



Fraunhofer Institut
Experimentelles
Software Engineering

Intelligente Assistenz in der Softwareentwicklung 2006: Zusammenfassung der Ergebnisse

Autoren:
Jörg Rech
Eric Ras
Björn Decker

IESE-Report Nr. 045.06/D
Version 1.0
29. Mai 2006

Eine Publikation des Fraunhofer IESE

Das Fraunhofer IESE ist ein Institut der Fraunhofer-Gesellschaft. Das Institut transferiert innovative Software-Entwicklungstechniken, -Methoden und -Werkzeuge in die industrielle Praxis. Es hilft Unternehmen, bedarfsgerechte Software-Kompetenzen aufzubauen und eine wettbewerbsfähige Marktposition zu erlangen.

Das Fraunhofer IESE steht unter der Leitung von
Prof. Dr. Dieter Rombach (geschäftsführend)
Prof. Dr. Peter Liggesmeyer
Fraunhofer-Platz 1
67663 Kaiserslautern

Abstract

Die Entwicklung von Software ist eine komplexe und wissensintensive Aufgabe, bei der viele Informationen durch die Mitarbeiter recherchiert, generiert und weitergegeben werden. Informationen über die Software wie z.B. Entwicklungsdokumente, Quellcode und dokumentierte Erfahrungen werden in den unterschiedlichsten Datenformaten erstellt und in verschiedenen Systemen gespeichert. Der Informationsbedarf eines Mitarbeiters in einem Softwareprojekt hängt von der aktuellen Arbeitstätigkeit, dem Arbeitsdokument und der eigenen Erfahrung ab und soll möglichst schnell und präzise befriedigt werden. Intelligente Assistenzsysteme können an dieser Stelle die wissensintensiven Tätigkeiten wesentlich unterstützen. Sie können dabei in den verschiedensten Formen auftreten und sehr unterschiedliche Arbeitsschritte unterstützen.

Diese explorative Umfrage hatte zum Ziel, zu ermitteln, wo die intelligente Assistenz in der heutigen Werkzeug-Landschaft in Softwareorganisationen am meisten benötigt und welche Art von Assistenz bevorzugt wird. Dieser Bericht fasst die wesentlichen Kerndaten der Umfrage zusammen, an der 135 Teilnehmer aus deutschen Unternehmen und Einrichtungen teilnahmen.

Schlagworte: Intelligent Assistance, Semantic Work Environments, Experience Management

Inhaltsverzeichnis

1	Einleitung	1
1.1	Assistenz im Software Engineering	1
1.1.1	Beispiele für Assistenz	2
1.2	Prinzipien von Assistenzsystemen	3
1.2.1	Anbieten von Assistenz	4
1.2.2	Konstruktion der Assistenz	5
1.2.3	Datenerfassung für Assistenz	6
1.3	Intelligente Assistenz: Gestern, Heute und Morgen	6
2	Die Umfrage	8
2.1	Charakteristiken der Umfrage	8
2.2	Charakteristiken der Teilnehmer	9
3	Ergebnisse der Umfrage	17
3.1	Informationsbedarf bei der Softwareentwicklung	17
3.1.1	Übersicht über Informationsbedürfnisse	17
3.1.2	Grund für die Informationsermittlung	18
3.1.3	Lernspezifische Assistenz	19
3.1.4	Eigenschaften und Umgang mit Information	20
3.2	Durchführung von Softwareprojekten	21
3.3	Softwareentwicklungsinfrastruktur	22
3.4	Bekanntheit und Gestaltungsformen von Assistenz	24
3.4.1	Bekanntheitsgrad und Akzeptanz von Formen der Assistenz	24
3.4.2	Nützlichkeit von Darstellungsformen	26
3.4.3	Vorgehen bei Assistenz	27
3.4.4	Grad von Assistenz	27
3.5	Potenziale für den Einsatz von Assistenz in der Softwareentwicklung	29
3.5.1	Assistenz in welchen Phasen	29
3.5.2	Arbeitsprodukte	30
3.5.3	Bedürfnisse an Intelligenter Assistenz bei der Erstellung von Dokumenten	31
3.5.4	Bedürfnisse an intelligente Assistenz bei der Wiederverwendung von Dokumenten	32
4	Fazit	34
	Literatur	36

1 Einleitung

Die Softwareindustrie ist eine hoch innovative Branche, deren Endprodukte oft in Form von elektronischen Dokumenten (z.B. Quellcode, Anforderungsdokumente, etc.) vorliegen. Um die Erstellung, Modifikation oder Wartung dieser Arbeitsprodukte zu unterstützen, gibt es heutzutage die unterschiedlichsten Arbeitsumgebungen, die teilweise mit mehr oder weniger intelligenten Assistenzsystemen ausgestattet sind.

Forscher im Bereich Intelligente Assistenz erforschen seit den frühen 1970er Jahren solche meist wissensbasierten Systeme, die zu Prototypen entwickelt wurden, aber selten ihren Weg in die industrielle Praxis fanden. Während heutzutage viele Umgebungen implizit oder explizit Ideen aus der Forschung im Bereich Intelligente Assistenz nutzen, sind sich die Benutzer nicht immer deren Existenz bzw. deren Potenzial bewusst.

Das Ziel dieser Umfrage bestand darin, die Konzepte der intelligenten Assistenz zu klären, die Motivation für intelligente Assistenzsysteme zu beschreiben, einige Beispiele für intelligente Assistenz zu besprechen und eine Umfrage über die Einstellung gegenüber intelligenter Assistenz in deutschen Softwareunternehmen sowie über die Nachfrage danach zu präsentieren.

1.1 Assistenz im Software Engineering

Intelligente Assistenz im Software Engineering ist ein relativ alter Forschungsbereich, der dennoch auch heute für Software-Ingenieure von großem Interesse ist. Benutzer von Programmierungs-, Entwurfs-, Anforderungs- oder anderen softwarebezogenen Umgebungen müssen unterstützt werden, da das Produkt ihrer Arbeiten normalerweise sehr komplex ist und von vielen Leuten beeinflusst wird.

Das Kernziel von intelligenter Assistenz im Software Engineering besteht in der Ermöglichung und Optimierung:

- *der Automation* einfacher oder repetitiver Softwareentwicklungsaufgaben wie Kompilation, Erzeugung von Testfällen, Entdeckung von Code-Smells, etc.
- *des Einblicks* in das sich in Entwicklung befindliche System über Querverweise, Anfragemöglichkeiten oder Visualisierung,

- *der Interaktion mit und Verhandlungen* zwischen den Beteiligten (z.B. Benutzer(n) bzw. dem/den assistierenden (Sub-)System(en), um kooperatives Arbeiten zu unterstützen oder die Assistenz zu erklären.

