

# 16. Jan. 22.13.04 RES172\_HIFIS

Willkommen zum Forschungspodcast der Helmholtz-Gemeinschaft.

Ich bin Holger Klein.

Diesmal reden wir mal nicht über konkrete Forschung, glaube ich jedenfalls, dass wir da nicht drüber reden, sondern wir reden über Infrastruktur.

Die Infrastruktur, die den Forschern in der Helmholtz-Gemeinschaft in ihrem Arbeitsalltag so zur Verfügung gestellt wird, die sie brauchen.

In der Helmholtz-Gemeinschaft, da gibt es die, das den Hive ist.

Das sind die Helmholtz Federated IT Services.

Daran nehmen elf von 18 Helmholtz-Zentren teil, unter anderem das Deutsche Zentrum für Luft- und Raumfahrt und der Deutsche Elektronen-Synchrotron.

Uwe Jant vom DESI ist Koordinator für Hiveis und Carina Haupt vom DLR ist dort Gruppenleiterin für Sustainable Software Engineering und Mitarbeiterin bei den Software Services von an im Hiveis.

Hallo ihr beiden.

Hallo.

Hallo.

Ihr merkt schon, ich habe nicht die leisste Ahnung, worüber wir jetzt sprechen.

Ist es das Hiveis, die Hiveis, der Hiveis?

Artikel benutzen wir generell selten.

Ja, man kann glaube ich sagen, also die Federated IT Services, also das ist ja ein Plural, also aber eigentlich benutzen wir das tatsächlich, tatsächlich Artikel relativ selten und wir machen auch viel auf Englisch, muss man schon sagen, da fällt das gar nicht so sehr auf.

Arbeitet ihr denn am Hiveis, im Hiveis, für, mit oder wie nennt man das?

Weil das ist ja kein Institut in dem Sinne, ne?

Genau, also ich würde sagen in Hiveis.

Hiveis ist ja eine Zusammenarbeitsgruppe, Gruppenbezeichnung sozusagen, jetzt keine eigene Institution, deswegen.

Habt ihr denn Räumlichkeiten oder seid ihr so eine ganz virtuelle Geschichte, weil aus elf Zentren Menschen da mitmachen?

Wir sind komplett verteilt.

Jeder sitzt wirklich da, wo er auch sonst arbeitet, in seinem Zentrum und wir haben jetzt vor Corona aber versucht regelmäßig zusammenzukommen, auch in der großen Runde, damit man natürlich sich ja am Anfang bei einem Kick-Off kennenlernt und später dann auch immer wieder austauscht.

Aber sonst machen wir das alles, ja, via Tools, die uns unterstützen beim verteilten Arbeiten.

Kick-Off heißt, ihr seid noch gar nicht so alt?

Also das Hiveis startete ja im 2019, das heißt damals gab es eben die Kick-Off-Veranstaltungen, aber man muss schon sagen, dass ein Großteil des Personals,

das jetzt in Hiveis arbeitet, tatsächlich auch erst später dazukam und zwar eben als dann schon Pandemiezeit war.

Das heißt, viele kennen sich auch gar nicht physisch und haben dann sozusagen Hiveis auch fast nur virtuell kennengelernt und das ändert sich gerade jetzt erst in diesen Tagen langsam ein bisschen.

Und ihr arbeitet trotzdem zusammen.

Was genau arbeitet ihr denn eigentlich?

Was ist Hiveis?

Hiveis besteht aus mehreren Teilen.

Zum einen gibt es die Cloud-Services, dann gibt es Backbone-Services und die Software-Services.

Software-Services ist der Bereich, wo ich zusammen mit auch mehreren Kollegen aus meiner Gruppe tätig bin, wo ich am ehesten zu was sagen kann.

Vielleicht, Uwe, magst du was zu Cloud- oder Backbone-Services sagen?

Ja, Cloud ist so iCloud, ne?

Untertrott versteht jeder was anderes.

Bevor wir uns da in die falsche Richtung bewegen, kann ich da nicht.

Ich habe irgendwann mal zynisch gesagt, Cloud bedeutet, meine Daten liegen auf anderer Leute Rechner.

Das kommt auch hin.

Ja, das kann man tatsächlich so sagen.

Wobei wir natürlich den Anspruch haben, dass es nicht irgendwelche Leute sind, auf deren Rechnern das liegt, sondern dass wir das Ganze schon in Helmholtz behalten.

Also tatsächlich hat Hiveis den Anspruch und den Auftrag und tut das inzwischen ja auch, die Infrastruktur für die Nutzung von IT-Diensten zur Verfügung zu stellen, und zwar von Helmholtz für Helmholtz.

Also wir haben verteilte Fähigkeiten in Helmholtz, die vorher als Insellösungen teilweise schon existierten, aber der Anspruch ist jetzt, das für alle Helmholzer grundsätzlich und deren Kollaborationspartner zur Verfügung zu stellen.

Und dazu gehört einmal das reine Zugänglichmachen, das sind die Cloud-Services, also dieser Cluster, der dafür sorgt, dass man darauf überhaupt zugreifen kann und dass es diese Services gibt.

Und dann gibt es noch den Auftrag, Best Practices und nachhaltige Vorgehensweisen im Umgang mit solchen Services zu vermitteln, ganz besonders eben für wissenschaftliche Anwendungen.

Und da ist der Software-Cluster eben sehr stark aufgestellt.

Also zur Verfügungstellung und das Vermitteln von Best-Nutzungs-Practices dafür.

Der Cluster für die Cloud, ist das ein Zentralrechner oder ist der auch verteilt über alle Zentren?

Der Cluster, das ist natürlich ein... Du hast den Cluster gesagt.

Der Cluster bezieht sich hier auf eine Ansammlung von Arbeitsgruppen.

Also Cluster hat jetzt nicht die Bedeutung eines Rechensystems, sondern ist die Ansammlung von Arbeitsgruppen an verschiedenen Zentren.

Und vielleicht noch als... Carina hat eingeatmet.

Ja, genau.

Ich wollte mich noch ergänzen, warum das so wichtig ist.

Denn, dass wir unsere eigene Infrastruktur betreiben, auf Helmholtz-Ebene, ist halt extrem wichtig.

Denn wir dürfen oft auch nicht jeden Cloud-Dienst, den es so im weiten Web gibt, einfach so nutzen.

Es geht schließlich um Forschungsergebnisse.

Aber wir haben eben auch als Forscher besondere Bedürfnisse teilweise und möchten halt das Ganze auf unserer eigenen Infrastruktur haben.

Und das heißt, es ist einfach wahnsinnig wichtig, dass wir als Zentren uns solche Technologien selbst bereitstellen.

Und gleichzeitig ist es auch total gut, dass das Ganze auf Helmholtz-Ebene passiert und nicht jedes Forschungszentrum alle Dienste selbst hosten muss oder bereitstellen muss.

Denn es ist doch eine Vielzahl, die man ja so braucht.

Jeder, der irgendwie mal was entwickelt hat, jetzt allein im Softwarebereich kennt das oder jetzt auch beim verteilten Zusammenarbeiten, was es da nicht alles an tollen Tools und Werkzeugen gibt.

Und wenn das halt jeder für sich selbst macht, dann bedeutet das sehr, sehr viel

Aufwand und Arbeit.

Und so kann man sich das halt teilen, aber gleichzeitig sicherstellen, dass es eben nicht irgendwelche Daten an US-Unternehmen oder so gehen oder wer auch immer dahinter steht, sondern dass das Ganze wirklich bei uns in der deutschen Forschungsgemeinschaft bleibt.

[Siebert] Aber die Forschungsdaten sind doch letztendlich, ist das doch öffentliches Gut.

Wo ist da das Problem, wenn es öffentlich wird?

[Klein] Nicht alle Daten, die aus der Forschung fallen, sind öffentlich und auch nicht alle Software, die wir schreiben.

Ich bin zwar ein großer Verfechter von Open Source und Open Data und auch im Rahmen von HIFIS setzen wir uns dafür ganz stark ein.

Aber es gibt eben Sachen, die nicht direkt oder überhaupt raus dürfen.

Also bei mir, beim Deutschen Forschungszentrum für Luft- und Raumfahrt, dem DLR, gibt es halt doch auch einiges an industrienaher Forschung in Kooperation mit Partnern, die eben nicht einfach so raus darf.

[HL] Genau.

Und dazu kommt natürlich auch noch, dass die Forschung jedenfalls in ihren publizierfähigen Ergebnissen natürlich öffentlich sein muss, aber nicht unbedingt jeder Entstehungsprozess dahin.

Also man will sich ja auch finden, Projekte organisieren, man will Rohdaten verarbeiten und so weiter.

Das ist nicht zwingend von Anfang an immer öffentlich.

Und außerdem, wenn man Dienste nutzt, sowas wie amerikanische bekannte Dienste, geht es oftmals ja dann nicht nur um die Öffentlichkeitsmachung, die man vielleicht nicht gleich will, sondern auch um Rechteverwertung.

Also dass man da teilweise ja auch die Rechte an solchen Dingen abgibt oder nicht ganz sicher ist, wie frei das dann noch ist, was man da eigentlich gerade an Daten reinkippt.

[SD] Und vor allen Dingen nicht sicher sein kann, ob die das morgen nicht abschalten.

[T] Das auch noch, genau.

Also da sind viele Unsicherheitsfaktoren, die man zwar aus Bequemlichkeit oft ausblendet, dann macht man eben doch wieder sein bekanntes X-Docs-Dokument irgendwo auf, aber das ist natürlich nicht sehr nachhaltig aus verschiedenen genannten Gründen.

[HT] Moment, sein eigenes X-Docs-Dokument, das heißt, ihr bietet auch Dienste an, die völlig normal sind, die auch ich benutze, was weiß ich hier, Textverarbeitung, Tabellenkalkulation, um jetzt mal nicht Markennamen zu nennen.

Genau, sowas bieten wir auch an.

Das ist auch auf jeden Fall etwas, das sozusagen die low-hanging fruits für uns sind, weil das braucht jeder, jeder in jedem Zusammenhang mit Forschung, außerhalb Forschung, für die Vorbereitung von Forschung.

Und das ist auf jeden Fall auch Teil des Cloud-Service-Portfolios und wird auch intensiv genutzt, ja.

[SD] Aber das ist nichts, was ihr selber gecodet habt, oder doch?

[HT] Nein, das wir entwickeln, außer für diese zentralen Infrastrukturen, die das Ganze miteinander verknüpft, entwickeln wir aktuell nicht wirklich Dienste von scratch, das muss man ja auch nicht, weil vieles gibt es eben schon als Software, die aber open-source sein sollte und auch offene Datenstrukturen bieten soll, dass wir nicht irgendwie ein Vendor-Login oder sowas kriegen.

Und das implementieren wir, und wir implementieren das so, dass eben ganz Helmholtz darauf zugreifen kann, möglichst einfach, und deren externer Partner auch, wenn sie eingeladen sind.

[SD] Das heißt ganz konkret irgendwie Open Office, und ihr sorgt dafür, dass jeder Wissenschaftler, jede Wissenschaftlerin in der Helmholtz damit arbeiten kann?

