



Projektarbeit

Internationale Hochschule Duales Studium

Studiengang: Informatik

Teamarbeit in der Informatik

Müller Korbinian

Matrikelnummer: 102302316

Adresse

Lohwald Straße 59

86356 Neusäß

Abgabedatum: 31.03.2024

Inhalt

Einleitung.....	3
Aufgaben	3
Eine sinnvolle Team-Aufgabe	4
Wie organisiert sich das Team.....	4
Tobit David.....	4
Trello.....	4
GitLab.....	5
PM.....	5
Wie nützlich ist die Whitehead Ausführung.....	6
Projektvorgaben und die Projekt-Funktionen	6
Erstellen Projektdesign.....	6
Verwalten von Dependencies.....	6
Dependencies von Entwickler	7
Bugtracking.....	7
Projektdokumentation	7
Fallbeispiel	7
Unsere Werte	9
Wissen entwickeln und Teilen.....	9
Pragmatisch handeln	9
Der Teamgeist ist alles.....	10
Wie helfen die Werte die Aufgabe zu erfüllen	10
ACM Code of Ethics and Professional Conduct.....	11
Wie lässt sich die Teamarbeit fördern.....	12
Teamlearning.....	12
Pair Programming	13
Feedback zu den Vorschlägen	13

Einleitung

Im Folgenden wird die Aufgabestellung im Rahmen des Praxisprojektes im ersten Semester beantwortet. Für die Beantwortung der Fragen ist der relevante Praxispartner die Firma mpunkt GmbH. Die Firma ist Süddeutschlands älteste Webagentur. Das Spezialgebiet ist sowohl die Beratung als auch das Umsetzen von Webprojekten. Von Webseiten, Webshops bis Web-Apps ist alles möglich. Die Firma mpunkt ist die älteste Internetagentur im süddeutschen Raum. Insgesamt sind es 18 Angestellte. Darunter sind neben Softwareentwickler auch Mediengestalter. Die meisten unserer Projekte basieren auf dem CMS-System Typo3. Hier findet PHP in der Version 8.0 Verwendung. Aber React, React Native oder Nextjs sind und nicht fremd.

Aufgaben

Im folgenden sind die Ausführung zu den Gestellten Fragen.

Eine sinnvolle Team-Aufgabe

Eine der wichtigsten Aufgaben, die die Entwickler machen, ist die technische Realisierung der Kundenprojekte. Hier beginnt die Teamarbeit mit dem ersten Meeting mit dem Kunden. Jedes Projekt ist einzigartig. Um dem Kunden die beste Lösung zu präsentieren, nimmt hier bereits ein Entwickler am Meeting teil. Nachdem die ersten Informationen gesammelt wurden, geht es an das Erstellen eines Projektplanes. Durch die Zusammenarbeit von Entwicklern und Designern entsteht eine optimale Lösung für den Kunden. Sobald die Rahmenbedingungen geschaffen wurden, geht es mit der Entwicklung los. Bei jedem Projekt ist auf jeden Fall ein Designer und ein Entwickler involviert. Es wird per Default interdisziplinär im Team gearbeitet. Während der Designer sich zunächst um die Gestaltung kümmert, setzt der Entwickler das Projekt auf. Dies beinhaltet unter anderem das Anlegen neuer Repositories und Datenbanken. Mit dem Fertigstellen des Screendesigns geht es mit dem Umsetzen los. Da die meisten Projekte umfangreicher sind, insbesondere das Entwickeln von Webshops, werden nun weitere Entwickler dazugezogen.

Im Team werden indessen die Aufgaben anhand der persönlichen Stärken verteilt. Jeder kümmert sich um seinen Bereich. Es wird sich im Team abgestimmt und ausgetauscht. Durch die enge interdisziplinäre Zusammenarbeit wird das Projekt umgesetzt. Bereits während der Entwicklung werden Reviews durchgeführt, um eine hohe Qualität zu gewährleisten. Dabei übernimmt ein Kollege den Review. Wenn die Entwicklung so weit abgeschlossen ist, wird die Software auf einer Testumgebung deployt. Die Software wird jetzt strengen Tests unterzogen. Dabei helfen auch Kollegen, die nicht in der Entwicklung beteiligt waren. Dadurch wird sichergestellt, dass die Webseite/Webshop intuitiv zu bedienen ist. Danach ist die Software bereit, dem Kunden übergeben zu werden.

Wie organisiert sich das Team technisch

Tobit David

Das Herzstück der Kommunikation bildet dabei die Software Tobit David. Die Software beinhaltet E-Mail-Versand. Sowohl Firmen intern als auch extern. Intern können E-Mails mit Anmerkungen versehen werden. Die Anmerkungen können wie ein Chat benutzt werden. Eine weitere Funktionalität ist der Chat. Man kann Einzelchats mit Kollegen führen oder Gruppenchats einrichten. Unersetzlich ist der integrierte Einzel- und Gruppen-Kalender.