1.1.1 Beispiele für Assistenz

Es existiert heutzutage eine Vielzahl von Assistenzsystemen, die für den Software-Ingenieur mehr oder weniger nützlich sind. Im Folgenden findet sich eine Liste mehrerer beachtenswerter Systeme, die entweder allgemein bekannt oder sehr nützlich sind:

- Eines der bekanntesten Systeme ist wahrscheinlich die Assistenz in den Office-Programmen von Microsoft (z.B. Clippy, Einstein oder Merlin). Sie unterstützt den Benutzer bei der Arbeit mit dem Office-Werkzeug und erklärt Funktionen oder Fehlermeldungen. Obwohl sie vom erfahrenen Benutzer typischerweise als nicht sehr hilfreich angesehen wird, so mag sie dem Neuling, der über kein anständiges Handbuch verfügt, doch eine Hilfe sein.
- In der eclipse IDE (und vielen ähnlichen IDEs) wurden viele Assistenzsysteme zur Auto-kompilation implementiert, ebenso Wizards, die ein Skelett eines Softwaresystems erzeugen (z.B. eclipse Plugins) und Features, die automatisch Refactorings auf ein ausgewähltes Element anwenden. Ferner wird hier die Vervollständigung von Code geboten (d.h. es erscheint eine Liste möglicher Methoden einer Klasse sowie die entsprechende JavaDoc), die Erzeugung von kleinen Code-Fragmenten (z.B. Getter-/Setter-Methoden) und die Auswirkungsanalyse auf Code-Ebene (z.B. durch Aufzeigen von Problemen, wenn sich die Signatur einer Methode ändert).
- Entwurfs-Kritiksysteme wie sie z.B. in argoUML integriert wurden, enthalten eng integrierte intelligente Assistenzsysteme, die Fehler oder Unvollständigkeiten im Entwurf oder falsche Zuordnungen von Schnittstellen aufzeigen. Die Kritik an dem Softwareprodukt (z.B. am Entwurf) regt dabei zum Nachdenken und zur Reflektion über das Produkt an. Dabei können auch alternative Entwürfe vorgeschlagen (z.B. aufgrund von Problemen bei der Generierung von Code aus dem Entwurf) oder weiterführende Ratschläge gegeben (z.B. einen Kollegen, der ein ähnliches Problem gemeldet hat) werden.
- Ein weiteres Beispiel für Assistenz ist die .NET Sprachumgebung von Microsoft für „Spec#“. Dieses Programmiersystem enthält den statischen Programmverifizierer „Boogie“. Er unterstützt den Entwickler bei der formellen Spezifizierung des Quellcodes und prüft die Korrektheit in späteren Änderungsaktivitäten.

Natürlich stellt dies nur eine kleine Auswahl von Beispielen für bestehende Assistenzsysteme in heutigen Softwareentwicklungswerkzeugen dar.

1.2 Prinzipien von Assistenzsystemen

Obwohl das Gebiet möglicher Assistenzsysteme sehr groß ist, kann man Gruppen von Assistenzsystemen durch ihre Eigenschaften bezüglich des Informationsangebots und der Datenextrahierung unterscheiden. Wie in Abbildung 1 dargestellt, wird Assistenz in unterschiedlichen Werkzeugen (ganz rechts) integriert. Die Art der Assistenz kann den Benutzer dabei beim Lernen, Handeln (z.B. Programmieren), Ausarbeiten (z.B. Vervollständigen), Wiederverwenden oder bei anderen Tätigkeiten unterstützen. Um diese unterstützenden Informationen zu generieren, werden mittels Inferenztechniken Daten und Informationen über den Kontext (z.B. den Benutzer oder ähnliche Dokumente) verwendet.

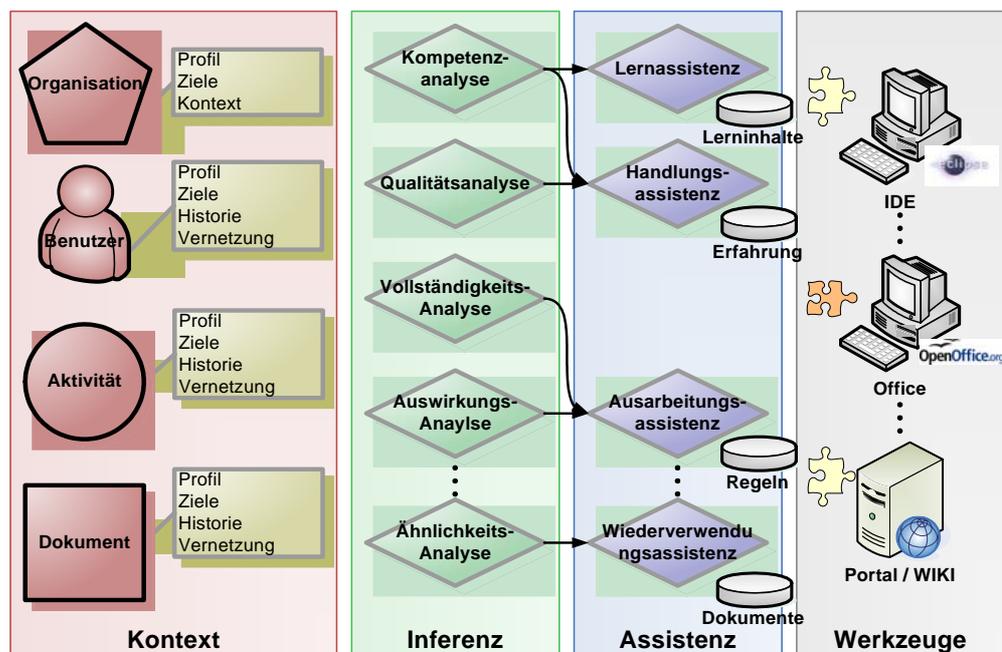


Abbildung 1: Konzeptionierung von Assistenzsystemen im Software Engineering

Der allgemeine Datenfluss in einem Assistenzsystem ist in Abbildung 2 dargestellt. Informationen über den Inhalt der Arbeit (das vorliegende Dokument), den Prozess (d.h. die derzeitige Aktivität), den Status des Werkzeugs, sowie Informationen über den Benutzer werden gesammelt und in einem einheitlichen, maschinenlesbaren Format zur Verfügung gestellt. Diese Daten werden von den Kernassistentenalgorithmus dazu verwendet, kontextspezifische Informationen zu produzieren, die dem Benutzer angeboten werden.

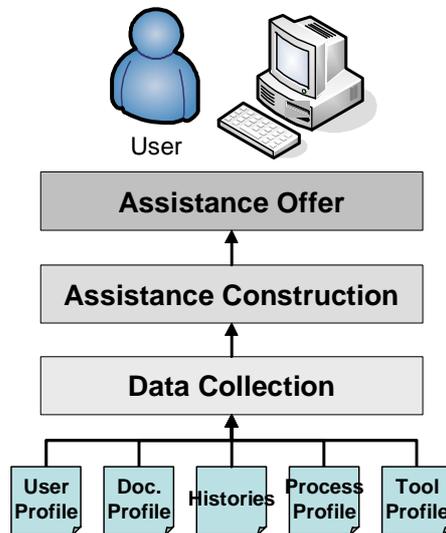


Abbildung 2: Datenfluss bei Assistenzsystemen

1.2.1 Anbieten von Assistenz

Die dem Benutzer angebotene Assistenz basiert auf dem Konstruktionsalgorithmus. Während das Ergebnis dieses Algorithmus' festliegt, kann die Präsentationmethode durch die folgenden Eigenschaften differenziert werden:

- *Wann soll assistiert werden:* Wenn jeder Klick des Benutzers auf eine potenzielle Aktion hinweist, so stellt sich die Frage, wann der Benutzer Assistenz erhalten sollte? Assistenz kann proaktiv erzeugt werden (d.h. vor einer Aktion), während einer Aktion oder nach einer Aktion, und kann nach Bedarf oder auf Anfrage präsentiert werden.
- *Wie soll assistiert werden (Form der Medien):* Da moderne Computer oft Multimedia-Arbeitsumgebungen darstellen, kann die Form des Mediums, in dem die Assistenz angeboten wird, unterschiedlich sein. Zurzeit kann Assistenz in Textversion angeboten werden, visuell (d.h. als Figur oder Animation), akustisch (z.B. als Audio-Kommentar) oder als Video.
- *Wo soll assistiert werden:* Die Informationen, die das Assistenzsystem anbietet, können in Tooltips eingebettet sein, in Pop-ups, Tabellen, speziellen Sound-Effekten (z.B. Beeps), Blinkeffekten, in der Seitenleiste eines Dokuments oder an speziell markierten Orten (z.B. Sichten in der eclipse IDE). Ferner können sie mit einem aktiven Werkzeug, mit einem speziellen Werkzeug eines Drittanbieters oder im Betriebssystem selbst angezeigt werden.
- *Warum soll assistiert werden:* Was ist überhaupt die Begründung für die Assistenz? Es könnte eine Kompetenzlücke im Profil des Benutzers existieren, es könnte ein komplexer Prozessschritt anstehen, ein neues Tool-Feature

könnte bei einem Update installiert worden sein, oder es wird gerade ein Algorithmus entwickelt, der typischerweise fehleranfällig ist.