[HT] Im Prinzip ja, ja.

[SD] Im Prinzip?

[HT] Naja, im Prinzip, das heißt jetzt nicht Open Office, das nennt sich dann Only Office und ist dann in eine Software eingebunden, aber letzten Endes ist es genau das.

Dazu kommen aber natürlich weitere Services, die über Versionierung von Codes zum Beispiel beinhalten, dass man eben auch kollaborativ an komplexeren Codestrukturen arbeiten kann, Programme für Programme oder für Projektmanagement.

Viele weitere Services, auch Rechenkapazitäten werden auf diese Art und Weise vermittelt, also wir können darüber eben auch Mitgliedern aus allen Zentren ermöglichen, Zugriff auf High-Performance-Computing-Ressourcen zu haben, und das mit möglichst wenig Aufwand, und das ist ja letzten Endes der Sinn der Veranstaltung.



[SD] Das heißt, ich muss nicht in Jülich anrufen und fragen "Darf ich mal eben?", sondern ich kann gucken, ob noch irgendwo ein bisschen Rechenzeit frei ist.

[HT] Tatsächlich sind die ganz teuren Ressourcen mit einem gewissen Antragsprozedere verbunden, damit das nicht komplett verstreut wird, aber tatsächlich sind die Hürden da sehr niedrig und es gibt jetzt auch kein grundlegendes Ausschlusskriterium in dem Sinne von "Wir lassen jetzt nur Zentrum A, B, C zu, aber Zentrum D leider nicht, weil es nicht".

Das ist tatsächlich so, dass wir da ganz hellem Holz versorgen wollen und das zum großen Teil auch schon tun.

[SG] Carina, du hast eben gesagt, die Bedürfnisse der Wissenschaftlerinnen und Wissenschaftler, welche sind das?

[C] Das ist sehr breit aufgestellt.

Uwe hat ja gerade schon das eine oder andere erwähnt, es geht halt um Computing Power, also irgendwo, wo man seine Berechnungen laufen lassen kann, aber es geht auch darum, eben zum Beispiel sein Code zu schreiben, gemeinsam da zusammen zu arbeiten.

Das ist natürlich, sag ich mal, ein klassisches Bedürfnis aus der Informatik.

Was nicht so klassisch ist, ist, dass die meisten Leute, die in der Forschung Code schreiben, gar keinen Background in der Informatik haben und dadurch entsteht auch eine andere Art von Bedürfnissen, nämlich es ist nicht nur das Bedürfnis, dass bestimmte Tools oder Services angeboten werden und verfügbar sind, sondern dass auch gezeigt wird, wie geht man damit eigentlich um.

Uwe hatte das vorhin schon mal kurz angesprochen, dass es darum best practices geht in dem Umgang und das bedeutet wirklich, dass wir Trainings geben, dass wir unterstützen eben, wie können diese Tools eigentlich angewandt werden, wie kann man sie benutzen, wie macht man das Ganze möglichst

effizient, wobei ich da nicht den Begriff der best practices, sondern good practices bevorzuge, aus dem einfachen Grund, dass ich der Meinung bin, es gibt immer gute Empfehlungen, die beste Empfehlung kann niemand geben, die ist nämlich ganz, ganz speziell abhängig von dem eigenen Use Case.

Und genau, so unterstützen wir die Leute, also die Forschenden dabei, diese Services effizient zu nutzen und somit auch effizienter ihre Forschung vorantreiben zu können, denn heutzutage ist Forschung ohne Softwareentwicklung in großen Teilen gar nicht mehr möglich.

Und genau, das ist jetzt so ein Aspekt, den wir anbieten, aber es gibt auch noch ein paar mehr, die wir auch anbieten.

Also es geht nicht nur darum, Dienste sozusagen aufzusetzen und verfügbar zu machen, sondern eben die Leute auch konkret dabei zu unterstützen, diese gut anzuwenden.

Wird das angenommen, also wird das nachgefragt oder müsst ihr rumrennen und sagen, hier hör mal, guck mal, lass mich dir mal womansplaining.

Also die Trainings sind sehr nachgefragt, also die sind fast alle ausgebucht und wir bieten die auch sehr regelmäßig an.

Mein Rechner macht gerade irgendwas, ich wollte gerade in Pause-Modus.

Also die Trainings werden sehr gut angenommen, wir sind fast immer ausgebucht, wir müssen oft schon Zusatztermine noch reinpacken und wir haben mittlerweile auch wirklich eine kleine Vielzahl an Trainings.

Wir vom DLR bieten relativ viele an, aber eben auch andere Zentren sind da aktiv dabei und wir unterstützen uns da auch gegenseitig.

Und ja, also das wird sehr gut angenommen.

Weiter gibt es noch einen Consulting-Service, wo wirklich sich Forschende mit einem ganz konkreten Problem bei uns melden können und sagen können, okay, hier, ich entwickle eine dieser Forschungssoftware und ich habe jetzt folgendes Problem.

Und da kommt dann alles von, was wäre das richtige Tooling, um irgendwie das und das Problem zu lösen, wie kann ich das und das testen, wie kann ich überhaupt meine Software vielleicht besser machen von der Qualität bis hin zu, ich möchte das Ganze veröffentlichen, was muss ich denn dabei betrachten, was sind denn gute Open-Source-Lizenzen etc.

Also auch da können sich Leute einfach bei uns melden und da können wir uns auch nicht beschweren, dass es uns langweilig ist.

Tooling?

Achso, Tooling, was heißt Tooling?

Man fällt immer so in sein Sprech.

Ja, macht ja nichts, dazu bin ich ja da.

Also da geht es darum, man hat halt ein konkretes Problem und möchte es jetzt lösen.

Es gibt natürlich ganz viele verschiedene Wege und irgendwelche, ja im Endeffekt wieder Werkzeuge, die einen dabei unterstützen, jetzt aber nicht Werkzeuge wie so große, wie so ein Office oder so was einen unterstützt in der Art, sondern eher irgendwelche speziellen Implementierungen, also im Bereich Machine Learning zum Beispiel gibt es eine ganze Vielzahl von verschiedenen Ansätzen oder eben auch von anderen Firmen entwickelte, ja, Werkzeuge, nennt man dann oft Frameworks oder so was, die einen dabei unterstützen, die man halt verwenden kann und oft ist die Frage, okay, ich möchte das und das erreichen, das und das ist mein Anwendungsfall, was wäre denn jetzt eine gute

Wahl?

Und ja, da versuchen wir dann zu verstehen, worum geht es eigentlich und was ist technisch die sinnvollste Lösung, aber auch, was ist zum Beispiel üblich in der Domäne, in der die Wissenschaftler unterwegs sind.

Also da haben wir auch eine ganze Brandbreite und man muss natürlich etwas wählen, was dann auch in dem Bereich, in dem die Leute aktiv ist, sozusagen der ein bisschen dem Standard entspricht, damit halt auch andere damit arbeiten können oder verstehen und das halt einfach gut zusammenpasst.

Das ist nämlich in Forschungsbereichen gar nicht so unüblich, dass in den verschiedenen Forschungsbereichen es eben bestimmte, ja, so quasi Standards gibt, also bestimmte Tools, die alle verwenden.

Das kennt man ja vielleicht auch so ein bisschen.

Kann ich vielleicht noch ein Wort zu den Nutzungszahlen sagen, weil da gibt es noch einen oder zu einem angenommen werden durch Helmholtz.

Da gibt es, da hatten wir einen sehr, sehr, sehr dynamischen Effekt.

Also wir hatten eine Zeit lang den Effekt, dass uns die Leute sozusagen die Tür eingerannt haben, also dass wir mehr, mehr Zuwachs hatten, als wir eigentlich so in dem Aufbauzustand, den wir teilweise hatten, so erwartet hatten vielleicht sogar.

Das betraf zum einen, was Carina schon sagte, die teilweise sehr gut ausgebuchten Education Services, wo man dann noch Zusatztermine und so weiter einschieben muss.

Zum anderen aber auch die Cloud Services, wo wir aktuell ungefähr bei knapp 5000, glaube ich, Nutzern sind aus Helmholtz, die die Services benutzen.

Und das sehr stark steigend ist, obwohl wir noch gar nicht das Final schon, also schon finalisiert haben.

Es fehlen noch ein paar Organisationsframeworks dazu, um das schon als produktiv zu bezeichnen.

Organisationsframework.

Ja, das versuche ich dann gleich zu erklären.

Okay.

Das heißt, wir sind tatsächlich, was die Cloud Services geht, immer noch offiziell in einem Status, den man offiziell Pilotstatus nennen würde, weil da sind eben verschiedene Dinge noch nicht ganz geklärt.

Beispielsweise der berühmte Helmholtz Cloud Rahmenvertrag oder Rahmenvertragswerk, den man dafür noch braucht, der muss da noch finalisiert werden.

Das sind einfach ein paar rechtliche Dinge, die noch geklärt werden müssen, damit beispielsweise ein Zentrum sagen kann, okay, ich schaffe jetzt diesen Service, den ich jetzt bisher lokal bei mir hatte, ab und schicke meine Nutzer komplett auf den Service, den irgendein anderes Zentrum anbietet.

Und das würde zwar technisch heute schon gehen, aber dazu fehlt eben noch ein bisschen der rechtliche Rahmenhintergrund dafür.

Und das können wir bis jetzt, wird praktisch gemacht für Forschungsgruppen.

Also das heißt, es machen einzelne Forschungsbereiche oder Forschungsgruppen mit ihren jeweiligen Mitarbeiterinnen, aber wir können das noch nicht für ganze Zentren machen.

Das heißt, da fehlt noch ein bisschen ein paar organisatorische Bausteine.

Trotzdem haben wir wirklich eine hohe Nutzerzahl und sind jetzt eben gerade dabei, das so umzubauen, dass wir in den Produktivbereich gehen und dann noch tatsächlich ganz Helmholtz ansprechen wollen.

Und dann insbesondere auch die Zentren, die bis jetzt vielleicht ein bisschen unterrepräsentiert sind.

Da gibt es auch ein paar Zentren, die stärker bei uns repräsentiert sind als andere.

Es gibt vor allem sieben, die gar nicht dabei sind.

Ja gut, das ist einmal tatsächlich auch so top down festgelegt worden.

Ja, es sollten für HIFIS eben eine bestimmte Anzahl von Zentren dabei sein.

Für die anderen Inkubator-Plattformen, das haben wir jetzt noch gar nicht erwähnt.

Es gibt ja von HIFIS noch Schwesterplattformen.

Gibt es dann weder eine andere Konstellation von Zentren, die beteiligt sind.

Das ist erst einmal soweit so gut, weil es heißt ja, diese elf sollen das organisieren für alle.

Diese elf, die dabei sind, sind ja die aktiven Partner, aber mitmachen sollen am Ende schon alle und partizipieren auch alle.

Trotzdem gibt es noch eine Schiefelage, was die Gleichverteilung der Nutzungszahlen angeht.

Das merken wir schon, dass Zentren, die mit HIFIS nicht ganz so eng verbandelt sind, auch ein bisschen unterrepräsentiert sind.