Trello

Mittels Trello wird das Kanban-Verfahren implementiert. Hier haben nicht nur Teammitglieder Zugriff auf das Projektboard, sondern auch ausgewählte Externe. Dadurch können neue Tickets von beiden Seiten angestoßen werden. Dies bringt sowohl Vorteile als auch Nachteile mit. Zum einen haben die Stakeholder

direkten Einblick, wie der aktuelle Stand ist. Ebenso können sie auch direkt Bugs auf diese Weise melden. Auch die beteiligten Entwickler haben einen Überblick über die noch zu erledigenden Tickets. Sowohl das Beheben von Bugs als auch das Anstoßen von Reviews wird dadurch verschleunert. Aber es kann auch zu Komplikationen führen. Für die Stakeholder ist es einfacher, die Projekt-Parameter zu ändern und zu ergänzen. Dies führt zu einem erhöhten Kommunikationsaufwand. Es wird gerne unterschätzt, dass einfache Änderungen teilweise für einen erheblichen Mehraufwand sorgen. Ebenso haben die Stakeholder eine andere Auffassung, welche Tickets nun eine höhere Priorität haben.

GitLab

Wie bei allen Firmen, in denen kollaborativ an Software entwickelt wird, kommt auch bei mpunkt Versionskontrolle zum Einsatz. In diesem Fall ist es GitLab. Die Versionskontrolle ermöglicht zur gleichen Zeit an der Codebasis zu arbeiten. Es funktioniert, indem Versionsstände, Branches, definiert werden. Grundsätzlich wird hier zwischen dem Main, Staging, Fix und Feature unterschieden. Main und Staging spiegeln hier die Stände wider, die auf ausgerollt werden. Das Feature dient der Entwicklung neuer Funktionalitäten. Fix stellt die Fehlerbehebung dar. Jede Codeanpassung wird in Form eines Comit auf den jeweiligen Branches gespeichert. Dadurch ist nachvollziehbar, welcher Entwickler, zu welcher Zeit, Änderungen am Code vorgenommen hat. Sobald im Laufe der Entwicklung Fehler auftreten, kann anhand der Commits herausgefunden werden, wann und durch welche Codeänderungen die Fehler entstanden sind. Wenn jetzt ein Zeitpunkt identifiziert wurde, an dem der Fehler noch nicht existierte, kann der Branch bis zu diesem Commit rückgängig gemacht werden. Destruktive Aktionen, wie das Löschen von Dateien, können auf die Art rückgängig gemacht werden. Wenn die Arbeit an einer neuen Funktionalität fertiggestellt ist, werden die Änderungen der einzelnen Entwickler zusammengeführt. Dieser Vorgang nennt sich Mergen. Einfache Änderungen werden direkt von GitLab übernommen. Sollten es nun Stellen geben, an denen die Commits einen deutlichen Unterschied aufweisen, kommt es zu einem Merge Konflikt. Ein Beispiel dafür wäre, wenn eine Methodensignatur deutlich unterschiedlich wäre. In diesem Fall entscheidet dann der Entwickler, welchen Codestand übernommen wird.

PM

PM, auch bekannt als Projekt Manager, ist eine Eigenentwicklung von mpunkt. Er dient dazu, die Verwaltung der Projekte zu übernehmen. Hier werden die einzelnen Kunden erfasst. Jedem Kunden werden dann Projekte zugewiesen, die umgesetzt werden. Hier wird das erste Mal Einzelheiten zu dem Projekt definiert. Dazu zählt, wer der Ansprechpartner bei dem Kunden ist und wie man ihn am besten erreichen kann. Ebenso wird der Projektleiter und die Projektbeteiligten hinterlegt. Ferner kann auch die verwendete Software und deren Version hinterlegt werden. Innerhalb eines Projektes gibt es Aufträge. Hier könnte Screendesign oder die technische Umsetzung als Aufträge eingetragen sein. Für jeden Auftrag gibt es ein Stundenbudget. Über den PM wird überwacht, wie viel Zeit in jedes Projekt gesteckt wird.

Wie nützlich ist die Whitehead Ausführung

J.Whitehead(2007) beschreibt am Anfang seines Artikels Ziele, die beim Zusammenarbeiten von Softwareingenieuren relevant sind. Diese sind die Definition der Projektvorgaben und die Projekt-Funktionen, das Zusammenführen auf ein Systemdesign, das Verwalten von Dependencies, die Abhängigkeit zwischen den Entwicklern zu reduzieren, Fehler erkennen und beheben und die Projekte dokumentieren. Ebenso führt er an, dass Programme eingesetzt werden, damit die Ziele besser erreicht werden. Dabei können die Programme in vier Kategorien unterteilt werden. Zunächst einmal gibt es die modellbasierten Tools, dabei steht die formale Dokumentation, z. B. UML, im Vordergrund. Die nächste Kategorie sind die Prozess-Support-Tools, mit diesen wird der gesamte Arbeitsfluss in der Entwicklung abgebildet. Die dritte Kategorie sind die Awareness-Tools. Hier liegt der Fokus darauf, alle Projektbeteiligten mit neuen Informationen zu versorgen. Laut Whitehead ist die letzte Kategorie Kollaboration-Infrastruktur. Diese Tools sorgen für eine Integration der anderen Tools und stellen eine Plattform dar, um Daten auszutauschen.

Projektvorgaben und die Projekt-Funktionen

Die Projektvorgaben und die daraus resultierenden Funktionen sind der Grundstein für jedes Projekt. Jedoch ist es nicht direkt ein Aufgabenbereich von mpunkt. Dies liegt daran, dass wir als Dienstleister im Bereich Webentwicklung die Vision unserer Kunden umsetzen. Dementsprechend besitzen die Kunden bereits eine konkrete Vorstellung, was das Projekt ist, welche Funktionen es besitzen soll und was für Parameter existieren. Nur wenn es aus unserer Sicht zu Problemen führt oder unvollständig ist, wird eingeschritten. Dies geschieht in der Regel in einem Meeting mit dem Kunden. Hier werden natürlich die gängigen Kommunikationsmittel eingesetzt.