Während ein Assistenzwerkzeug mindestens eine festgelegte Eigenschaft besitzt, kann es auch möglich sein, dass das System selbst entscheidet, welche Informationen dem Benutzer wann und wie präsentiert werden sollten. Diese Entscheidung könnte beeinflusst werden durch Informationen darüber, warum und für wen Assistenz generiert wurde.

1.2.2 Konstruktion der Assistenz

Assistenz gibt es in allen Formen und Farben – von der Erläuterung durch ein Werkzeug bis zu vollständigen E-Learning-Angeboten. Während die Funktionalität oder das Ergebnis als solches schwer zu klassifizieren ist, kann man die Assistenzalgorithmen wie folgt charakterisieren:

- *Assistenz für wen?* Wem soll mit der konstruierten Information geholfen werden? Je nach Benutzerprofil müssen die Ergebnisse personalisiert oder dem Erfahrungsniveau angepasst werden.
- *Assistenz zu was?* Welche Art von Objekt soll mit Assistenzinformationen angereichert werden? Ist es ein Anforderungsdokument, ein Testwerkzeug, ein Prozess- oder Aktivitätsmodell, allgemeines Hintergrundwissen, Informationen über Experten, etc.?
- *Assistenz bei welchem Prozess?* Der Prozess bzw. die Aktivität, in der der Benutzer derzeit involviert ist, kann zu einer besonderen Anforderung für Assistenz führen. Beispielsweise hat ein Programmierer, der Software entwickelt (und am Quellcode arbeitet), andere Anforderungen an Assistenz als ein Tester oder Inspektor, der sich den gleichen Quellcode anschaut.
- *Assistenz in welcher Werkzeugumgebung?* Wenn keine Information über den Prozess zur Verfügung steht, dann kann auch der Kontext der Werkzeugumgebung dazu verwendet werden, die Assistenz zu optimieren. Ein laufendes Testwerkzeug kann ein Hinweis auf den derzeitigen Prozess sein; die Verwendung eines Texteditors (z.B. OpenOffice) für ein Anforderungsdokument kann auf mangelndes Wissen oder fehlende RE-Werkzeuge hindeuten; und eine aktualisierte Kodierungsumgebung (z.B. eine neue Version der eclipse IDE mit mehreren neuen Plugins) kann ein Zeichen für neue Funktionen sein, die dem Benutzer nicht bekannt sind.

1.2.3 Datenerfassung für Assistenz

Um Informationen zu generieren, die dem Benutzer helfen, müssen in der Konstruktionsphase Daten aus mehreren Quellen bearbeitet werden. Die Eigenschaften, die die verschiedenen Ansätze unterscheiden, sind:

- *Wo sollen Daten extrahiert und vorbearbeitet werden?* Es gibt mehrere Quellen mit Informationen, die für die Konstruktion von Informationen für Assistenz nützlich sein können. Zumindest kann man Daten vom Benutzer, vom Dokument, vom Prozess und von den Werkzeugbeschreibungen, die sich in dem aktiven Werkzeug befinden (z.B. im Inhalt des Dokuments), oder von externen Datenbanken (z.B. LDAP-Server für Benutzerdaten) extrahieren.
- *Wann sollen Daten extrahiert und vorbearbeitet werden?* Der optimale Zeitpunkt für die Extraktion hängt von der Art der Assistenz und der Variabilität der Daten ab. Ein Benutzerprofil bleibt wahrscheinlich im Verlauf einiger Stunden gleich, wohingegen ein Dokument vielleicht dramatisch an Inhalt zunimmt. Man kann die Daten während der Arbeit kontinuierlich extrahieren, dies aktionsgetrieben tun oder nach Bedarf. Außerdem kann die Extraktion sogar ausgesetzt werden, je nachdem, welche Assistenzalgorithmen zum betreffenden Zeitpunkt aktiv sind.
- *Wie sollen Daten extrahiert und vorbearbeitet werden?* Die Daten aus verschiedenen Datenquellen müssen extrahiert und vorbearbeitet werden. Die Extraktion hängt davon ab, wo sich die Daten befinden, sowie von den vorhandenen Schnittstellen oder Abfragesprachen. Die Vorbearbeitung bietet ähnlich dem ETL-Prozess beim Data Warehousing mehrere Techniken für die Vereinigung oder Diskretisierung von Daten zur Schaffung homogener Daten für die nächste Phase. Außerdem geht es darum, wie das System mit fehlenden, unvollständigen oder inkorrekten Daten umgeht. Einige Systeme ignorieren, korrigieren oder filtern die bestehenden Daten, während andere sie blind verwenden.

1.3 Intelligente Assistenz: Gestern, Heute und Morgen

Die Wurzeln der intelligenten Assistenz im Software Engineering liegen in den frühen siebziger Jahren des letzten Jahrhunderts, als Terry Winograd über intelligente Assistenz für Programmierer schrieb ⁰ und Teitelman den "Programmer's assistant" beschrieb ⁰. Sie schufen die Bühne für Systeme, die Entwickler durch automatisierte Aufgaben unterstützen sollen – davon sind heutzutage einige allgegenwärtig (z.B. die Undo-/Redo-Funktionalität), andere nicht (z.B. automatisierte Parameterchecks). In den achtziger Jahren wurden Umgebungen wie Marvel ⁰ und "The Programmer's Apprentice" ⁰ für intelligente Assistenz entwickelt, die den Entwicklungsprozess modellierten, die Software automatisch im Hintergrund verarbeiteten (z.B. Kompilierungs- und Analyseaufgaben)

und dem Benutzer zusätzliche Assistenz zur Verfügung stellten, wenn bestimmte Regeln griffen. Die neunziger Jahre brachten Systeme für das Requirements Engineering hervor, wie z.B. "The Requirement's Apprentice" [0], oder für den Entwurf von Software, wie z.B. der "Design Apprentice" [0], der "The Software Architect's Assistant" [0] oder argoUML und seine Entwurfskritiker [0]. Sie halfen bei Aufgaben wie schrittweiser Verfeinerung, automatisiertem Layout, Konsistenzüberprüfungen oder Codegenerierung.

Heutzutage ist Assistenz weit verbreitet in IDEs wie eclipse, IntelliJ oder Visual Studio .NET, wo Informationen angeboten werden über den Kompilierungsprozess (z.B. Warnungen und Fehler), die Korrektheit einer Funktion (z.B. Verifikation von Invarianten in Spec# [0]) oder durch das Anbieten einsetzbarer Refactorings. Weitere Assistenzfeatures sind Syntax-Highlighting, Kontextassistenz (z.B. kontextabhängige Hilfe oder Vervollständigung von Code), Wizards, die lauffähige Skelette von Anwendungen erzeugen, automatisierte Codeinspektionen oder Quick-Fixes (d.h. kleine, typische Aktionen, die ein aktuelles Problem lösen würden), um nur einige zu nennen.

In Zukunft werden wir vielleicht noch mehr intelligente Assistenz in Umgebungen für Programmierung, Wartung, Entwurf oder Requirements Engineering sehen, die proaktive Unterstützung für Aufgaben wie Wiederverwendung von Software, Lernen nach Bedarf oder automatisierte Konfiguration von Produktvarianten leisten und die Lücke zwischen den Phasen der Softwareentwicklung überbrücken. Die modellgetriebene Architektur (MDA) stellt eine Art von "Programmierungsassistenz" dar, die Modelle automatisch in Quellcode übersetzt – wobei sie die notwendige Expertise bezüglich Programmieren und Softwarequalitäten (d.h. Quellcode) in sich einschließt. Außerdem könnte eine qualitätsorientierte Assistenz, die ein System bearbeitet, um eine bestimmte Softwarequalität wie Wartbarkeit oder Performanz (siehe ISO 9126) zu betonen, die Kosten und den Zeitaufwand für die Entwicklung von Software und Systemen reduzieren.