Und da müssen wir tatsächlich auch noch Werbetrommel rühren und auch die Vorteile noch stärker vermitteln vielleicht, als wir das bisher tun konnten.

Und da sind wir jetzt auf allen Kanälen noch unterwegs dafür.

[HT] Karina, kannst du das mit dem Tooling, also mit den Tools und den Frameworks, kannst du das mal konkret machen?

Forscher X arbeitet an Projekt Y und braucht dazu Z?

[KZ] Ja, genau.

[HT] Hast du da was aus dem Alltag?

[KZ] Ich kümmer mich bei uns vor allem um die Open-Source-Sachen momentan.

Da könnte ich dir ganz viel erzählen.

Zu den Toolings muss ich jetzt gerade überlegen, aber eigentlich ist es wirklich so klassisch, dass jemand kommt und sagt, wir wollen eben Machine Learning anwenden für, ja, ein konkreter Use Case fällt mir jetzt gerade nicht ein, müsste ich mal nochmal nachgucken.

Aber dann kommt halt die Frage wirklich, welches ist denn das passende Framework, was ich benutzen kann?

Oder vielleicht ein anderes Beispiel, da ging es ums Testen.

Also, wie kann ich meinen Code testen?

Und ich habe diese und jene Programmiersprache.

Und welches ist denn jetzt das richtige Test-Framework?

Weil da gibt es einfach eine ganze Vielzahl.

Und dann kommt es natürlich darauf an, welche Art von Tests möchte man schreiben, in welchem Umfang?

Und wie gesagt, auch eventuell gucken, welche Tools sind einfach am weitesten verbreitet und sind auch einfach am schnellsten zu erlernen für den Use Case.

Also, muss man immer ein bisschen schauen, gibt es irgendwas ganz Spezielles, was abgedeckt sein muss oder geht es eher ums Klassische?

Und in der Regel führen wir halt mit den Leuten ein Gespräch und schauen uns dann gegebenenfalls auch nochmal selbst die Tools ein bisschen an.

Wir sind ja auch nicht immer Experten in allem und geben dann halt Empfehlungen und unterstützen gegebenenfalls auch dann am Anfang so und zeigen mal, okay, hier, so schreibt man den Test, so sieht das aus, so bringt ihr das in eure Automatisierung, dass immer, wenn ihr Code irgendwie speichert so ungefähr und das auf das Codespeichersystem hochladet oder irgendwas, dass es dann halt ausgeführt wird, automatisiert.

Also solche Sachen, da unterstützen wir wirklich und halten sozusagen Händchen.

[Siebert] Ich glaube, ich habe eine noch viel zu naive Vorstellung davon, wie Software in der Forschung benutzt wird.

Also für mich ist das so, also ich habe ein Problem.

Problem heißt, ich muss Worte zu Papier bringen, dann starte ich eine



Textverarbeitung, schreibe da die Worte rein und dann sind die sozusagen zu Papier gebracht.

So geht das in der Forschung gar nicht.

Da hast du für jedes einzelne Projekt auch eine einzelne Software, die hinterher die Daten verarbeitet, oder?

[Kehl] Das ist unterschiedlich.

Also generell ist es natürlich, dass man erstmal ein Forschungsprojekt hat, wo man sagt, ich möchte das und das erreichen und dann muss man halt überlegen, wie kann ich das erreichen?

Und es kann bedeuten, dass es irgendeine Software gibt vielleicht schon, die dann einfach nur benutzt wird.

Das ist dann oft in der Regel, werden dann irgendwelche Skripte geschrieben, um sozusagen verschiedene existente Software-Teile in einer Reihenfolge aufzuführen oder eben bestimmte Daten da rein zu stecken und so weiter.

Das heißt da... [Siebert] Das ist für so Doofe wie mich, das ist wie wenn ich dann so eine kleine Funktion in eine Zelle einer Tabellenkalkulation rein... [Kehl] Ja, so ein bisschen.

Also man sagt halt, nimm diese Daten und dann steckt das erst in dieses Programm und dann in dieses Programm und dann in dieses Programm, so ungefähr.

Und dann malen auch die Ergebnisse schön.

Das ist so ein bisschen der Klassiker.

Das heißt, da entwickelt man auch eigenen Code, aber es geht halt eher darum,

was Kleines zu schreiben, wo man sagt, okay, ich möchte diese Daten haben und jenes machen.

Das ist halt eher so eine Anleitung zu schreiben, sozusagen, was der Computer jetzt bitte mit den Sachen machen soll.

Es gibt aber auch Projekte, wo dann gesagt wird, okay, hier muss wirklich eine ganz neue Software für geschrieben werden.

Es gibt nichts, was diese Dinge schon kann, was wir machen möchten.

Und dann wird entwickelt.

Und da gibt es natürlich eine ganz große Bandbreite.

Das kann eben von kleineren Projekten sein, die wirklich übersichtlich sind, was sie machen.

Irgendein kleines Programm, wo man ein paar Sensoren auslädt und das Ganze ausliest und das Ganze noch in eine Datenbank packt und irgendwie schön darstellt und durchsucht, was man macht oder so.

Aber es kann halt auch wirklich ein großes Projekt werden, ein richtiges Softwareprojekt, sozusagen.

Und das sind dann oft... Also Beispiele dafür sind Simulationscodes, also wo zum Beispiel Luftströmung an Flugzeugen simuliert wird wirklich.

Da gibt es dann sehr große Programme, wo auch über eine lange, lange Zeit entwickelt wird und auch über mehr als ein Forschungsprojekt.

Das heißt, da guckt man dann immer nach einer Anschlussfinanzierung etc., um dann eben die nächsten Teile auch implementieren zu können.

Das heißt, so eine Software kann im Rahmen eines Forschungsprojekts leben.

Die kann aber auch darüber hinaus bestehen.

Das kommt dann immer stark darauf an.

Und gerade wenn der Wunsch aber besteht, dass es darüber hinaus bestehen soll, das ist der Moment, wo die dann oft zu uns kommen und sagen, okay, wir möchten es zum Beispiel open sourcen oder eben damit andere das auch benutzen können oder es für die nächsten Projekte zur Verfügung steht.

Wenn es dann so eine Software gibt, die beispielsweise open sourcen wollen für andere Projekte, für andere Zentren, musst du die dann nicht auch irgendwie verschlagworten, sodass die anderen überhaupt wissen, dass es das gibt?

Genau, das ist auch ein Problem.

Da spricht man von der Sichtbarkeitmachung von Forschungssoftware, also dass man die überhaupt auffindbar macht.

Natürlich gibt es irgendwelche Software-Repositories im Web, also die klassisch bekanntesten, jetzt nenne ich doch mal, sind GitHub.com oder GitLab.com, wo man suchen kann nach Software.

Und natürlich sind auch Forschungszentren auf diesen Plattformen unterwegs.

Aber das ist natürlich funktioniert für Sachen, die man jetzt open source macht.

Aber wir haben natürlich auch Forschungssoftware, die vielleicht nicht öffentlich ist, die wir aber genauso bekannt machen wollen.

Und deswegen entwickeln wir unter anderem in HIFIS gerade ganz aktuell ein Research Software Directory, also wirklich sozusagen ein Katalog für Forschungssoftware.

Und den starten wir gerade damit zu befüllen.

Es wird in den nächsten, weiß ich gar nicht, Wochen, hoffe ich, doch noch online gehen mit sogenannten Leuchtturmprojekten.

Das heißt, haben wir jetzt auch alle Zentren befragt.

Nennt uns doch mal die Projekte, die eine besondere Strahlkraft haben bei euch und gebt uns da die wesentlichen Informationen, damit wir die zentral aufbereiten und bereitstellen können.

Und das ist ein Wunsch, der in HIFIS schon lange hegt und jetzt umgesetzt wird.

Und etwas, was ich eben auch sehr, sehr wichtig finde, denn es ist wichtig zu zeigen, dass Forschung eben auch aus Software entbesteht und vor allem diese Software auch wiederum für andere verfügbar machen, damit, wie vorhin ich schon erwähnt habe, andere darauf auch ihre Forschungs wieder drauf aufbauen können und das weiterführen können.

Uwe, du hast eben noch so einen Begriff gesagt, Inkubatorplattformen.

Ja, sehr schön.

Ist auch nicht um ein bisschen gewechselt.

Tolles Wort.

Was ist das?

Ja, also das kommt aus einem größeren Prozess, den Helmholtz 2016 angestoßen hat.

Da wird jetzt übrigens gerade demnächst dann der Fünfjahrestag groß gefeiert.

Also der Prozess sozusagen, der übergreifende, die übergreifende Überschrift darüber ist, dass man eben tatsächlich die Fähigkeiten von Helmholtz bündeln möchte, um insgesamt einfach Synergieeffekte auszunutzen und auch eine größere Durchschlagskraft zu entwickeln und auch eine größere domänenübergreifende Zusammenarbeit zu ermöglichen und so weiter.

Und das wird also auf verschiedenen Ebenen gemacht und deswegen gibt es innerhalb dieses Inkubators fünf Plattformen, die jeweils unterschiedliche Aufgabenschwerpunkte haben.

Eine davon ist eben Hyfus.

Das ist die Plattform, die im Inkubatorprozess dafür zuständig ist, diese IT-Fähigkeiten zu bündeln.

Das sind wir halt.

Dann gibt es aber noch andere Plattformen wie die Imaging-Plattform, die dafür zuständig ist, große Bildverarbeitungsketten zu vereinheitlichen und auch zusammenzuführen, die in Helmholtz stattfinden.

Also Beispiele sind, dass man eben für Tomografie-Systeme große Mengen an Bilddaten erzeugt oder in allen möglichen anderen bildgebenden Verfahren letzten Endes Bilddaten in großen Mengen einmal erzeugt und dann wieder in großen Mengen auch weiterverarbeitet und das Ganze auch domänenübergreifend genutzt werden kann.

Dann gibt es noch die Helmholtz AI-Plattform, ehemals Heiko, die letzten Endes sich für künstliche Intelligenzverfahren interessiert, wie man die domänenübergreifend nutzen kann.

Und dann gibt es noch die Helmholtz Metadata Collaboration, HMC, die das so ein bisschen in die Richtung geht, was Carina gerade schon angesprochen hat,

nämlich die Auffindbarkeit in dem Fall weniger von Services, sondern mehr von Daten zu erleichtern, dass man also Datensätze finden kann, anhand von sehr vielen Hintergrundinformationen, die man eben irgendwo hinterlegt hat.

Und dann die fünfte Plattform, dann bin ich auch schon durch, ist die HIDA.

Das ist also eine Plattform, die dafür zuständig ist, die Lehre und die Nachwuchsausbildung zentral zu koordinieren und da auch beste Möglichkeiten zu schaffen.

[Siebert] Metadata Collaboration finde ich ganz witzig.

Ich war an sowas mal beteiligt in einem ganz anderen Kontext und das hat sieben Jahre lang nicht funktioniert.

Funktioniert das bei euch?

[Siebert] Die sind genauso neu wie wir.

[Siebert] Also die Metadata Collaboration sogar noch jünger.

Also funktionieren ist jetzt ein Punkt, wo man sagen würde, jetzt gibt es ja schon fertige Produkte.