Erstellen Projektdesign

Hier muss unterschieden werden, um was für ein Projekt es sich handelt. Geht es um das neue Erstellen oder einen Bereich einer Webseite, gibt es bereits vordefiniertes Projektdesign. Dieses basiert auf dem CMS-System Typo3. Hierfür gibt es auch einen bereits bestehenden DEV-OPS-Plan. Für eine höhere Effizienz wird mit einer Boiler Plate gearbeitet, die an die Kundenvorgaben angepasst wird. Bei einer neuen Entwicklung wird versucht, so viel wie möglich von dem bestehenden Projektdesign zu übernehmen. Die anderen Aspekte werden zusammen mit dem Kunden ausgelotet.

Verwalten von Dependencies

Die Verwaltung von Dependencies wird mittels der Features der Programmiersprachen und deren Frameworks gelöst. Unsere Projekte bauen auf unterschiedlichen Webstacks auf. Vor allem wird Node und PHP mit Composer eingesetzt. Damit sind die Dependencies gut überschaubar und anpassbar. Um dies noch weiter zu optimieren, wird Docker verwendet. Die Dependencies werden folglich nicht mehr lokal verwaltet und installiert, sondern in Containern. Dadurch arbeiten alle Beteiligten mit den gleichen Dependencies.

Dependencies von Entwickler

Auch bei mpunkt wird die Abhängigkeit zwischen den Entwicklern reduziert. Jedoch wird es nicht auf einer Workspace Basis realisiert. Hier kommt das Tool GitLab zum Einsatz. Hier legt sich jeder Entwickler sogenannte Branches an. Dabei handelt es sich um separate Codestände. Jeder Entwickler arbeitet an seinem Codestand. Sobald Features entwickelt und getestet wurden, werden die einzelnen Stände zusammengeführt. Die daraus resultierenden Codestände bilden die Grundlage für die Entwicklung weiterer Codestände.

Bugtracking

Für die Dokumentation von Bugs wird keine separate Software eingesetzt. Diese werden in den Kanban-Bord der Projekte erfasst. Hier wird auch dokumentiert, was die Ursache war und wie das Problem gelöst wurde. Falls es um kleinere Fehler handelt, dann werden diese in der Regel direkt von Finder behoben. Erst, wenn dieser keine Kapazität mehr hat, wird es dokumentiert und an den Zuständigen übergeben.

Projektdokumentation

Die Projektdokumentation findet an mehreren Stellen statt. Dokumentation, die sich auf den Code bezieht, wird entweder direkt in Form eines Kommentars an der entsprechenden Stelle erledigt oder in einem Readme. Für die Projektdokumentation wird ein digitales Archiv verwendet. Hier sind neben der formalen Projektdokumentation auch E-Mails, Screendesigns und Medien zu finden. Für die Dokumentation des im Projekt erlangten Wissens ist das Intranet Wiki da. Hier wird es alles festgehalten und dient als Nachschlageplatz.

Fallbeispiel

Im Folgenden wird ein aktuelles Projekt genauer betrachtet. Bei diesem Projekt handelt es sich um den Relaunch der Webseite fsz-Nordbaden.de. Die Webseite soll ein Clone der Schwester-Webseite fsz-Koblenz darstellen. Insgesamt sind drei Personen für das Umsetzen zuständig. Dabei sind SS, Mediengestalter und Projektleiter, PH, Azubi im dritten Jahr. PH hat im Laufe des Projektes das Team verlassen, um sich auf seinen Abschluss zu konzentrieren. Die konkrete Aufgabenverteilung ist wie folgt: SS erstellte den Projektplan und nahm Änderungen an dem Screendesign vor, PH erstellte die Backendvorlagen. Insbesondere für die Angebotsseiten. Ich entwickelte die nötigen Plug-ins, pflegbare Elemente und kümmerte mich um das Debuggen.

Folgendes sind die Projektspezifikationen: Das Screendesign ist das Gleiche, auch soll die Seite die gleichen Funktionalitäten beinhalten. Insbesondere die Verbindung mit der ADAC API für Buchlangstrecken. Das Projekt wird mit dem CMS-System Typo3, in der Version 12, und PHP 8.0 umgesetzt. Dadurch sind die Funktionalitäten und die Requirements bereits fest vom Kunden vorgegeben. An dieser Stelle würden weitere Tools keinen großen mehr Wert bieten.