2 Die Umfrage

Drei Hauptmotivationen liegen der Umfrage zur intelligenten Assistenz im Software Engineering zugrunde. Zuerst sollte der Bedarf von SE-Praktikern in Bezug auf ihre SE-Umgebung und die im Rahmen der Softwareentwicklung benötigten Informationen ermittelt werden. Zweitens sollte der aktuelle Kenntnisstand dieser Praktiker über Assistenz erhoben und ein Meinungsbild über die aktuell verfügbaren Formen von Assistenz erfasst werden. Drittens sollten die Präferenzen in Bezug auf Assistenz ermittelt, sowie aktuelle und ein Überblick über mögliche Einsatzzwecke von Assistenz in der Softwareentwicklung gewonnen werden.

2.1 Charakteristiken der Umfrage

Um die derzeitigen Ansichten der Praktiker zu erfassen, wurde eine Online-Umfrage mit deutschen Unternehmen durchgeführt. Insgesamt wurden von 460 Personen befragt, von denen 135 die Umfrage beendeten, d.h. welche die letzte Seite aufrufen. Die Studie, die zwischen dem 2. März und dem 9. April 2006 durchgeführt wurde, lieferte quantitative Daten über die derzeitige Praxis der Informationsbeschaffung und die Notwendigkeit von Assistenz in Software-Engineering-Projekten.

Die Umfrage bestand aus acht Seiten mit insgesamt 38 Fragen (inkl. Kontextinformationen), die durchschnittlich 30 Minuten zur Beantwortung erforderten. Als Fragetyp kamen überwiegend Multiple-Choice-Fragen zum Einsatz, da diese in der Regel leichter zu beantworten sind. Die beiden häufigsten Arten von Multiple-Choice-Fragen waren "wähle alle zutreffenden Antworten" und "wähle die beste Antwort". Um unerwartete Antworten zuzulassen, besaßen die meisten Fragen eine Auswahlmöglichkeit "anderes" mit zusätzlichem Raum für eigene Kommentare. Es wurde festgestellt, dass dies besonders nützlich ist, wenn der Umfang der Antworten zu groß sein könnte (z.B. Werkzeuge für Anforderungsanalyse) oder wenn Zweifel darüber bestanden, ob die gegebenen Antworten vollständig sind. Dies machte es möglich, mithilfe der Umfrage einige bis dahin unbekannte Daten zu erfassen.

Zur Entwicklung der Umfrage wurde das Werkzeug OPST der Firma Globalpark (<http://www.globalpark.de/>) eingesetzt.

Die Adressen der Teilnehmer wurden zum einen über im Softwarebereich operierende Firmen aus der Online-Datenbank von Hoppenstedt (<http://www.hoppenstedt.de/>) ermittelt. Außerdem wurden, über deutsche

Mailinglisten zu Softwareentwicklungsthemen, einige Softwareentwickler angesprochen. Dabei wurden alle zwei Wochen Erinnerungen an die Mailinglisten geschickt, wodurch auch die Abonnenten über die Umfrage informiert wurden.

Bei der Auswertung der Umfrage wurden die Datensätze von den Teilnehmern berücksichtigt, welche die Umfrage beendet hatten. Weiterhin wurden Datensätze entfernt, welche sich durch die Umfrage „hindurchgeklickt“ hatten. Dies wurde anhand der Antworten und einer Bearbeitungszeit von weniger als 5 Minuten bestimmt. Nach dieser Bereinigung der Datensätze wurden die im Folgenden präsentierten Ergebnisse ermittelt.

2.2 Charakteristiken der Teilnehmer

Im folgenden Abschnitt werden die Teilnehmer basierend auf ihren Angaben charakterisiert:

- *Typ von Unternehmen*: Diese Charakteristik gibt an, aus welchem Unternehmensumfeld die Teilnehmer stammen.
- *Unternehmensgröße*: Die Größe von Unternehmen wurde über die Anzahl ihrer Mitarbeiter bestimmt. Dadurch kann ermittelt werden, ob bestimmte Formen der Assistenz von der Unternehmensgröße abhängen.
- *Zielgruppe*: Die Art der Zielgruppe der entwickelten Software hat Einflüsse auf die Softwareentwicklungsprozesse (z.B. auf die geforderte Qualität und den Grad an Wiederverwendung).
- *Branche*: Auch die Branche der jeweiligen Softwareentwickelnden Organisation hat Einfluss auf die Abläufe und die Anforderungen an die Qualität der Software. So werden bei der Entwicklung von Spielsoftware andere Qualitätsanforderungen an die Software gestellt als im medizinischen Bereich.
- *Art der Software*: Diese Charakteristik beschreibt die Anwendungsklassen der entwickelten Software. Auch diese Charakteristik hat Einfluss auf die jeweiligen Softwareprozesse.
- *Angaben zur Person*: Hier wurde die Position des Teilnehmers im Unternehmen abgefragt. Dadurch können Rückschlüsse darüber gezogen werden, ob unterschiedliche Rollen unterschiedliche Ansprüche an die intelligente Assistenz stellen.
- *Jahre Berufserfahrung*: Die Jahre an Berufserfahrung wurden ermittelt, um zu erkennen ob es Unterschiede bei der Akzeptanz und Art der intelligenten Unterstützung zwischen unerfahrenen und erfahrenen Teilnehmern gibt.

In Abbildung 3 wird die Verteilung der Teilnehmer in Bezug auf ihren Organisationstyp dargestellt. Dabei gibt die erste Zahl die Anzahl der Teilnehmer, die zweite Zahl den prozentualen Anteil in der jeweiligen Kategorie an. Den über-

wiegenden Anteil bilden Teilnehmer aus Unternehmen mit 67 Prozent. Universitäten und Forschungseinrichtungen umfassen 6 Prozent der Teilnehmer. Aus dem Open-Source-Umfeld stammen 4 Prozent der Teilnehmer. Acht Prozent sind Freiberufler. Ein Prozent gaben andere Typen von Unternehmen an, während 14 % keine Angaben zu ihrem Typ von Unternehmen machten. Aufgrund dieser Verteilung, insbesondere durch den hohen Anteil von Unternehmen, ist davon auszugehen, dass die Teilnehmer über eine hohe praktische Berufserfahrung verfügen.