Das gibt es tatsächlich nicht.

Das ist halt auch viel zu frisch dafür.

[Herr Kopp] Also das Nicht-Funktionieren in meinem Zusammenhang war halt wirklich, dass man es in sieben Jahren nicht geschafft hat, sich auf einen einheitlichen Metadatensatz zu einigen.

[Siebert] Also da kann ich vielleicht was zu sagen, weil ich bei HAMP 10 Stück weit auch involviert bin.

Es ist allen schon klar, es wird nicht den einen Standard-Metadatensatz für alle geben.

Es gibt natürlich klar ein paar Grunddaten, die sehr elementar sind, die alle gemeint haben.

Aber hier geht es wirklich darum, pro Forschungsdomäne sich zu einigen auf Standards oder zumindestens auf wenige und dann da Möglichkeiten zu finden.

Es nennt sich oft Crosswalks, wie man von dem einen zum anderen kommt.

Das heißt, dass man das eine in das andere überführen kann.

Und Ziel ist also erst mal zu gucken, was sind existierende Quasi-Standards, sich dann gegebenenfalls zu einigen und dann eben zu schauen, okay, welche Grundaspekte gibt es, die für welche Domäne wichtig sind.

Und vielleicht finden wir wirklich so einen ganz kleinen Kern, aber das wird wirklich auf sehr, sehr wenig Attributen sich beschränken.

Also irgendwie so Name und Datum oder so.

Aber Ziel ist es eher, wirklich eine Sammlung zu haben und zu sagen, okay, je nach Domäne gibt es bestimmte Sachen, die wir empfehlen.

Und es gibt vielleicht auch kleine Tools drum rum, die helfen, genau diese Art von Metadaten aufzuzeichnen oder ineinander zu überführen, damit man eben dann am Schluss auch wieder irgendwie eine sinnvolle Suchung etablieren kann und solche Sachen.

[HT] Ist das denn nachhaltiger Umgang, von dem Uwe vorhin gesprochen hat?

[SM] Es geht um den nachhaltigen Umgang von Forschungsdaten in dem

Bereich auch.

Denn wenn ich die nicht wieder auffinden kann, dann habe ich sie erstellt und dann sind sie eigentlich – sie sind zwar noch da, aber eigentlich sind sie weg.

Weil wenn ich sie nicht finden kann, ist das ein bisschen wie wenn man etwas in den Keller bringt.

Man weiß, es ist noch irgendwo im Keller, aber eigentlich ist es weg.

Und genauso ist es hier auch.

Und deswegen ist es natürlich schon im Sinne der Nachhaltigkeit, wenn man eben diese Daten auffindbar und auch vor allem für weitere Leute nutzbar macht.

Das bedeutet, dass man ja auch wissen muss, was ist mit denen schon passiert?

Was finde ich da eigentlich drin?

Was ist das für ein Format?

Wie muss ich mit denen arbeiten?

Und das sind halt alles Sachen, die sind natürlich sehr aufwendig, das Ganze ordentlich zu dokumentieren und sich da jetzt auch noch zu einigen.

Und deswegen ist das ein sehr langwieriger Prozess, der halt gerade so seinen Start hat.

[HT] Vor allen Dingen, du verhinderst ja dann am Ende auch, dass ein und dieselbe Arbeit immer und immer wieder gemacht wird.

Und zwar in jedem Zentrum mindestens einmal und dann wahrscheinlich auch noch in irgendwelchen Forschungsgruppen jeweils noch einmal.



[SM] Genau.

Und es ist so, dass wir jetzt auch nicht versuchen, nur innerhalb von HMC oder auch nur innerhalb von HIFIS oder so uns für Dinge zu entscheiden oder Dinge durchzusetzen, sondern in all diesen fünf Inkubatoren, also zum einen sind die selbst miteinander vernetzt und tauschen sich aus, aber diese sind auch vernetzt mit weiteren bereits bestehenden Zusammenschlüssen, Gruppierungen, Gesellschaften.

Also es gibt ja auch alles Mögliche schon.

Also es ist jetzt auch nicht alles komplett neu.

Also es gibt irgendwie NFDI, die schon einige der Themen aufgreifen.

Das ist die, jetzt muss ich das wieder richtig zusammenkriegen.

[HT] Nationale Forschungsdateninfrastruktur.

[SM] Uwe hat es gerade.

[HT] Nationale Forschungsdateninfrastruktur.

[SM] Genau.

Die beschäftigen sich da mit einigen Themen und da wird natürlich ein Austausch gesucht.

Und so gibt es auch in anderen Bereichen, wo natürlich ein Austausch gesucht wird, jetzt nicht nur bei HMC, jetzt kommen wir ja ein bisschen weg von unserem HIFIS-Thema eigentlich, sondern auch bei anderen Bestrebungen.

Wie HIFIS wird natürlich rechts und links geguckt.

Was machen andere?

Wie läuft das bei anderen?

So gibt es zum Beispiel aus den Niederlanden das eScience Center, die auch sich zum Beispiel mit einem Research Software Directory beschäftigen und da was Tolles entwickelt haben.

Das schaut man sich natürlich an und ist mit den Leuten im Austausch.

Genauso haben die auch eine tolle Trainingsstruktur, weswegen wir uns aus dem Education-Bereich mit denen ein bisschen kurz geschlossen haben, um sich auszutauschen, was hat bei euch Erfolg versprochen, was nicht.

Und genauso gibt es auch in Deutschland zum Beispiel die DERSI.

Das ist der Deutsche Verein für Research Software Engineers, also wirklich Software Engineer, die in der Forschung tätig sind und da unterstützend tätig sind.

Und da tauscht man sich natürlich auch dann mit Leuten aus, die in allen möglichen Zentren auch außerhalb von Helmond sind.

Und das heißt, wir haben da über unsere Finger hin und wollen gar keine Insellösungen unbedingt schaffen, sondern wirklich in den Austausch treten mit, was haben eigentlich andere gemacht.

Und wir haben wirklich den Luxus hier eine Finanzierung auch zu haben, um solche Sachen dann auch in die Praxis umzusetzen, wo sich viele einig sind, dass das der richtige Weg ist.

Und gleichzeitig versuche zumindest ich mit meiner Gruppe das nicht nur umzusetzen, sondern auch direkt natürlich, wir sind ja auch Forscher, Forschung

ein bisschen zu begleiten.

Das heißt, zu verstehen, wie entwickeln zum Beispiel Entwickler wirklich, was benutzen sie, wo brauchen sie Unterstützung, wie kann man ihnen helfen?

Also für uns ist gleichzeitig das, was wir aktiv machen, auch wiederum eine Datenquelle, die wir anzapfen, um allerdings wieder zu verbessern, was wir tun, also besser zu verstehen, was sind denn die Bedürfnisse?

Das heißt, da haben wir dann auch ein bisschen die ganz klassische Informatikforschung so ein bisschen verlassen und sind eher im fast sozialwissenschaftlichen Bereich, dass wir Leute halt auch mal interviewen oder Umfragen machen, um rauszufinden, ja, was sind die Bedürfnisse, was wird angenommen auch gut und was müssten wir vielleicht anpassen?

Wenn so viele Entitäten miteinander vernetzt sind, das erhöht ja dann auch irgendwann das Rauschen.

Ist der langfristige Plan, das dann letztendlich wieder zu vereinheitlichen oder zu vereinfachen?

Also eine große Struktur drüber zu setzen sozusagen?

Nein, also ich glaube nicht, dass wir so monolithisch werden wollen und auch gar nicht sollten.

Das sind wir auch jetzt ja nicht, ja.

Also es gibt ja zum Beispiel, hatten wir schon vorhin erwähnt, nicht die HIFIS-Institutionen.

Wir sind auch kein zentraler IT-Blob, der irgendwie Helmholtz, der IT für Helmholtz zur Verfügung stellt, sondern wir sind ja nur eine Kollaboration von dezentralen Anbietern.

Und was wir aber machen und was Carina ja schon schön angesprochen hat, ist wirklich den Vernetzungsgedanken auch über Helmholtz hinaus zu tragen.

Also wir wollen jetzt nicht eine Insellösung durch die andere Insellösung ersetzen, sondern wirklich die Vernetzung nach außen ermöglichen.

Auch technisch zum Beispiel sind wir mit den ganzen europäischen Infrastrukturen vernetzt und tatsächlich kann sich auch jeder aus allen möglichen anderen europäischen Kontext bei unseren Cloud-Services einloggen, beispielsweise.

Das funktioniert tatsächlich auch schon über die AI-Infrastruktur.

Und das ist also etwas, wo wir schon Grenzen sehen und auch gewissermaßen behalten wollen, dass nicht alles mit allem vermischt wird.

Aber diese Vernetzung, die Kompatibilität einfach auch immer gewährleistet sein soll.

Aber es wird sicherlich nicht das Ziel sein und auch nicht wünschenswert, da eine überbordende monolithische Struktur darüber zu bauen.

Du hattest, Uwe, vorhin schon eine Weile her von Computing Power gesprochen und dass ich mich über eure, also über HiFis auch an Rechenzeit beteiligen kann bzw. daran komme.

Skaliert das eigentlich?

Also bin ich in der Lage zu sagen, ich brauche jetzt 100 Computer, die gleichzeitig laufen oder 1000?

Im Prinzip kann man das machen, klar.

Es ist natürlich so, wenn das jetzt jeder gleichzeitig machen würde, dann käme man natürlich relativ schnell in Limitierungen.

Ja, Karina könnte da was verschreiben, ein Tool, das das dann wieder verwaltet, oder?

Ja, dafür gibt es erstmal einen gewissen, deswegen sind die Antragsprozesse ja auch gedacht, dass man ein kleines bisschen erklärt, wozu ist es eigentlich gut.

Also immer da, wenn Ressourcen teuer werden, also tatsächlich einfach Geld dafür irgendwann mal bezahlt worden sein muss, dann gibt es dann auch die Notwendigkeit, ein bisschen zu begründen, wofür das eigentlich sinnvoll ist und im Zweifel dann auch Beratung einzuhören, wie Karina schon sagte, um dann Better Practices zu vermitteln, also wie man beispielsweise die Ressourcen auch effizienter einsetzen kann.

Karina, du hast von Softwarequalität gesprochen.

Wie misst man die?

Ja, das ist relativ eine interessantere Frage, eine gute Frage.

Danke.

Das ist nämlich gar nicht so einfach.

Es gibt natürlich Tools, die das versuchen, also die analysieren den Code und schauen eben, wo hat man besonders komplexe Strukturen verbraucht, oder hält man sich an Standards von Programmiersprachen, also da kann man wirklich gucken, oder wo wird viel, also ganz klassisches Metric, also ein Kriterium ist zum Beispiel, wie viel des Codes wird wirklich getestet.

Das nennt sich dann Testabdeckung, wie hoch ist die und solche verschiedenen Sachen.

Also da gibt es verschiedene Dinge, die man sich anschauen kann und wofür es auch schon Tools gibt, um das automatisiert zum Beispiel ablaufen zu lassen.