Das Gleiche gilt auch für Tools für die Projektarchitektur. Die Vorgaben in Bezug auf die Softwarearchitektur werden von Typo3 bereits vorgegeben. Diese wurden durch mpunkt extensiv erweitert und verbessert. Der Fokus liegt in der Erweiterung des Bestehenden, nicht in der Neuentwicklung. Die Nachvollziehbarkeit wird mittels GitLab umgesetzt. Jede Änderung im Code wird als Commit archiviert. Es ist zu jeder Zeit klar, wer am Projekt arbeitet und was die Person entwickelt hat. Die Testphase für das Projekt hat noch nicht angefangen, stand 28.03.2024. Der aktuelle Abschnitt im Projekt ist das Pflegen der Inhalte. Dabei werden in Typo3 alle gewünschten Seiten angelegt und mit Texten und Medien des Kunden gefüllt. Erst, wenn alle Inhalte vorhanden sind, geht es mit dem Test los. Die Tests sind auf Browser und Gerätetests fokussiert. Hier wird die Webseite auf Anzeigefehler und Benutzerfreundlichkeit getestet. Alle Funde werden in Trello, unserer Kanban-Software, als Tickets angelegt. Die Bearbeitungsreihenfolge wird durch die Schwere des Bugs festgelegt. Kritische Fehler sind als solche definiert, dass diese das Nutzen der Webseite verhindern. Bei normalen Fehlern sind einzelne Features nicht ganz einsetzbar oder schwerwiegende Stylingfehler. Leichte Fehler sind in der Regel alle Stylingprobleme. Da an diesem Projekt nur ein kleines Team daran arbeitet und nur ein Entwickler mit dem Debuggen beschäftigt ist, sind weitere Tools für das Bug-Tracing nicht notwendig. Process centered collaboration ist in Teilen umgesetzt. Jedoch wurden auch hier keine weiteren Software-Tools eingesetzt. Um das Aufsetzen zu beschleunigen, existiert eine Typo3 Boiler Plate. Es ist ein fertiger Docker Container. Dieser wurde auch hier eingesetzt. In diesem wurden bereits die Grundeinstellungen vorgenommen. Auch wurden hier bereits die nötigen Schnittstellen vorbereitet. Im Großen und Ganzen war mein Teil der Entwicklung nur die bereits in der Boiler Plate vorhandenen Standard Content-Elemente und Plug-Ins, um die für das Projekt relevanten zu erweitern. Die Aufgabenverteilung wurde direkt zu Beginn festgelegt. Da es zu keiner großen Änderung im Projekte kam, war auch kein Bedarf, diese zu ändern. An dieser Stelle im Unternehmen gibt es auch Projekte, an denen mehr Entwickler daran arbeiten; bei diesen ist es tatsächlich eine Überlegung wert, weitere Tools einzuführen.

Die Collaboration Awareness wurde mittels GitLab verwendet. Hier ist die Gliederung nicht wie im Paper beschrieben nach Workspace, sondern nach Branche. Jeder Entwickler hat auf seinem Branch gearbeitet und wurde benachrichtigt, sobald neue Features auf dem Gemeinsamen Branch veröffentlicht wurden.

In der Firma wird die IDE PHP-Storm von JetBrains benutzt. GitLab ist hier mit integriert.

Zusammenfassend lässt sich sagen, dass das Paper einige interessante Blickwinkel vertritt. Insbesondere in Hinsicht auf die Ziele, die die kollaborative Softwareentwicklung aufweist. Einige der vorgeschlagenen Sachen werden bereits aktiv umgesetzt und haben einen positiven Einfluss auf die Arbeit. Jedoch werden keine der erwähnten Programme direkt eingesetzt. Die meisten Programme hätten nur wenig bis gar keinen positiven Einfluss auf den Workflow im Beispielprojekt. Es ist dem geschuldet, dass es sich um ein relativ kleines Projekt handelt, knapp 200 Arbeitsstunden, und da das Projektteam nur aus wenigen Mitgliedern besteht. Das Einführen von neuer Software, explizit für dieses Projekt, würde zu viel Zeit in Anspruch nehmen, alle

Beteiligten zu schulen. Dennoch sind einige gute Ansätze vorhanden, denen ich weitergehe und sehe, wohin diese führen.

Unsere Werte

Wissen entwickeln und Teilen

Wissen wird sehr großgeschrieben, ganz nach dem Motto „For knowledge itself is power“ (Francis Bacon, 1598). Vor allem in einer Berufssparte, die sich so schnell weiterentwickelt wie die Informatik. Dazu kommt noch, dass bei uns kein Projekt wie ein anderes. Dadurch gibt es immer was Neues zu lernen. Zu Beginn eignen sich die Teilnehmer des Projektes das Wissen an. Während des Projektes wird das Wissen gefestigt. Nach dem Abschluss wird das Wissen mit allen anderen in Form einer Schulung geteilt. Nicht nur projektspezifisches Wissen wird geteilt. Auch Allgemeines über SEO, SEM, Best Practices oder allgemeine Neuheiten. Es wird auch befürwortet, sich neues Wissen anzueignen. Sei es den Umgang mit neuen Tools oder möglichen Technologien.

Um Wissen zu sammeln, ist auch der Besuch von Messen und Masterminds jederzeit möglich. Alles mit dem Ziel, Wissen zu haben und dann sinnvoll zu nutzen. Denn nur mit dem nötigen Wissen ist es möglich, den Kunden den bestmöglichen Service anzubieten. Sowohl in der Beratung als auch dann im Umsetzen des Projektes.

Ein weiterer Vorteil ist, dass sich jeder für andere Themengebiete interessiert. Dadurch entwickelt jeder sein eigenes Fachgebiet und bildet dort geballtes Know-how. Es ist insbesondere wichtig, da sehr unterschiedliche und ausgesprochen spezielle Projekte umgesetzt werden. Egal, welche Probleme auftreten, es ist sehr wahrscheinlich, dass das Problem in ein Fachgebiet eines anderen fällt. Dadurch können die Probleme sehr effizient gelöst werden. Das Schöne daran ist, dass das Wissen weitergegeben wird.