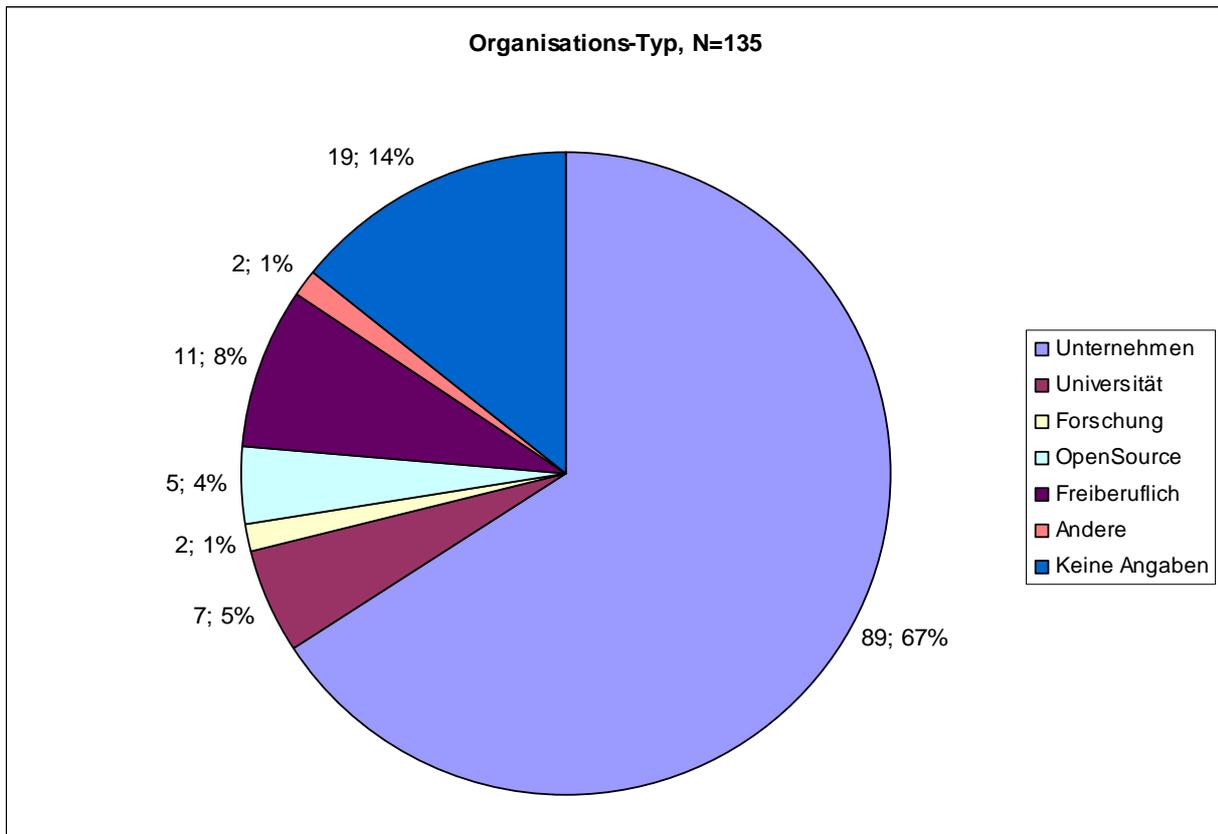


Abbildung 3: Für welchen Typ von Unternehmen oder Gesellschaft arbeiten Sie?

In Abbildung 4 wird die Größe der jeweiligen Organisation mittels der Anzahl von Mitarbeitern dargestellt. Aus Kleinstunternehmen mit 1-9 Mitarbeitern stammen 27 Prozent der Teilnehmer. Kleinunternehmen mit 10-49 Mitarbeitern gehörten 33 Prozent der Teilnehmer an. Aus mittleren Unternehmen mit 50-250 Mitarbeitern nahmen 18% der Befragten teil. Großunternehmen mit 250 und mehr Mitarbeitern bildeten 8 Prozent der Teilnehmer. 13 Prozent der Teilnehmer machten keine Angaben zur Größe ihres Unternehmens.

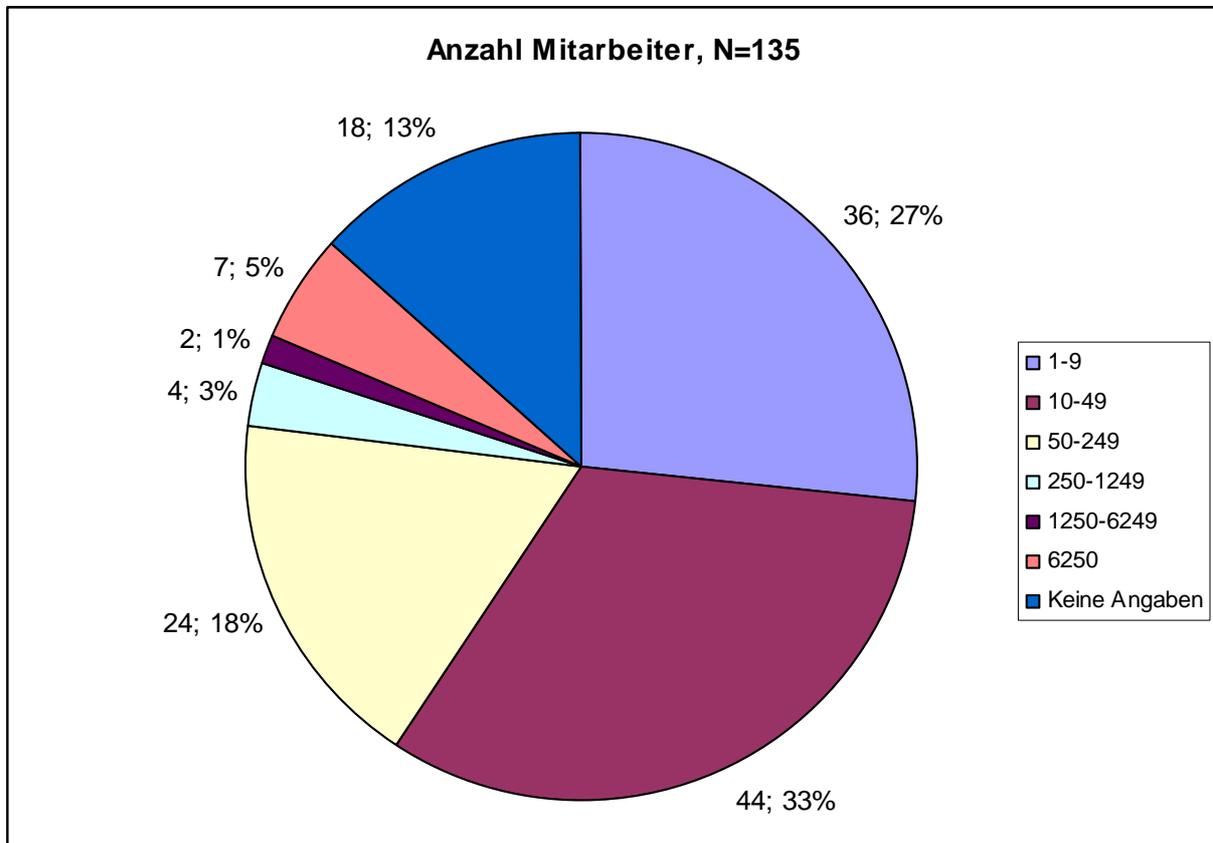


Abbildung 4: Wie viele Mitarbeiter arbeiten in ihrem Unternehmen oder ihrem OSS-Projekt?

Damit stammt der überwiegende Teil der Teilnehmer aus Unternehmen mit überschaubarer Größe, bei denen zu erwarten ist, dass tendenziell der persönliche Austausch im Rahmen der Softwareentwicklung überwiegt.

Abbildung 5 gibt einen Überblick über die Zielgruppe an, für welche die jeweilige Software entwickelt wird. Hier überwiegt die Individualsoftware mit 77 Teilnehmern, die sich diesem Gebiet zugeordnet haben, gefolgt von der Eigenentwicklung mit 58 Teilnehmern. Beiden Zielgruppen ist gemein, dass hier eher Frameworks zum Einsatz kommen, die im Rahmen der jeweiligen Projekte angepasst werden. Standardsoftware mit 51 Teilnehmern liegt auf Platz drei. Hier steht das Versionsmanagement der jeweils entwickelten Software im Vordergrund. Prototypische Entwicklung ist die Zielgruppe von 27 Teilnehmern. Hier steht die schnelle Umsetzung von Ideen und Konzepten im Vordergrund.