Aber die große Frage ist oft für die Leute, dass sie merken, ich habe ein Stück Software, an dem schreibe ich, ich entwickle, aber ich merke immer wieder, ich ändere was und dann funktioniert irgendwas nicht.

Oder vielleicht kriege ich das auch erst Tage später mit und dann ist für sie die Frage, wie kann ich das jetzt verbessern.

Und das ist natürlich so ein klassischer Fall für die Testabdeckung.

Aber es gibt auch andere Aspekte wie die Lesbarkeit, das heißt, ich habe zum Beispiel ein Stück Software und mein Kollege möchte das mit mal nutzen und dann sage ich, ja klar, hier, es liegt da und da, kannst du die runterladen, kannst du benutzen.

Und der steht dann da vor und sagt, ich verstehe nicht, was es tut, wofür ist es eigentlich gut und auch das geht in die Softwarequalität.

Also, dass man halt sagt, es ist alles ordentlich dokumentiert, es ist alles ordentlich benannt, aber es ist auch schön strukturiert, es ist auch lesbar, es ist ein bisschen wie bei so einem Text, da kann man ja auch mit Formatierung und die Lesbarkeit stark erhöhen und das Gleiche gilt für Quellcode.

Das heißt, das sind all diese verschiedenen Aspekte, die zusammenkommen, so zu einer ganzen Vielzahl von Metriken.

Und ja, es hat sich bis heute noch nicht so richtig durchgesetzt, dass man sagt, sozusagen, es gibt jetzt so eine Skala und du hast jetzt irgendwie eine 5 oder 7 oder sowas, sondern es geht darum, einfach zu schauen, wie kann ich meinen Code so gut machen, dass er für das, wofür ich ihn brauche, gut ist.

Das bedeutet, in meinem Verständnis nämlich, wenn ich jetzt zum Beispiel ein kleines, wie ich vorhin erwähnt hatte, so ein kleines Skript habe, um irgendwie eine Analyse von ein paar Daten zu machen und mir ein paar schöne Diagramme auszugeben, dann sollte jetzt immer noch jeder verstehen, was habe ich mit den Daten gemacht und wie sind daraus jetzt diese Diagramme entstanden und was sehe ich in diesen Diagrammen und das sollte halt lesbar sein, für jemanden nachvollziehbar sein.

Aber den Anspruch, den ich habe, wie gut das jetzt dokumentiert, notiert und aufgeräumt ist und wie schön das gemacht ist, ist natürlich ein ganz anderer, als wenn ich sage, ich habe hier einen großen Simulationscode, mit dem wir irgendwie jetzt simulieren, wie Flugzeuge fliegen und damit neue Flugzeugentwürfe erstellen oder solche Sachen.

Dann muss der Code natürlich viel stabiler, robuster haben und eine höhere Qualität haben.

Das heißt, Codequalität ist immer ein Stück weit subjektiv zu dem Use Case, auch wenn es natürlich schon generell objektiv bewertbare Attribute gibt, wie eben Testabdeckung oder so was.

Das Testabdeckung musst du mir auch noch mal erklären.

Ich dachte, der Code sei sein eigener Test, weil entweder funktioniert es oder nicht.

Also ich bin in meinem Leben über HTML nicht rausgekommen, nur um den Realitätsabstand zwischen uns beiden.

Also wenn man seine Software manuell testet, das ist das, was du so ein bisschen vielleicht kennst.

Wenn du bei dem LDT-NASCO kommst, aber dann auch, dann hast du deine Webseite gestartet und geguckt, sieht die so aus, wie du dir das vorgestellt hast.

Das ist der manuelle Test.

Ist das richtige Hintergrundbild da?

Steht da der Text, den ich haben möchte und ist die richtige Stelle irgendwie eine Überschrift?

Und wenn ich jetzt automatisiert teste, dann machst du das nicht du, sondern der Computer macht das für dich.

Das heißt, ich muss vorher schon eine Software geschrieben haben, die weiß, wie die Webseite aussehen soll, um zu beurteilen, ob der Code, der die Webseite erzeugt, ordentlich gearbeitet hat.

So ungefähr.

Aber das ist doch verrückt.

Und wenn wir jetzt mal weg von der Webseite gehen und was nehmen, was uns einfacher sich vorstellt, dann gibt es zum Beispiel eine Funktion.

Du möchtest eine Funktion schreiben, die irgendwie, naja, nehmen wir jetzt mal was ganz ganz Simples.

Es geht wirklich um so was wie die Quadratwurzel ziehen.

So, und dann schreibst du eine Funktion, die soll die Quadratwurzel ziehen.

Das schreibst du dann irgendwie auf und das musst du jetzt testen, um zu wissen, ob da auch das richtige Ergebnis gibt.

Jetzt kannst du natürlich die Software starten und einfach mal eingeben 4 und wenn da 2 rauskommt, ist gut.



Aber das möchtest du halt nicht immer manuell machen, sondern du möchtest eigentlich sicherstellen, dass das, egal wenn du eine andere Stelle änderst, das weiter funktioniert.

Das heißt, du möchtest das automatisieren.

Das heißt, du schreibst einen kleinen Test, in dem du sagst, wenn ich diese Funktion aufrufe mit dem Wert 4, dann soll da bitte 2 rauskommen.

Oder wenn ich es mit dem Wert 9 aufrufe, dann soll da 3 rausgehen.

So, jetzt kannst du aber auch so was machen, dann, was soll passieren?

Also, das ist ja erstmal, wo man sagt, okay, klar, das gibt dann die richtigen Ergebnisse, super.

Aber beim Testen geht man dann noch einen Schritt weiter und sagt, haha, jetzt übergebe ich mal eine -1, was passiert denn dann?

Und die Wurzel von -1 ist jetzt nicht so richtig möglich.

Dann sollte natürlich irgendein Verhalten kommen und das sollte vielleicht nicht sein, dass die Software abstürzt und irgendwie weg ist, sondern dann sollte irgendwie eine Meldung kommen, geht nicht.

Oder was passiert, wenn ich da jetzt ein Wort eingebe in diese Box?

So, und das heißt, man überlegt sich beim Test, was ist wirklich Verhalten, was passieren könnte?

Entweder weil ein anderer Computer das irgendwie falsch eingibt oder weil ein Mensch irgendwas Falsches eingibt und damit sozusagen sich dann auch überlegt, wie sollte das Ganze dann sich verhalten?

Und da gibt es auch eine Entwicklungsmethodik, die heißt Test Driven Development, wo es wirklich darum geht, dass man sich erst überlegt, wie soll unter bestimmten Situationen sich die Software verhalten?

Dafür schreibt man einen sogenannten Test und dann schreibt man die Software so, dass sie das Ganze erfüllt.

Ganz wichtig in dem Zusammenhang ist ja dann auch, wenn man später diese Wurzelfunktion verwenden will und inzwischen vergessen hat, was man da eigentlich damals vielleicht vor zwei Jahren geschrieben hat oder dass jemand anders das auch verwenden will, diese Funktion und nicht genau weiß, wie diese Funktion funktioniert und die dann verwenden will und dann plötzlich irgendwelche invaliden Zwischenergebnisse auch erzeugt werden.

Wenn aber der Test das vorher schon wegfängt, weil der schon mal sauber geschrieben wurde, dann können solche Fehlerfortpflanzungen dann auch unwahrscheinlicher, sagen wir mal, auftreten, weil eben solche Abhängigkeiten dann unterbrochen werden.

Was ist denn dieser Test?

Ist der Test nicht auch eine Software?

Er ist auch ein Stück Code, ja.

Aber der will doch dann auch wieder getestet werden.

Im Endeffekt sind auch Tests anfällig für Fehler, aber eigentlich ist die Idee, dass Tests immer so einfach aufgebaut sind, dass dort die Wahrscheinlichkeit für Fehler deutlich niedriger ist.

Und es geht halt eigentlich immer oder viel um eben, dass man bestimmtes, also dass man sich manuell überlegt, also die Idee ist, dass man wirklich sich von

hinsetzt jetzt sozusagen und weiß, das und das ist der Wert, der rauskommt und damit sozusagen die Kontrolle macht, immer, dass man die über einen zweiten Weg zu diesem Ergebnis gekommen ist und dann diesen Test macht.

Aber im Endeffekt, ja, es gibt Fälle, wo eben Tests die gleichen Fehler reproduziert haben, also einfach genau der gleiche Fehler vorkam, der dann auch sozusagen eine Implementierung ist oder das auch einfach bereichernd nicht getestet wurden und dann halt fehlen.

Das heißt, es gibt natürlich keine hundertprozentige Sicherheit.

Aber die Idee ist, wenn du ein großes Stück Software hast und anstelle A was änderst, du immer noch danach sicher sein kannst, dass anstelle B diese Änderung keinen Einfluss hatte, weil die Tests von B immer noch erfolgreich durchlaufen.

Und das ist bei Software gar nicht so unüblich, dass du sozusagen links was änderst und das hat dann aber Auswirkungen auf irgendwas, was rechts passiert.

Und wenn die Tests aber weiter erfolgreich durchlaufen, kannst du dir sicher sein, dass die Software sich weiterhin richtig verhält oder so verhält, wie du es erwartest, zumindest in den Aspekten.

Während wenn du das nicht hast, dann stellst du diese Fehler vielleicht erst fest.

Also sagen wir mal, ich überlege gerade, ob ich ja oder der Anwender feststelle, das ist noch schlechter.

Na, wenn er es überhaupt feststellt, weil der sieht ja nur das Ergebnis.

Der wirft vorne Daten rein, hinten kommt ein Ergebnis raus und der muss sich ja eigentlich auch darauf verlassen können, dass das Ergebnis stimmt.

Das ist richtig, genau.

Also zum einen geht es darum, das zu erhöhen, dass man halt dieses Vertrauen hat, dass diese Software das Richtige macht, weil es eben ordentliches Testing geht.

Aber zum anderen geht es auch um die Sicherheit des Entwicklers, dass wenn er eben eine Stelle anfasst, das keine Auswirkungen auf die andere hat.

Das wäre jetzt zum Beispiel, wenn du die Wurzelberechnung wiederum auf andere Berechnungen basiert.

So, ja, sei es irgendwo auf eine Division oder so was und du veränderst, wie die Division funktioniert, dann könnte ja plötzlich sein, dass du die Division weitertestest und das alles richtig aussieht mit deinen Beispielen, aber es plötzlich irgendwelche Probleme macht bei der Wurzelbildung, weil du irgendwas nicht berücksichtigt hast.

Und dann ist es halt wichtig, dass du diese Tests hast, weil einfach zeigt, okay, ich habe unterliegend irgendwas, was benutzt wird von dieser Funktion verändert, aber ich komme weiter zu den gleichen Ergebnissen.

Dass die richtig sind, das ist ein separates Problem.

Das muss man halt, wie gesagt, immer gut überprüfen.

Aber es geht halt gerade darum, eben diese Sicherheit zu haben.

Ich verändere was und alles verhält sich noch genauso, wie es vorher hatte.

Einen wichtigen Aspekt möchte ich vielleicht noch dazu sagen.

Viele dieser Softwareprojekte, gerade im Forschungsbereich, entstehen ja aus kleinen Projekten.

Man skriptet sich was, man codet sich was, um ein Problem zu lösen.