Pragmatisch handeln

Das Sammeln von Wissen markiert lediglich den ersten Schritt in einem umfassenden Prozess. Neben der reinen Akkumulation von Fakten und Erkenntnissen ist auch die praktische Erfahrung von entscheidender Bedeutung. Als eine der etabliertesten Internetagenturen in Süddeutschland hat mpunk im Laufe der Jahre ein reichhaltiges Reservoir an Erfahrungen angehäuft. Es gibt kaum eine neue Technologie, die nicht eingehend erforscht und analysiert wurde, kaum ein Konzept, das nicht ausgiebig erprobt und evaluiert wurde. Diese jahrelange kumulierte Erfahrung bildet die Grundlage für unser gegenwärtiges Handeln, das von einem starken Pragmatismus geprägt ist.

Obschon einige diese Ansammlung an Erfahrungen möglicherweise als eine Einschränkung betrachten mögen, stellt sie letzten Endes sicher, dass wir über ein fundiertes Verständnis und umfassende Kompetenzen in unserem Fachgebiet verfügen. Unsere Kunden können sich darauf verlassen, dass wir genau

wissen, was wir tun, und dass wir in der Lage sind, ihre Anforderungen mit höchster Professionalität und Effizienz zu erfüllen.

Der Teamgeist ist alles

Von seitens des Arbeitgebers wird ein sehr hoher Wert auf die Teamdynamik gelegt. Es wird alles daran gelegt, die Teamdynamik zu verbessern. Auch auf Seite der Arbeitnehmer wird daran gearbeitet. Das Resultat lässt sich auf jeden Fall sehen. Es herrscht eine sehr gute Teamdynamik und ein familiärer Umgang. Dieser starke Teamgeist ist wichtig, da wir kollaborativ Probleme lösen. Durch den familiären Umgang bringt sich jeder ein und trägt zur gemeinsamen Lösung des Problems bei. Ein weiterer Vorteil ist es, dass es eine offene und effektive Kommunikation zwischen Kollegen erlaubt. Auf diese Weise wird der Informationsaustausch erleichtert. Jedoch ist das Allerwichtigste, dass es zu großem Vertrauen untereinander führt. Es gibt nicht Wichtigeres, als sich auf seine Kollegen verlassen zu können. Durch die Unterstützung der Kollegen fällt es leichter, Risiken einzugehen und neue Ideen voranzutreiben.

Wie helfen die Werte die Aufgabe zu erfüllen

Die gelebten Werte haben auf jeden Fall einen sehr großen Einfluss auf die Qualität der von uns umgesetzten Projekte. Ab dem ersten Schritt im Projekt ist der Einfluss deutlich sichtbar. Denn im ersten Treffen sorgt das geballte Wissen und Erfahrung dafür, dass der Kunde optimal beraten wird. Mit dem Wissen und der Erfahrung wird das Projekt analysiert. Handelt es sich um einen einfachen One-Pager? Ist es eine Webseite, die mittels eines CMS-Systems verwaltet werden kann? Ergibt es Sinn, das CMS-Headless zu benutzen? Können wir mit dem bereits vorhandenen arbeiten oder wird es eine komplette Neuentwicklung? Auf einige der Fragen weiß der Kunde bereits eine Antwort. Für alle anderen ist unsere Erfahrung unverzichtbar.

Nachdem geklärt wurde, was das Projekt genau beinhaltet, geht es an die Planung, wie das Projekt nun umgesetzt wird. Durch die langjährige Arbeit mit Typo3 ist es das Default-CMS-System, das eingesetzt wird. Ebenso hat sich Docker für die Entwicklung und die Live-Stellung bewährt.

Insgesamt haben unsere Werte in dem Projektplan dafür gesorgt, dass der Kunde besser beraten wird. Ebenso sorgen sie dafür, dass ein vollständigerer Projektplan erstellt wird.

Der nächste Schritt ist das Umsetzen des Projekts. Hier haben sich über die Jahre einige Docker-Projekte, Vorlagen gebildet. Das Anlegen des Projekts ist dadurch schnell und effizient möglich. Wegen der Docker-Architektur ist auch das Lokale aufsetzen schnell erledigt. Nun geht es an das Eingemachte, die technische Umsetzung des Projektes. Hier sorgen das angesammelte Wissen und die pragmatische Einstellung für eine schnelle und qualitative hochwertige Entwicklung. Insbesondere was die Themen Optimieren der Webseite angeht. Sei es den Page-Speed oder SEO.

Unser Wissen und unsere Erfahrung sind zusammen dafür verantwortlich, dass wir unsere Arbeit so gut machen können. Wir erledigen jedes Projekt schnell und effizient, nach den Wünschen und Vorgaben der Kunden.

ACM Code of Ethics and Professional Conduct

Die erste Sektion des ACM Code of Ethics and Professional Conduct hat nur wenig gemeinsam mit den oben aufgeführten Werten. Dennoch ist jedes der dort aufgeführten Elemente wiederzufinden. Wir legen Wert, ehrlich und transparent mit unseren Kunden umzugehen und handeln stets in deren bester Interesse.