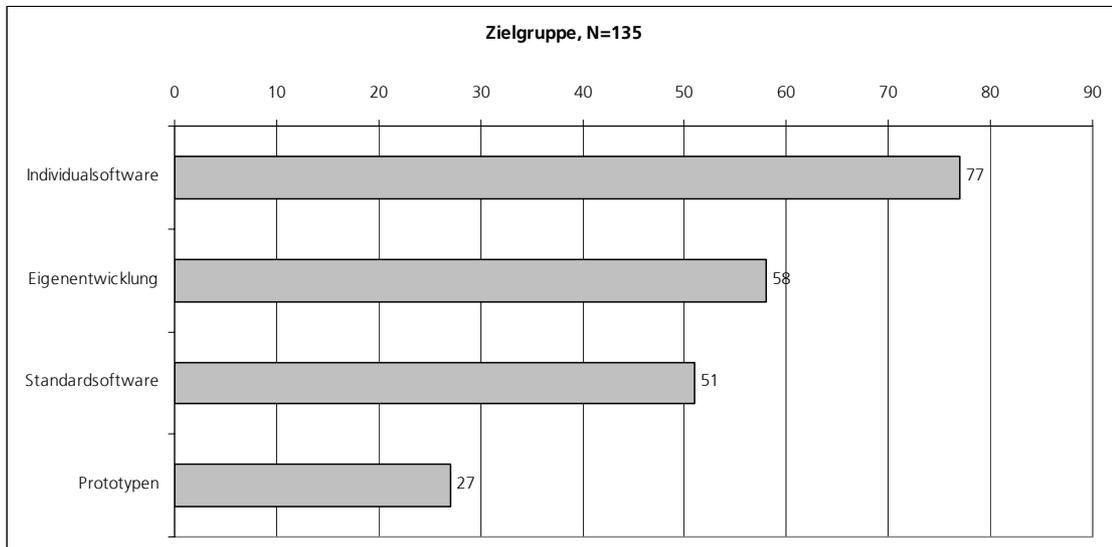


Abbildung 5: Für welche Zielgruppe entwickelt ihr Unternehmen Software?

Abbildung 6 stellt die Branche dar, für welche die jeweilige Software entwickelt wird. Da hier insgesamt 19 Kategorien abgefragt wurden, werden die Ergebnisse dieser Frage zusammengefasst. Mit der Informations- und Kommunikationstechnologie (38 Antworten) auf Platz eins und Softwareentwicklung und IT-Systemhersteller (33 Antworten) auf Platz zwei sind zwei Branchen vertreten, welche der primären Software-Industrie hinzuzurechnen sind. Bei dieser primären Software-Industrie ist die Software das Hauptergebnis der jeweiligen Entwicklung. Bei der sekundären Softwareentwicklung fließt die Software in das jeweilige Produkt ein, ist aber für den Endnutzer des jeweiligen Produktes „unsichtbar“. Ein Beispiel sind die Branchen auf Platz drei und vier: Finanzdienstleistungen (33 Antworten) und Fahrzeugbau (25 Antworten). Hier gilt es, die Softwareentwicklungsprozesse mit den Gesamtentwicklungsprozessen zu koordinieren.

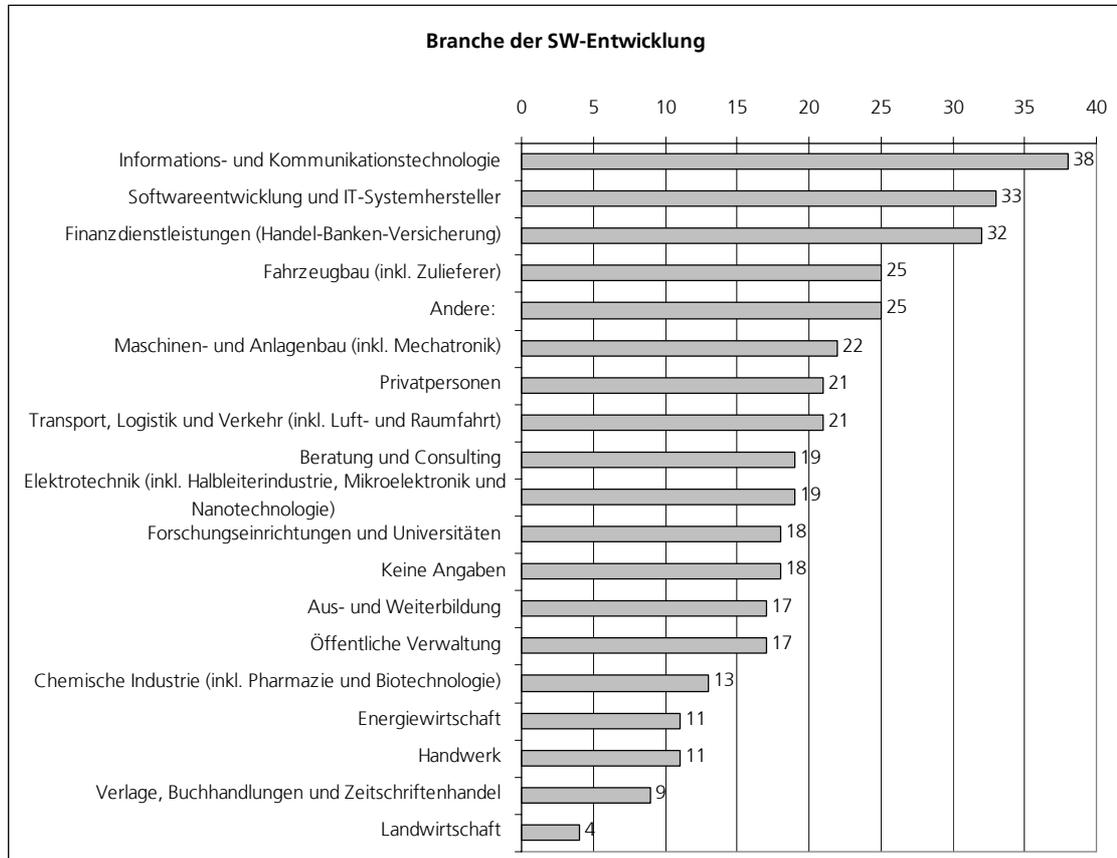


Abbildung 6: Für welche Branche entwickelt ihr Unternehmen Software?

In Abbildung 7 wird die Verteilung der Teilnehmer in Bezug auf die Art der entwickelten Software angegeben. Auch hier werden nicht alle Ergebnisse detailliert dargestellt, sondern nur eine Übersicht gegeben. Dabei sind auf Platz eins und zwei mit Informations- und Datenbanksystemen (70 Antworten) und Internet-basierter Software (62 Antworten) zwei Branchen vertreten, welche der primären Software-Industrie hinzuzurechnen sind. Auf Platz drei und vier liegen mit einigem Abstand betriebswirtschaftliche Systeme (38 Antworten) und Kommunikationssysteme (24 Antworten). Der Bereich Softwareentwicklungsumgebungen, in dem intelligente Assistenz in der Softwareentwicklung umgesetzt wird, entfielen 6 Antworten. Durch diesen verhältnismäßig geringen Anteil dieser Gruppe ist zu erwarten, dass die Anforderungen an intelligente Assistenz aus Anwendersicht beschrieben werden.