Dann wächst das, dann wird es komplexer, dann wächst es noch weiter, dann kommen andere Forscher dazu, dann teilt man das vielleicht und schwuppdiwupp hat man plötzlich aus einem kleinen Projekt ein großes Projekt gemacht, aber vielleicht noch nicht die Good Practices implementiert, die man eigentlich vielleicht von Anfang an hätte machen sollen.

Zum Beispiel eben automatisches Testen, vernünftige Versionierung und so weiter.

Und das ist der Punkt, wo wir dann eben auch versuchen, möglichst früh schon das Bewusstsein dafür zu wecken, dass das von Anfang an nützlich ist und nicht erst dann, wenn man schon ein großes, komplexes Projekt hat, was dann vielleicht schwer noch test- und debugbar ist.

[Siebert] Seid ihr denn dann auch in der Lage, nachträglich, ich sag mal, diese Projekte, diesen Code zu reparieren?

[Kleinwächtinge] Genau.

Also das ist das, was wir genau über dieses Consulting anbieten.

Das heißt, Leute kommen zu uns und sagen, okay, wir haben hier ein Stück Software, das ist einfach über die Jahre entstanden und gewachsen.

Das ist jetzt aber relativ wichtig für uns und wir merken immer mehr, dass genau das, was ich gerade so ein bisschen geschrieben habe, der Fall ist.

Wir verändern was, irgendwas geht kaputt, aber es ist total wichtig für uns, dass das funktioniert.

Und wir möchten das jetzt wirklich als zentralen Bestandteil behalten.

Und dann melden die sich bei uns und sagen, okay, wo fange ich eigentlich an?

Wie kann ich das denn jetzt so Stück für Stück machen?

Weil natürlich ist das erst mal diese Mammutaufgabe, vor der man steht.

Und wie bereits vorhin erwähnt, handelt es sich ja hier um Forschende, die natürlich Experten in ihrer Domäne sind, aber nicht unbedingt in der Softwareentwicklung.

Das heißt, da kommen wir dann mit unserem Software-Expertenwissen dazu und sagen, okay, und schauen uns auch den Code wirklich an, sagen, okay, erster Schritt wäre, wir räumen mal dies auf, wir tun mal jenes, wir schreiben erste Tests.

Zum Beispiel ganz klassisch ist so ein sogenannten Durchstoßtest zu machen, der einmal sagt, sozusagen, um sicherzustellen, dass diese Software grundsätzlich an einem Beispiel noch alles tut.

Also wirklich so ein klassisches Beispiel, wenn man jetzt irgendwelche Daten hat, die das eben bearbeiten soll, also dann nimmt man ein paar Beispieldaten und lässt das einmal durchlaufen, komplett das ganze Stück und hat dann am Schluss was, was rauskommt.

Und dann hat man erst mal so eine Basis, dass man sagen kann, okay, also wenn sich da jetzt was verändert, da müssen wir aufpassen, was wir verändert haben, aber so gibt es uns so eine Sicherheit, wir können jetzt anfangen, diesen Code aufzuräumen, umzustrukturieren und das Ganze halt besser zu machen nach diesen Standards und dann eben iterativ, also wirklich Schritt für Schritt, immer mehr diesen Code in eine gute Qualität zu bringen.

Und das bedeutet manchmal eben nicht, einen uralten Code unbedingt jetzt irgendwie dann nochmal für alles Test zu schreiben, sondern vor allem für Änderungen, für neue Sachen, weil man muss ja auch einfach was realisieren,

was zur Arbeitsweise der Forschenden passt und sich auch in ihren Alltag integrieren lässt.

Das heißt zu sagen, okay, ihr müsst jetzt irgendwie zwei Monate lang das Ganze hier aufräumen, ist vollkommen unrealistisch.

Aber ihr macht das dann auch nicht, also ihr habt jetzt auch nicht eine Putztruppe und sagt, hier, mach mal den Code sauber.

Nein, das können wir auch größtenteils nicht, denn das Fachwissen, was in diesem Code steckt, das braucht man für viele dieser Stellen.

Das heißt, man kann nicht einfach hingehen und sagen, okay, ich gucke mir jetzt diesen Code an und ich schreibe euch das jetzt mal in schön, sondern es geht eher darum, sich gemeinsam hinzusetzen, zu identifizieren, welche Probleme gibt es, welche kann man gut adressieren, welche kann man einfach adressieren, was sind jetzt Strategien, die wir entwickeln gemeinsam für die Zukunft, damit sozusagen die Situation über lange Zeit einfach dann verbessert werden kann.

Und dazu ist hoffentlich der erste Schritt, auch die Wissenschaftler erstmal eben zu schulen, um ihnen beizubringen, so und so macht man es richtig, diese Prozesse einzuüben und dann nachzuschauen, dass alles, was neu dazu kommt oder so, dass das mit der Zeit einfach alles verbessert wird, dass sozusagen jede Stelle, die man nochmal anfasst, man dann bei der Gelegenheit aufräumt.

Denn man muss auch sagen, wenn man ein Stück Code hat, was wirklich jetzt drei Jahre sehr gut funktioniert hat, dann muss man vielleicht nicht mehr alles hinterherrennen und sauber machen, weil man erstmal sich darauf verlassen kann, was drei Jahre für alle gut funktioniert hat, funktioniert jetzt auch erstmal weiter gut, aber es geht darum natürlich dann bei neuen Änderungen oder so, oder sobald man immer Probleme feststellt, dann an die Stelle zu gehen und zu sagen, okay, jetzt räume ich hier auf.

Das heißt, man renoviert sozusagen nicht das ganze Haus auf einmal, aber wenn

man jetzt merkt, oh, hier ist was kaputt gegangen, dann macht man halt nicht nur einen Kleber drüber, sondern fängt dann an, die Stelle wirklich ordentlich mal aufzuräumen, ordentlich zu renovieren und dann geht man weiter ans nächste Stück und so macht man das begleitend.

Und da sind wir dann auch wirklich längerfristiger bei Projekten dabei und unterstützen die.

Also viel ist es, dass wir am Anfang eben so einen großen Start machen, ein paar Empfehlungen mitgeben, sie auf Schulungen mitnehmen und dann eben sagen, okay, sobald ihr wieder einen Punkt habt, meldet euch wieder und dann sitzen wir daneben und helfen euch.

[Siebert] Das klingt aber trotzdem so, als wäre das viel zu viel Aufwand für die zwei Jahre, die mir da irgendwelche Forschungsgelder bewilligt worden sind.

Schaffen die Forscherinnen und Forscher das in der Zeit, die ihnen zur Verfügung steht, hinten dann mit so einer "good practice" rauszukommen und es für dahin ordentlich zu machen?

[Flieck] Also das ist gerade ein ganz großes Problem, was du ansprichst, nämlich die Finanzierung solcher Arbeiten.

Denn finanziert wird die Softwareentwicklung schon, aber es ist gerne das, wo auch dran gestrichen wird oder das eben, oder also gekürzt wird, was gerade finanzielle Mittel angeht und dann eben auch Personalmittel.

Aber es ist eben auch ein Problem, dass gerade so Sachen wie das Ganze sauber zu schreiben, zu maintain oder überhaupt die Leute erst zu schulen, gar nicht mit groß eingerechnet wird.

Das heißt, ja, es ist eine große Herausforderung, gerade in kurzen Projekten.

Deswegen ist unser langfristiges Ziel, die Leute frühzeitig abzuholen, eben zum



Beispiel durch diese Schulungen, die wir auch für Leute, die neu anfangen und das erste Mal mit Softwareentwicklung zu tun haben, sehr stark empfehlen, dass wir sagen, kommt mal zu uns, wir zeigen euch ein paar Grundlagen, wir zeigen euch die wichtigsten Sachen, damit ihr eine Vorstellung habt, was man tun kann und was möglich ist und was hilfreich vor allem ist und dann, damit ihr halt wirklich von Anfang an auch eben nachhaltig eure Forschungssoftware entwickelt.

Sprich, so dass sie eben auch für euch selbstverständlich ist, über längere Zeit etc.

Und man muss sagen, das funktioniert ganz gut.

Wir haben halt die Leute, die frisch aus der Uni kommen, die entweder schon vermittelt bekommen haben, wie wichtig auch in ihrer Domäne Software ist oder wir haben auch die erfahrenen Leute, die einfach schon viele Jahre in der Forschung sind und einfach mehrere Softwareprojekte gesehen haben, die vor die Wand gefahren sind, weil die eben nicht ordentlich entwickelt wurden, sondern nur auf schnell und hauptsache es funktioniert irgendwie für dieses Projekt und die dann sagen, okay, so geht es nicht weiter, wir wollen das jetzt richtig machen und die dann zu uns kommen und sich auch ein Stück weit die Zeit nehmen.

Aber wie gesagt, wir versuchen immer, möglichst effizient zu sein, das heißt möglichst wenig Aufwand für möglichst viel Effekt und dass die Leute das nicht nur für das Projekt mitnehmen, sondern vor allem auch in ihre andere oder für das nächste Projekt, was sie machen.

Und das ist eigentlich das Ziel.

Also es geht nicht immer nur darum, sozusagen das Retten des einzelnen Projekts, sondern es geht darum, die Leute in die Lage zu bringen, das zukünftig auch von Anfang an vielleicht besser zu machen.

[Siebert] Wer genau ist das eigentlich, der da die Software schreibt?

Also in diesen Forschungsgruppen beispielsweise, sind das alle Forscherinnen und Forscher?

Also ist es der Typ, der auch die Sensor, der die Daten aus dem Sensor popelt, schreibt der sich die eigene Software oder hat der jemanden mehr oder minder zentral, dem er sagt, hier sind meine Daten, so sehen die aus, schreibt mir mal eine Software, die damit umgehen kann?

[Kleinwähler] In der Regel sind das die Leute selbst.

[Siebert] Das sind doch viel zu viele, als dass du das ordentlich über die gesamte Helmholtz-Gemeinschaft ausrollen könntest, jedenfalls nicht in kurzer Zeit.

[Kleinwähler] Das ist richtig.

Also wenn man bedenkt, wir geben so alle zwei Wochen eine Schulung oder fast, ja, und da sitzen dann so zwischen 20 und 25 Leute drin und dazu, ich weiß gar nicht wie viele Leute, muss ich sagen, auf Helmholtz-Ebene arbeiten, alle im DLR sind schon fast 10.000, dann ist es natürlich offensichtlich keine Relation zueinander.

Wir sind aber nicht die einzigen, die es was anbieten.

Also die Zentren haben teilweise intern auch ihre eigenen Schulungssysteme oder es gibt eben auch noch andere Sachen und es gibt auch viel öffentlich, was man machen kann.

Trotzdem ist es ein Stück weit natürlich ein "Fass ohne Boden" oder so.

Aber wir sehen trotzdem, dass es total wichtig ist, denn das ist ein Multiplikator.

Jeden, den wir unterrichten, geht in ein Projekt und sagt, hier, ich habe das und

das gelernt, wollen wir das nicht anwenden?