In dem Absatz 2 findet man deutlich mehr, das sich mit den Werten von mpunkt deckt. Absatz 2.1 besagt, dass danach gestrebt werden soll, die höchstmögliche Qualität an Arbeit zu liefern. Als Dienstleister ist es unsere Pflicht, dies zu erfüllen. Das Zusammenspiel unserer Werte, insbesondere das Bestreben, sich neues Wissen anzueignen und die Erfahrungen ermöglichen, die bestmögliche Arbeit zu liefern. Im Absatz 2.2 legt als Grundlage für eine exzellente Arbeit das Streben nach Lernen neuer Kompetenzen dar. Bei uns wird es in Form der internen Mitarbeiterschulungen umgesetzt. Jemand eignet sich die Kompetenzen an und gibt diese an andere weiter. Es entsteht ein Umfeld, das vom Lernen und Weiterentwickeln geprägt ist. Weiter in Absatz 2.4 wird die Wichtigkeit von Reviews dargelegt. Dadurch, dass das Team viele Projekte umgesetzt hat und große Expertise besitzt, sind die Reviews auf höchstem Niveau. Ebenso wird großer Wert auf Feedbackschleifen beim Kunden gelegt. Es gibt gewisse Meilensteine im Ablauf eines Projektes. Fertigstellen des Screendesigns, Klickdummy, Staging und live. Nach dem Erreichen eines jeden dieser Meilensteine wird das Resultat von Kollegen und den Stakeholdern berücksichtigt. Die Ergebnisse werden in den nächsten Entwicklungsschritten berücksichtigt. Abschnitt 2.6 steht etwas in Kontrast mit den Werten. Es besagt, dass nur Arbeit gemacht werden soll, die in der Expertise liegen. Unser Ziel ist es, unser Wissen zu erweitern; dazu zählt auch das Erschließen neuer Kompetenzen. Jedoch wird dies dem Kunden auch eindringlich mitgeteilt. Absatz 2.8 und 2.9 stehen werden strengstens eingehalten. 2.8 Besagt, dass Ressourcen nur mit expliziter Autorisierung abgerufen werden können. Es wird umgesetzt, indem jede Aktion in Form eines Angebots den Kunden vorliegt. Sei es das Pflegen von neuen Inhalten, ein Update oder Datenbanken-Exporte. Erst, wenn das Angebot angenommen wurde und daraus ein Auftrag entstanden ist, wird auf das System des Kunden zugegriffen und die Aktion ausgeführt. Im Absatz 2.9 dreht sich alles um die Sicherheit der Informationssysteme bzw. der Software. Es ist unser höchstes Bestreben, sichere Software zu entwickeln. Hier ist unsere Erfahrung und pragmatischer Vorgang in Kombination mit dem Wissen, was es aktuell an Schwachstellen gibt, sehr wichtig. Es erlaubt bereits in der Entwicklung an kritischen Punkten mögliche Schwachstellen zu entdecken und diese zu beheben. Im optimalen Falle sind so die meisten Schwachstellen behoben, bevor die Webseite auf der Staging-Umgebung ausgerollt wird. Da das Thema Sicherheit bei mpunkt sehr ernst genommen wird, wird darauf geachtet, die ISO-27001-Zertifizierung genau einzuhalten. Dadurch profitieren nicht nur die Industriekunden, sondern auch die Kleinunternehmer. Im dritten Absatz

geht es um das Projektmanagement. Im Großen und Ganzen spiegeln sich unsere Werte in den Absätzen 3.1, 3.3, 3.5 wider. Die anderen spielen auch immer mit.

Zusammen gefasst lässt sich sagen, dass viele Inhalte der ACM Code of ethics sich in unseren Werten wiederfinden lassen. Natürlich wird nicht jedes explizit beachtet. Aber dennoch ist es interessant, dass sich bei mpunkt die Vorstellung von professionellem Verhalten ziemlich nahe den hier definierten Richtlinien liegt.

Wie lässt sich die Teamarbeit fördern

Teamlearning

Die zweite These von Chamtitigul und Li(2021) sagt, dass die Teamperformance positiv vom Vorgang des Teamlernens beeinflusst wird. Dazu zählen das Beschaffen von Informationen, das Aufarbeiten, das Speichern und Abrufen der Informationen.

Chamtitigul und Li(2021) kommen bei ihrer Forschung zu dem Schluss, dass insbesondere das Aufarbeiten und das Bereitstellen der Informationen den größten Einfluss auf die Teamarbeit nehmen. Dabei erläutern sie, dass das Zusammenführen des einzelnen Wissens stattdessen dazu führt, dass neue Erkenntnisse gewonnen werden. Diese Erkenntnisse wiederum haben den größten Einfluss auf die Qualität des Ergebnisses. Ebenso betonen sie, dass die richtige Speicherung und das darauffolgende Abrufen wichtig sind. Durch die erhöhte Effizienz wird das Wissen schneller weitergegeben und erlernt.

Auch Moe et al. (2023) kamen zu einem ähnlichen Ergebnis. Sie betrachteten, wie die norwegische Firma SB1U ihre Mitarbeiter motiviert. Dabei wurde festgestellt, dass dedizierte Lernzeiten im Tagesalltag einen positiven Einfluss auf die Motivation der Mitarbeiter hat; dies wirkt sich auch positiv auf die Teamarbeit aus. Durch die allozierte Lernzeit haben sich Gruppen gebildet, die sich mit dem gleichen Thema beschäftigt haben.

