir genauso.

I [01:18:23] Also die Aufnahme stoppen.

P [01:18:26] Genau. Dann beschäftige ich mich jetzt mal damit.