Und wir haben da auch extra aufeinander aufbauende Schulungen, also die, die wir vom DLR anbieten, wovon die letzte auch irgendwie Softwareentwicklung im Team ist, wo es darum geht, wirklich Leute zu unterrichten, zum einen, was wichtig ist, was sie benutzen können, aber es muss nicht immer jeder in einem Projekt sein, der der Experte in allem ist, sondern dass man dann wirklich sagen kann, okay, so die Grundlagen, das sollte hoffentlich dann irgendwann alle sich entweder selbst beibringen oder eben so eine Schulung bei uns besucht haben, aber für weitere Sachen geht es darum, dass wir Multiplikatoren ausbilden wollen.

Leute, die die Institute gehen und sagen, okay, hier oder in ihre Forschungsgruppe, wir sollten das so und so machen, dann läuft das besser.

Und man muss sagen, das funktioniert in der Hinsicht, dass wenn nämlich so ein Projekt plötzlich in der Arbeitsgruppe mal richtig gut läuft, im Gegensatz zu anderen Softwareprojekten oder die Leute auch viel zufriedener sind.

Die meisten arbeiten ja nicht nur in einem Forschungsprojekt, sondern oft auch gerne mal in mehreren.

Das heißt, die übernehmen das einfach in ihr nächstes Projekt oder die Leitung sieht, oh, das hat gut funktioniert und sagt dann auch, ja, hier, dieses Projekt startet jetzt neu.

Macht euch doch mal bei denen schlau, die machen das so sauber.

Das heißt, das ist natürlich auch ein bisschen unser Ziel, dass wir, wie gesagt, nicht alle an der Hand nehmen, sondern die Idee ist, dass die Leute immer was mitnehmen und das multiplizieren und durchsetzen.

Wenn du sagst, wie viele Leute arbeiten eigentlich in HIFIS, Uwe?

Es sind um die 40.

Wobei für den Softwarecluster, also schon in der UK, sind es ungefähr 12 oder 15.

Also im UK sind es super wenige, aber da haben wir noch ein paar Leute, die unterstützen.

Ja, ja.

Also ich meine Softwarecluster insgesamt.

Und da wird tatsächlich auch auf nicht HIFIS-Leute teilweise mit unterstützen zurückgegriffen.

Aber es ist ja tatsächlich gesagt worden, wir können und wollen jetzt nicht Ende zu Ende jeden einzelnen Wissenschaftler, der einzelnen Wissenschaftlerin an die Hand nehmen.

Das ist auch gar nicht das Ziel, sondern eben tatsächlich auch diese guten Umgangsformen in der Softwareentwicklung auch über Multiplikatoren verbreiten.

Und das schafft man schon ganz okay.

Also ich glaube, die Education-Zahlen waren jetzt, sind knapp 100.000.

Es waren so 700 oder sowas in der Art Teilnehmer, die wir da zuletzt hatten.

Da kommen schon ein paar Zahlen zusammen, aber es ist klar, wir müssen noch ein bisschen mehr skalieren und auch die Breite in Helmholtz noch ein bisschen mehr erreichen.

Weil es ist ja auf jeden Fall schon klar, je früher man anfängt in der Softwareentwicklung, die guten Praktiken auch zu nutzen, desto weniger Arbeit

hat man hinterher, das dann wieder aufzunehmen.

Aber gerade deswegen ist es auch ganz wichtig, zum Beispiel im Education-Bereich, aber auch in anderen, dass wir unsere Materialien teilen.

Also unser Ziel ist gar nicht da, so einen Monopol zu haben auf "Wir sind die, die die Schulung geben", sondern wir sagen, okay, hier sind unsere Schulungsmaterialien, hier ist irgendwie so ein bisschen Unterstützung und Anleitung, so machen wir das.

Und wir laden auch Leute mal herzlich ein, an unseren Schulungen teilzunehmen, nicht um selbst zu lernen, sondern zu erleben, wie wir schulen, damit die das dann auch selbst machen können.

Weiß ich, ob im Rahmen von HIFIS oder nicht, ist uns vollkommen egal.

Das Wichtigste ist, dass es einfach angeboten wird.

Jetzt haben wir über Software geredet, wir haben über Cloud geredet.

Ein Wort steht hier noch, Backbone.

Uwe, was ist das?

Das war einer der Teile von HIFIS.

Ihr legt Kabel.

Ja, Backbone ist tatsächlich...

Natürlich geht es da auch um Kabel, ja, das ist gar nicht so ganz daneben.

Also Backbone Services ist tatsächlich ein Teil von HIFIS, der sozusagen die Grundlage ist dafür, dass andere Teile sauber funktionieren können.

Einen ganz wichtigen Aspekt aus dem Backbone habe ich vorhin mal ganz kurz als Abkürzung angesprochen.

Das ist die AHI, diese Authentifizierungs- und Autorisierungsinfrastruktur, ein ganz fürchterliches Wort.

Und die sozusagen ermöglicht es, dass man eben als Nutzer, Nutzerin von irgendwoher einen Service irgendwo anders nutzen kann, ohne dass das entweder anonym ist oder ohne dass man dafür irgendwelche Accounts mit Passwörtern und so ein Krimskrams anlegen muss.

Also man kann sozusagen sich mit seinem heimischen Account-Usernamen anmelden und dann fremde Ressourcen nutzen.

Das macht im Hintergrund diese AHI.

Und weil das im Hintergrund erst mal etabliert sein muss, damit zum Beispiel Cloud-Services vernünftig funktionieren oder damit zum Beispiel auch das kollaborative Versionskontrollsystem funktionieren kann mit entsprechenden automatischen Testings, die gerade angesprochen wurden, weil das sozusagen erst mal als Hintergrund funktionieren muss, ist das sozusagen in dem Backbone-Cluster, wenn man so will, ausgelagert oder extra behandelt und wird dort etabliert und gepflegt.

Auch andere Dinge wie zum Beispiel große Datentransfers werden dort untergebracht.

Beispielsweise, wenn es eben darum geht, ich habe auf einem, in einer Stelle große Datenmengen erfasst und rumliegen, aber woanders Rechenkapazitäten in großer Menge, dann ergibt sich ja das offensichtliche Problem, dass die Daten irgendwie transferiert werden müssen.

Das kann durchaus im Multiterraweit-Bereich liegen und dafür braucht man

wiederum auch Hintergrunddienste, die sowas ermöglichen und dafür gibt es beispielsweise eben einen HIFIS-Transfer-Service und das ist auch alles Backbone, weil das sozusagen im Hintergrund oder als Unterbau, wenn man so will, von höheren nutzbaren Diensten auch funktionieren muss, aber ohne dass es unbedingt immer gesehen wird von den Nutzern.

Ich wollte gerade sagen, das merken die Wissenschaftlerinnen und Wissenschaftler gar nicht.

Ja, sie merken es vor allem dann, wenn es nicht funktioniert.

[Lachen] Ja, also eine gewisse Kontaktfläche gibt es natürlich schon und also im Regelfall soll es relativ unauffällig im Hintergrund funktionieren.

Da wir aber, wie gesagt, an manchen Stellen auch in der Aufbauphase sind, neigen wir dann oft auch dazu, Dienste zur Verfügung zu stellen, weil sie auch wirklich sehr gefragt sind, die vielleicht noch nicht so super komfortabel, transparent und nutzbar sind, so hier, klick mal hier und los geht's, sondern da muss man vielleicht zweimal mehr klicken, aber es funktioniert dann trotzdem schon und in dem Moment kommt natürlich auch ein Nutzer mal mit Backbone-Services in Kontakt, aber das funktioniert dann für etliche tausend ja inzwischen doch ganz gut.

Wo sind eigentlich eure Grenzen, wobei könnt ihr nicht helfen?

Also bei der Software, was klassisch sowas kommt, ist, was du vorhin noch so ein bisschen hattest, als Beispiel, dass nämlich Leute kommen und sagen, okay, ich habe das und das Problem und könnt ihr das für mich komplett programmieren?

Aber wäre natürlich, wäre doch aber cool, wenn sowas zentral ginge, oder?

Das wäre mega, aber wir sind in diesem Consulting-Team von den Leuten, die jetzt konkret die Consulting machen, fünf Leute und so eine Software schreibt man auch nicht mal so eben an einem Nachmittag runter, was da gefragt wird,

sondern das ist dann doch eher Sachen, was so locker mal ein Jahr oder so dauert.

Ganz am Anfang haben wir bei kleinen Projekten das mal sogar machen können, weil wir wenig Leute waren, zumindestens mit entwickeln.

Das ist mittlerweile einfach nicht möglich.

Dafür haben wir zu viele Anfragen, zu wenig Leute.

Da ist halt unsere Grenze.

Wir können unterstützen, wir können beraten, aber wir können nicht die Entwicklung übernehmen.

Ja, was wir zum Beispiel auch nicht können, zumindest aktuell nicht, ist, neue Dienste anbieten, die es einfach nicht gibt.

Also die, wo man, die man vom Scratch entwickeln müsste oder die irgendwie ganz neu aufgesetzt werden müssten.

Es gibt dann manchmal so Wunschlisten, die bei uns auch im Hive-Asset-Desk aufschlagen.

Ich hätte mal gerne so eine bestimmte Projektmanagement oder keine Ahnung, einen ganz speziellen Use Case, wo wir tatsächlich mit dem, was wir schon haben, nicht direkt dienen können, wo wir dann zwar anbieten können, das könnte man so und so auch abbilden mit verschiedenen Diensten oder mit der Verknüpfung von verschiedenen Diensten, aber genau diesen Use Case können wir halt vielleicht nicht direkt befriedigen und dafür fehlen natürlich Kapazitäten, um das jetzt irgendwie frisch aufsetzen zu können.

Das ist auch nicht unsere Aufgabe.



Aber was wir auf jeden Fall machen können, ist, den Leuten, den Menschen beibringen, wie man mit den vorhandenen Diensten ihre Aufgaben, soweit wir das nachvollziehen können, eben auch gut erfüllen kann.

Eine große Herausforderung ist es, mit den Leuten auch mitzuteilen, welche Dienste es denn gibt.

Sind das so viele?

Ja, das sind nicht unbedingt so viele, aber das Problem ist, Forscher und Forscherinnen bleiben ja oft nur für ein paar Jahre in Forschungszentren.

Und also DLR gehört jetzt zu eines der größten Zentren im Helmholtz-Bereich oder ist das das größte Zentrum?

Und auch hier haben wir schon bei uns bei internen Services, dass Leute sagen, wie Moment, das gibt es bei uns.

Oder eben auch, ja hier, das gibt es jetzt auf Helmholtz-Level, weil wir haben natürlich keinen Mail-Verteiler, wo wir irgendwie alle Leute in Helmholtz erreichen können.

Das wäre auch ein bisschen hart.

Vielleicht interessiert es ja doch nicht jeden.

Die Unachtsamen, die dann "Reply all" machen, sind das Problem.

Ja, genau.

Das ist ja genau, was man nicht möchte.

Und deswegen ist natürlich für uns die Herausforderung, das, was Uwe schon vor einiger Zeit mal beendet, dieses Werbetrommel-Rühren.

Das heißt, erstmal alle darauf aufmerksam zu machen, was es gibt und auch wofür das gut ist.