Ein ähnliches Vorgehen gibt es bereits mit dem Sicherheitsteam. Dieses recherchiert, wie die aktuellen Projekte sicher gemacht werden können. Bis jetzt ist ein wöchentliches Treffen mit einer Stunde angesetzt. Es ist denkbar, es auf 2 Stunden auszuweiten. Falls es sich bewährt und zu guten Ergebnissen führt, ist es möglich, noch weitere Research-Teams ins Leben zu rufen. Die Teams sollten zum Ziel haben, jede Woche neue Themen zu erarbeiten und diese zu dokumentieren. Dabei sollten die Ergebnisse nicht nur Wissen weitergeben, sondern auch konkrete Anleitungen enthalten. Die primäre Stelle zur Archivierung der Resultate ist das Firmenwiki. Weiterhin wäre es denkbar, das Wissen in Form eines Newsletters aufzuarbeiten und an alle zu schicken. Aktuelle Themen, die dafür interessant wären, sind KI, Design oder DevOps. Durch das gemeinsame Recherchieren und Auswerten würde die Teamarbeit fördern. Gleichzeitig wird auch der Wissensschatz erweitert.

Pair Programming

Tkalich et al (2023) beschreiben den Vorgang des Pair Programming wie folgt: Zwei Entwickler arbeiten an einer Maschine an der gleichen Aufgabe. Einer übernimmt die Hauptrolle, der andere äußert Vorschläge und Ideen. Dabei wechseln sich die Entwickler ab. Dadurch stärken die Entwickler ihr Kommunikationsgeschick und sorgen gleichzeitig für eine höhere Codequalität.

Nach Absprachen mit anderen Team Kollegen ist herausgekommen, dass der Wunsch nach Code Reviews besteht. Dies bietet eine Grundlage Reviews auf Basis von Pair Programming einzuführen. Zusammen über den Code zu gehen könnte die Wahrscheinlichkeit erhöhen kritische Fehler zu finden.

Eine Hürde des Pair Programming wird von Moe et al(2023) als die Befürchtung andere zu Stören beschrieben.

Dies wird zusätzlich durch den Stundenkontingent für Projekte verstärkt. Um dies zu umgehen, wäre eine Möglichkeit, bewusst Zeit im Projektmanagement für Pair-Programming einzuräumen. Natürlich fiel deswegen zusätzlich Zeit an, die vermutlich dem Kunden nicht angerechnet werden könnte. Aber die Zeit wird nicht sinnlos verbracht. Zunächst einmal ist es eine exzellente Möglichkeit, die Kommunikationsfähigkeiten der Teilnehmenden zu verbessern. Ferner sorgt das gemeinsame Codereview für ein besseres Verständnis im Team. Insbesondere in Bezug auf das Analysieren der Problemstellung und die Lösungsfindung. Am wichtigsten jedoch ist die Tatsache, dass die investierte Zeit dafür sorgt, die Codequalität zu erhöhen. Daraus folgt eine geringere Zeit bei der Fehlersuche und Behandlung. Pair-Programming wird am Anfang eine kleine Umstellung für alle Beteiligten sein. Ebenso bringt es seine eigenen Nachteile mit. Jedoch überwiegen die Vorteile, sodass es auf einen Versuch ankommt. Allein dieser kann zeigen, ob sich daraus sinnhafte Fortschritte etablieren.

Feedback zu den Vorschlägen

Das Gespräch fand am 26.03.2024 zwischen 9.30 Uhr und 9.54 Uhr via einem Teammeeting statt. Das Gespräch wurde mit einem Senior-Softwareentwickler und Projektleiter, RN, geführt. Es wurde ein offenes Gespräch geführt. Die relevanten Fragen und Vorschläge wurden auf natürliche Weise ins Gespräch integriert.

Im Gespräch wurde zuerst das Thema Pair Programming besprochen, danach Teamlearning.

Das Gespräch wurde mit einer kurzen Zusammenfassung begonnen, wieso das Treffen stattfindet. Es wurde erklärt, dass es sich um das Thema „Wie kann sich die Teamarbeit fördern lassen“. Auf die Frage, wie er die aktuelle Teamarbeit bewerte, antwortete RN Folgendes: „Es ist schwer, darüber eine konkrete Aussage zu treffen. Grundsätzlich arbeiten alle solide im Team. Dadurch, dass die Teams dynamisch formen, gibt es immer neue Teams. Man erkennt, wenn Leute bereits an anderen Projekten zusammengearbeitet haben. Da ist die Teamarbeit von Beginn an besser“.