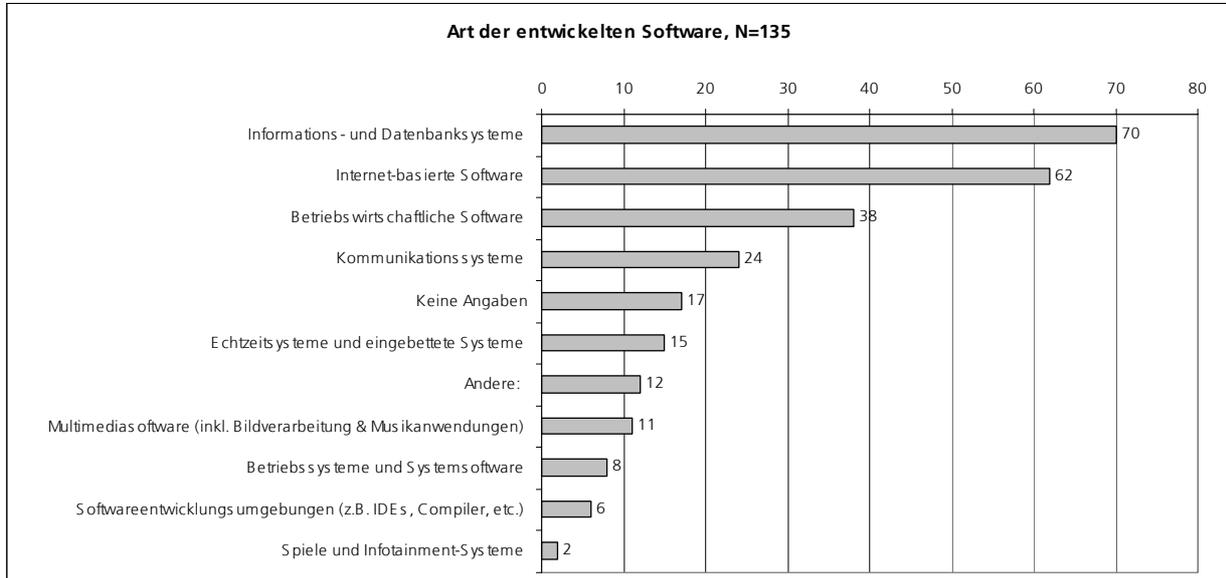


Abbildung 7: Welche Art von Software wird bei Ihnen am häufigsten entwickelt?

Die verbleibenden zwei Charakteristika beziehen sich auf den Teilnehmer selbst. Hier wurde zum einen nach der Position im Unternehmen gefragt, zum anderen wurden zur Bestimmung der Berufserfahrung die Jahre in der Softwareentwicklung erfasst.

Wie in Abbildung 8 dargestellt, sind 42 Prozent der Teilnehmer auf der operativen Ebene als Projektmanager (16 Prozent) oder Mitarbeiter (26 Prozent) tätig. Für diese Personengruppe ist zu erwarten, dass sie direkt mit Software-Engineering-Artefakten wie Code oder Anforderungsdokumenten arbeiten. Als Geschäftsführer oder im mittleren Management sind insgesamt 38 Prozent der Teilnehmer tätig und damit dem organisationalen Management zuzurechnen. Hier ist zu erwarten, dass eher Software-Engineering-Artefakte aus den frühen Phasen des SW-Entwicklungsprozesses relevant sind. Andere Positionen nahmen 4 Prozent der Teilnehmer ein, 13 Prozent machten keine Angaben zu ihrer Position.

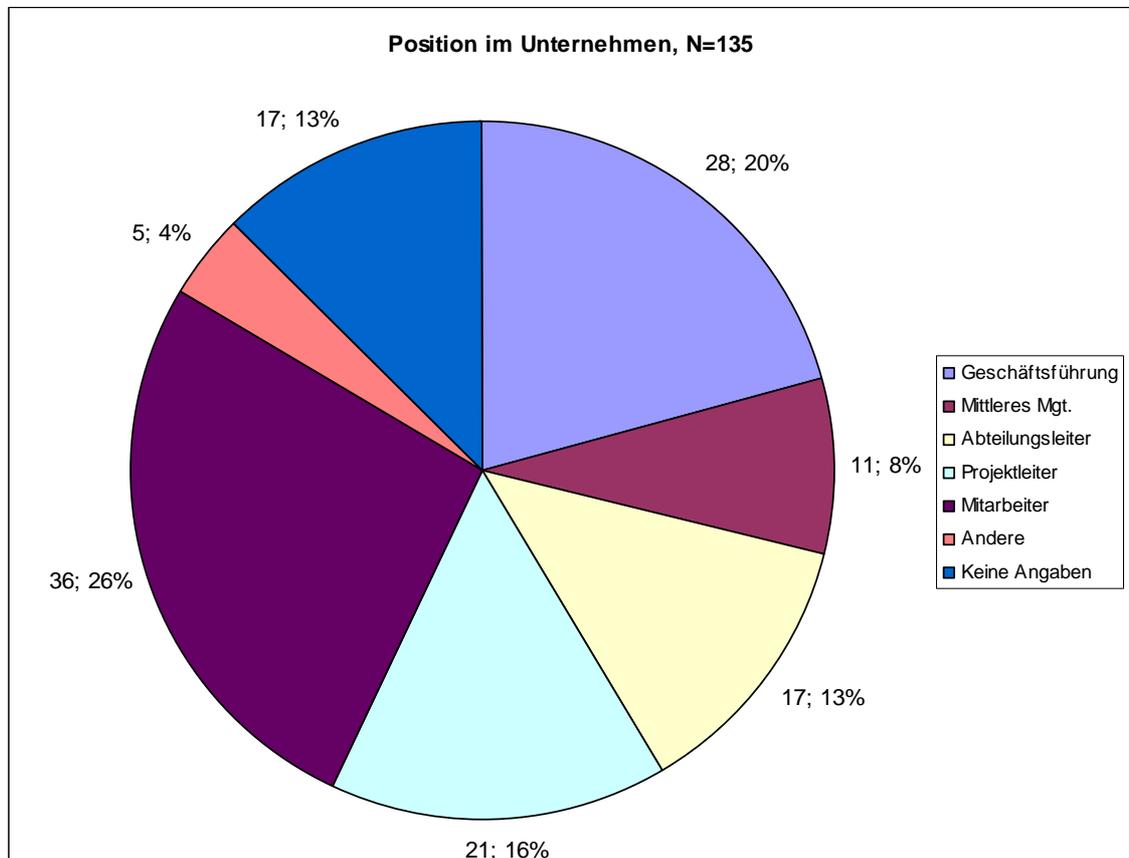


Abbildung 8: Welche Position haben Sie in ihrem Unternehmen (oder OSS-Projekt)?

Abbildung 9 beschreibt die Verteilung der Teilnehmer nach Berufsjahren und gibt damit einen Anhaltspunkt für deren Berufserfahrung. Ein bis fünf Jahre sind 21 Prozent der Teilnehmer in der Softwareentwicklung tätig. Damit ist ein recht hoher Anteil von „Berufseinsteigern“ in der Stichprobe enthalten. Diese Gruppe kann dabei insbesondere solche Formen von Assistenz bewerten, die den Erwerb von Kompetenzen unterstützen. Mit 28 Prozent bilden die Teilnehmer mit 6-10 Jahren Berufserfahrung den größten Anteil an den Teilnehmern. Mehr als 10 Jahre Berufserfahrung haben 34 Prozent der Teilnehmer, während 14 Prozent keine Angaben machten. Damit verfügen die Teilnehmer in der Regel über eine längerfristige Berufserfahrung. Aufgrund dieser Berufserfahrung kann dieser Personenkreis Formen von intelligenter Assistenz einschätzen, die nicht auf den Erwerb von Kompetenzen abzielen.