Weil wenn wir jetzt sagen, ja hier, neues Service XYZ, dann wissen ja Leute noch gar nicht, ja was mache ich denn damit so und haben auch vielleicht gar nicht die Zeit oder Energie für alles, was wir irgendwo mal hören, dann hinterher zu googeln oder zu suchen, was ist denn das, wofür kann ich das benutzen, wäre das vielleicht für mich hilfreich.

Das heißt, das ist noch eine der großen Herausforderungen, vor denen wir stehen und die wir mit den aktuellen Strukturen auf jeden Fall auch schwierig werden, weil, wie erwähnt, elf Zentren sind dabei, aber eben nicht allen.

Das heißt, in elf Zentren haben wir zumindest einen Ansprechpartner, in den anderen aber nicht.

Das heißt, da muss man dann auch erstmal gucken, wer ist der richtige Kontakt.

In dem Zusammenhang beibringen, was es für Services gibt und was man damit machen kann, möchte ich ganz kurz nochmal erwähnen, dass wir jetzt auch in letzter Zeit zunehmend kleine Tutorials schreiben, für was kann ich eigentlich mit Service X auch noch machen.

Also zum Beispiel das schon mal erwähnte GitLab bieten wir ja auch an als Versionskontrollsystem, aber was man jetzt eben unter [hives.net/news](https://hives.net/news), wer trägt, auch zum Beispiel ab und zu nachlesen kann, sind jetzt auch Tutorials, was man damit noch alles machen kann.

Also beispielsweise kann man damit auch wissenschaftliche Diagramme kollaborativ bauen, was man vielleicht nicht als allererstes im Sinn hat, wenn man GitLab hört.

Oder man kann beispielsweise auch seine wissenschaftlichen Publikationen in

LaTeX damit zusammenschreiben und das auch automatisiert durchtesten lassen.

Das sind so Sachen, die vielleicht nicht unbedingt gleich als allererstes kommen, wenn man GitLab hört, aber trotzdem sehr gut damit gemacht werden können.

Also so als Beispiel, dass wir da auch versuchen, Möglichkeiten aufzuzeigen und How-Tos, Tutorials aufzuzeigen, was man mit den Services so einfach machen kann, ohne dass man deswegen immer gleich eine Ende-zu-Ende-Education-Kommunikation machen müsste.

Ihr seid nur ein Angebot, oder wird es dann irgendwann so weit sein, dass die Forscherinnen und Forscher verpflichtet sind, sich mit eurem Angebot auseinanderzusetzen, beziehungsweise es auch zu nutzen?

Das hängt tatsächlich Zentren.

Das beantworte ich jetzt erst mal von meiner Seite, weil das relativ zentral ist zu dem Punkt, was ich vorhin schon mit dem Helmholtz-Cloud-Vertrag erwähnte.

Im Moment sind wir tatsächlich einfach ein freiwilliges Angebot.

Jeder User, jede Forschungsgruppe kann das frei für sich entscheiden, das zu nutzen oder eben auch nicht.

Das wird zukünftig ein bisschen so sein, dass es natürlich das eine oder andere Zentrum geben wird, das aus eigenen freien Stücken entscheidet, okay, also dieses Service, es gibt es ja dort schon, brauche ich hier nicht mehr lokal zu behalten.

Das ist im Moment noch nicht so, aber das wird vermutlich passieren können, wäre ja im Sinne des Synergie-Effektes auch wünschenswert.

Und dann würden natürlich Nutzer aus dem Zentrum, die den Dienst X brauchen,

dann auch logischerweise dann dahin geschoben werden, aber dafür brauchen wir eben dieses Contract-Framework, damit sowas dann auch sicher möglich ist.

Seid ihr eigentlich auch Forscher oder seid ihr nur Dienstleister?

Carina sagte vorhin, wir sind ja auch Forscher, aber was forscht ihr denn dann?

Also fallen da auch Grundlagenerkenntnisse raus?

Also bei mir ist es jetzt weniger Grundlagen, sondern es geht wirklich zu schauen.

Ich weiß nicht, wir geben zum Beispiel auch Workshops und da haben wir einfach mal uns dann angeguckt, eigentlich relativ elementar.

Wir geben also noch vor Corona, haben wir die eben vor Ort gegeben.

Wie schaut das aus, wenn wir die an verschiedenen Locations gegeben haben, mit verschiedenen Themenschwerpunkten, wer meldet sich eigentlich an?

Das Ganze ist so eine Workshop-Serie gewesen.

Empirische Sozialforschung.

Kommen die Leute wieder?

Genau, es ist so ein bisschen Empirische Sozialforschung, die wir da drauf machen.

Aber das ist so ein Aspekt, der mich interessiert, was gerade im Rahmen von HIFIS gut möglich ist, aber auch was ich zum Beispiel oder meine Gruppe erforscht, ist der Bereich von eben, wie Leute Software entwickeln und zwar gucken wir uns dafür den Code an und wie der sich verändert über die Zeit.

Das heißt, wir gucken uns wirklich den Code als Datenbasis an und da ist HIFIS für uns auch ein bisschen eine interessanter Datenfundmöglichkeit, weil zum Beispiel in Projekten, wo wir jetzt ein Consulting vornehmen, sagen wir dann, okay, dürfen wir eure Repositories mining, Repository heißt jetzt den Source Code, aber vielleicht auch da, wo ihr eure To-Dos trackt.

Das kann auch in so einem Tool wie GitLab sein oder in einem separaten.

Und dann sammeln wir uns diese Information raus und wissen dann ja, wann haben wir Änderungen vorgenommen.

Und dann ist für uns interessant, ist das Ganze auch sichtbar im Code?

Verändert sich das Verhalten, wie Code geschrieben wird?

Wird häufiger committed, nennt sich das, also sozusagen, man könnte jetzt vereinfacht sagen, gespeichert, der Code.

Wird mehr mit To-Dos gearbeitet, also sogenannten Issues, wo man reinschreibt, was man machen möchte.

Wird das in Verknüpfung gesetzt mit dem, was man dann sozusagen speichert.

Da gibt es dann so eine Nachricht, da kann man das referenzieren, dass das miteinander zu tun hat.

Also verhält sich dieser ganze Entwicklungsprozess, nennt man das, also wie einfach entwickelt wird, verändert sich das.

Und das interessiert uns einfach, hat das wirklich einen Einfluss?

Und vor allem natürlich für den Moment, wo wir den Einfluss haben.

Aber wie sieht das vielleicht auch ein halbes Jahr oder ein Jahr später aus?

Haben diese Sachen, die wir mit den Leuten gemacht haben, wirklich eine langfristige Auswirkung?

Oder fallen sie doch zurück in ihre alten Verhaltenspattern, sozusagen, also wie sie sich halt vorher entwickelt haben, den Code?

Und das ist was, was uns halt interessiert und wir dann auch erforschen, auf Basis von HIFIS, aber weniger in HIFIS selbst.

Uwe, hast du das so gewollt?

Dass da jetzt HIFIS missbraucht wird als Datengenerator?

Naja, also wir unterstützen ja Forscher.

Also der Hauptauftrag von HIFIS ist ja nicht unbedingt, selbst Forschung zu betreiben, sondern Forschung zu unterstützen und zu ermöglichen.

Also wir sind die Forschungs-Enabler, wie soll ich, Enabler.

Das heißt, wir sind unterschiedliche Teile von HIFIS, arbeiten teilweise sehr eng mit Forscherinnen und Forschern zusammen und haben deswegen auch einen engen Blick darauf.

Also zum Beispiel auch bei Cloud-Services kennt man das, dass dann eben beispielsweise bestimmte Datenverarbeitungstoolboxen dann umgesetzt werden in virtualisierte Umgebungen, damit man die beispielsweise gut skalieren und nach außen anbieten kann, wiederum als Service, ja.

Also dass das eben nicht nur auf einem heimischen Laptop läuft und da irgendwelche Daten verarbeitet, sondern dass man das von außen auch vernünftig, großskalig zugreifen kann.

Und dann sitzen natürlich HIFIS-Mitarbeiter oder auch solche, die mit HIFIS assoziiert sind, dann sehr nah bei den Forschern dran und implementieren das.

Und dann sind sie natürlich auch sehr forschungsnah.

Und dann käme eigentlich der Punkt, der vorhin schon mal angesprochen wurde, ins Spiel, dass eigentlich diese Research-Software-Entwicklung auch eine höhere Gewichtung eigentlich auch ein bisschen haben müsste in der Forschungs-Akzeptanz, in der Forschungsgelder-Akzeptanz.

Aber gut, das ist ein anderes Thema.

Also Teile von HIFIS sind sehr nah an der Forschung, aber HIFIS selbst insgesamt als große Unternehmung ist ja ein forschungsermöglicher und nicht selbst ein Forschungsprojekt.

Wie gehen denn die Wissenschaftlerinnen und Wissenschaftler mit euch um?

Sehen die euch so als dieser Techniker, der ist informiert, "Komm mal vorbei, mach mal heile!"

Oder behandeln die euch anständig auf Augenhöhe?

Alles.

Alles?

Es ist sehr unterschiedlich.

Ich muss sagen, im direkten Gespräch, zum Beispiel in Consultings, dann ist es eigentlich immer auf Augenhöhe, dass die eigentlich sehr dankbar sind, dass wir diesen Dienst anbieten, dass die um jede Hilfe eigentlich sehr, sehr glücklich sind und sehr froh sind, dass es so etwas wie HIFIS gibt.

Lange Jahre war es eben nicht so, und man war in diesem Bereich doch sehr auf sich selbst gestellt.

Aber natürlich gibt es auch den ein oder anderen, der mit so einer Erwartungshaltung kommt, "Ja, jetzt komm und mach mir das mal schön."

Die holen wir dann relativ schnell auf den Boden der Tatsachen zurück.

Wir helfen denen natürlich.

Und dann helfen wir ihnen so gut wir können.

Auf den Boden der Tatsachen zurück.

Wir müssen halt erst mal erklären, wir sind ja auch nur eine Handvoll Leute und wir können jetzt nicht eure komplette Softwareentwicklung übernehmen und ihr seid froh, dass das wer anders macht, weil ihr kein Geld und keine Zeit und keine Leute habt.

Sondern erklären wir ihnen, "Okay, das ist leider nicht das, was wir anbieten können."

Es basiert halt meistens auf Missverständnissen einfach, weswegen wir aber auch auf unseren Webseiten etc. sehr versuchen, klar zu kommunizieren, was wir anbieten und was nicht, damit sich halt kein falsches Bild aufbaut.

Also ich glaube, bisher kam da keiner mit bösen Absichten, sondern einfach mit Missverständnissen.

Und nach einem klärenden Gespräch, wo wir erklärt haben, "Okay, was ist HIFIS und was kann es bieten?"

waren alle doch sehr zufrieden mit der Hilfe, die sie von uns bekommen haben.



Auch wenn es nicht immer in initialen Vorstellungen entsprach.

Carina Haupt und Uwe Jant, vielen Dank.

Vielen Dank, dass du uns da hattest.

Vielen Dank für die Einladung.

[Musik] [Musik] [Ende]