Im Anschluss wurde das Gesprächsthema in Richtung des ersten Vorschlages gelenkt. Die Überleitung erfolgte, indem der Wunsch von Kollegen geäußert wurde, Code Reviews regelmäßiger und umfangreicher durchzuführen. Der Status quo ist, in erster Linie selber die Code-Reviews durchzuführen und ausgiebig zu testen. Der Wunsch der Kollegen teilt auch RN. Er bekräftigte, dass es ein sehr wichtiger Teil ist und wir eigentlich dafür noch mehr Zeit aufwenden sollten, um sicherzustellen, die bestmögliche Qualität zu liefern. Den Vorschlag, Code Reviews mittels Pair Programming zu realisieren, wurde im Anschluss kritisch diskutiert. Zunächst einmal wurde sich auf darauf geeinigt, dass es eine gute Möglichkeit wäre, die Teamarbeit zu fördern und gleichzeitig die Code-Qualität zu verbessern. Ebenso würde es Sinn ergeben, im Projekt extra Zeit zu kalkulieren, damit es von Anfang an vernünftig umgesetzt werden kann. Jedoch führt es bereits zu dem ersten Problem. Es entsteht ein mehr Aufwand für den Kunden. Die Extrazeit, die im Rahmen des Projektmanagements anfallen würde, muss zuerst von Kunden freigegeben werden. Dadurch werden die Projekte für die Kunden teurer. Bezüglich der aufgewendeten Zeit gibt es noch ein weiteres Problem. Das Vorgehen eignet sich nicht für jede Code-Änderung. Für kleinere Anpassungen und Änderungen, die insgesamt wenig Zeit benötigen, ein Arbeitstag oder weniger, ist der Aufwand für die Organisation und Durchführung unverhältnismäßig groß. Somit bleiben nur die Großprojekte übrig. Ein weiterer Punkt ist, dass die Mitarbeiterpaarung eine große Rolle spielt. Am meisten Sinn ergibt es, wenn ein weniger erfahrener Entwickler mit einem erfahrenen Entwickler zusammenarbeitet. Dadurch wird nicht nur die Teamarbeit gefördert, sondern auch die Wissensweitergabe maximiert. Jedoch entsteht dadurch das Problem, dass die Seniorentwickler deutlich mehr Arbeitsaufwand bekommen. Sie müssen dann nicht nur ihre Projekte vorantreiben, sondern auch noch wertvolle Zeit für Reviews bereitstellen. Ebenso stellt sich die Frage, wie sinnvoll es ist, wenn Seniorentwickler die Commits des anderen durchgehen. Insbesondere, wenn der Code bereits alle Tests erfolgreich bestanden hat.

Zusammengefasst ist es ein sinnvoller Vorschlag, Pair Programming in Verbindung mit Code Review durchzuführen. Die Auffassung, dass dadurch die Teamarbeit und die Codequalität verbessert werden, wird von RN geteilt. Jedoch würde es zu viel Zeit und Ressourcen kosten, es verpflichtend einzuführen. Als Alternative wurde diskutiert, die Kollegen darüber zu unterrichten und bei sinnvollen Themen durchzuführen.

In der zweiten Hälfte des Gespräches ging es um das Team Teamlearning. Hier erfolgte der Einstieg über das Darlegen der Studien. Insbesondere wurde hervorgehoben, dass das gemeinsame Zusammentragen des Wissens den größten Einfluss auf die Teamentwicklung besitzt. Zunächst einmal wurde der Vorschlag, den internen Wissenstransfer zu verstärken, positiv angenommen. Jedoch stellte sich im Verlauf des Gespräches schnell heraus, dass es schwierig wird, explizit Zeit für das Sammeln von Wissen bereitzustellen. Durch die laufenden Projekte sind alle Mitarbeiter gut ausgelastet. Es wäre denkbar, maximal eine Stunde pro Woche bereitzustellen. Jedoch ist eine Stunde wenig Zeit. Insbesondere, wenn es um das Einarbeiten und Dokumentation des Wissens geht. Bei Frameworks und Programmiersprachen ist es noch deutlicher. Um hier

Wissen zu sammeln, ist es wichtig, Projekte umzusetzen. Es gibt bereits Kollegen, die sich neben der bestehenden IT-Sicherheitsgruppe mit der Recherche und dem Aufarbeiten von neuem Wissen beschäftigen. Jedoch kommen sie aufgrund anderer Projekte nicht dazu, es voranzutreiben.

Insgesamt kann Teamlearning nicht eingeführt werden. Jedoch soll in Zukunft wieder mehr Wissensaustausch intern stattfinden. Insbesondere durch regelmäßige Schulungen.

Zum Schluss wurde noch gebrainstormt, wie die Teamorganisation und die Teamarbeit verbessert werden können. Grundlegende Sachen wie klar definierte Kommunikationskanäle, E-Mail, Livechat und Videokonferenzen sind umgesetzt. Es wird ebenfalls bereits hoher Wert auf Teamfähigkeit gelegt. Es wird bereits jede Menge Teambuildingaktivitäten durchgeführt. Im Zusammenhang mit dem Projektmanagement wird ebenfalls bereits viel umgesetzt. Eine schöne Erkenntnis war, dass Kollegen viele Best Practices bereits unterbewusst durchführen.

Literaturverzeichnis

Moe, N. B., Stray, V., Smite, D., & Mikalsen, M. (2023). Attractive Workplaces: What Are Engineers Looking for? *IEEE Software, Software, IEEE, IEEE Softw.*, 40(5), 85-85-93. IEEE Xplore Digital Library. <https://doi.org/10.1109/MS.2023.3276929>

Tkalich, A., Moe, N. B., Andersen, N. H., Stray, V., & Barbala, A. M. (2023). *Pair Programming Practiced in Hybrid Work* (edsee.10304797). 1-1-7. IEEE Xplore Digital Library. <https://doi.org/10.1109/ESEM56168.2023.10304797>

Weining Li & Nattaya Chamtitigul. (2021). The influence of ethical leadership and team learning on team performance in software development projects. *Team Performance Management: An International Journal*, 27, 240-240-259. OpenAIRE.

Whitehead, J. (2007). *Collaboration in Software Engineering: A Roadmap* (edsee.4221622). 214. IEEE Xplore Digital Library. <https://doi.org/10.1109/FOSE.2007.4>