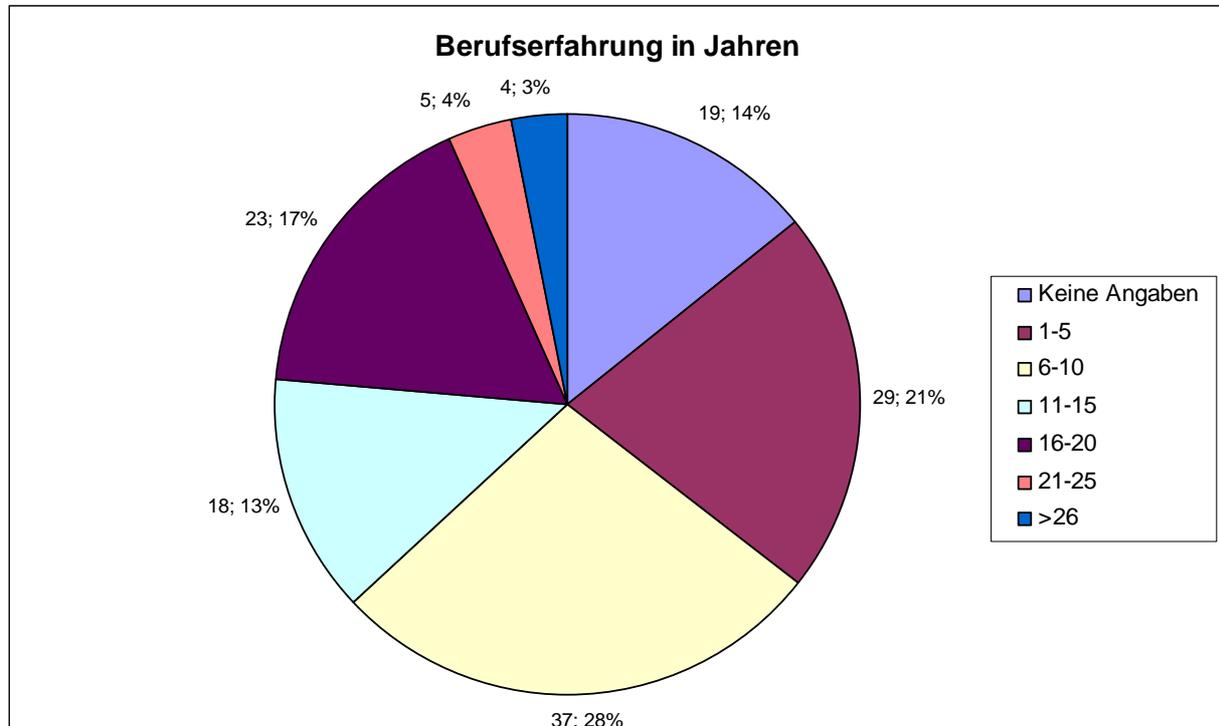


Abbildung 9: Wie lange sind Sie bereits in der Softwareentwicklung tätig?

Im Rahmen dieser Kurzfassung werden in der folgenden Darstellung der Ergebnisse nur die Ergebnisse für die Grundgesamtheit präsentiert. Die Darstellung der Ergebnisse nach den oben genannten Kategorien finden sich in der Langfassung dieser Studie.

3 Ergebnisse der Umfrage

In diesem Abschnitt werden die Ergebnisse – auf die Grundgesamtheit der Teilnehmer bezogen – präsentiert. Eine nach den in Abschnitt 2.2 genannten, Charakteristiken der Teilnehmer untergliederte Untersuchung findet sich in der Langfassung dieser Studie.

Die Präsentation der Ergebnisse ist in die folgenden Unterabschnitte unterteilt:

- Der Informationsbedarf bei der Softwareentwicklung in Abschnitt 3.1
- Die Durchführung von Softwareprojekten in Abschnitt 3.2
- Die Softwareentwicklungsinfrastruktur in Abschnitt 3.3
- Die Bekanntheit und Gestaltungsformen von Assistenz in Abschnitt 3.4
- Die Potenziale für den Einsatz von Assistenz in der Softwareentwicklung in Abschnitt 3.5

Die Langfassung der Studie enthält darüber hinaus noch folgende Punkte:

- Eine Übersicht über die Häufigkeit der Nutzung verschiedener Informationsquellen.
- Bei der Untersuchung der Potenziale intelligenter Assistenz wird weiter untersucht, in welchen Kombinationen Bedürfnisse auftreten.
- Die Untersuchung des Marktpotenzials von intelligenter Assistenz. Die entsprechenden Fragen beziehen sich auf die geplante und realisierte Integration von intelligenter Assistenz sowie das Budget für intelligente Assistenz.

3.1 Informationsbedarf bei der Softwareentwicklung

Dieser Abschnitt gibt einen Überblick über die Informationsbedürfnisse im Rahmen der Softwareentwicklung und über die Motivation für die jeweilige Ermittlung von Informationen.

3.1.1 Übersicht über Informationsbedürfnisse

Bei der Frage "Welche Arten von Informationen benötigen Sie bei ihrer Arbeit?" gaben 83 Prozent der Teilnehmer an, dass sie „sehr häufig“ und „häufig“ wieder verwendbare Dokumente (z.B. Quelltext, Anforderungen, Testfälle) benötigen (siehe Abbildung 10). 76 Prozent dieser Antworten entfielen auf

Vorlagen und Beispiele. Informationen in Form von Kursen wurden nur von 15 Prozent „sehr häufig“ und „häufig“ benötigt, während 39 Prozent diese Information selten benötigten. Die geringe Nachfrage nach Kursen ist in einem innovativen Umfeld wie dem Software Engineering erstaunlich. Eine mögliche Erklärung könnte sein, dass aufgrund des Termindrucks und der Vielfalt ähnlich gearteter Technologien die Aneignung neuen Wissens im normalen Tagesgeschäft bedarfsgetrieben erfolgt.

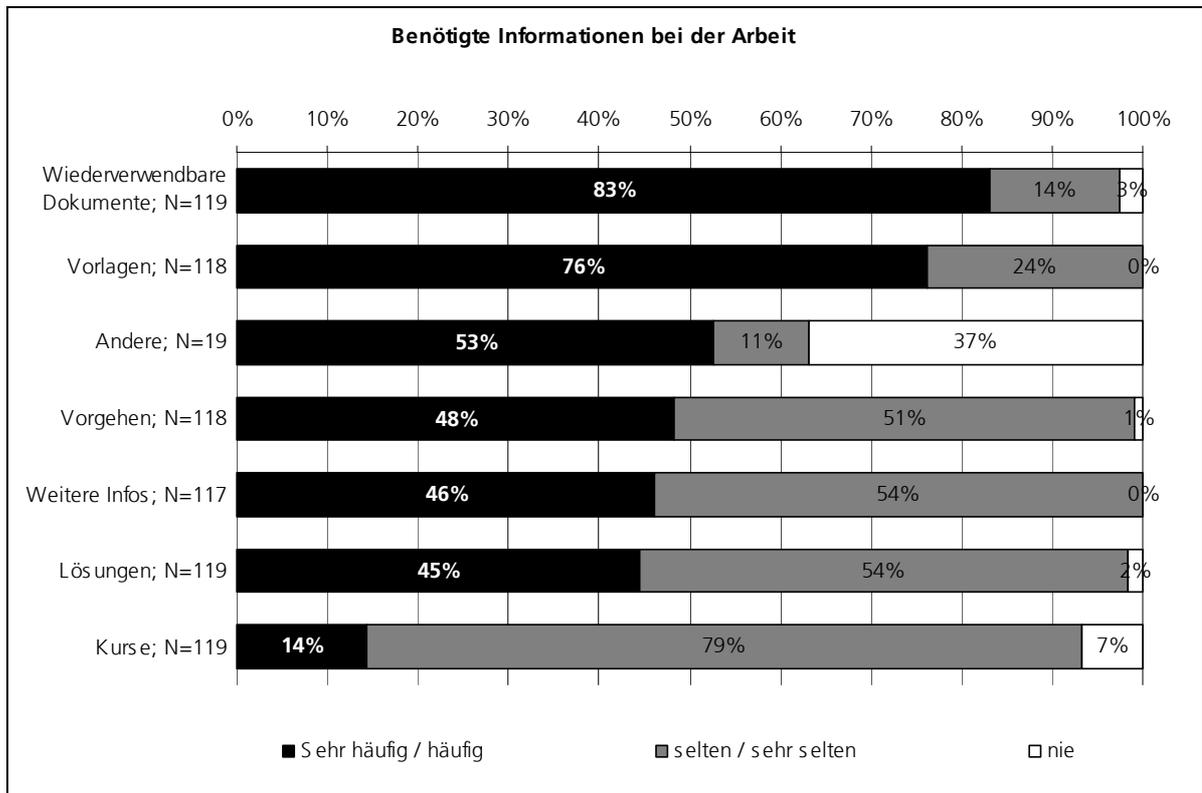


Abbildung 10: Welche Arten von Informationen benötigen Sie bei ihrer Arbeit?

3.1.2 Grund für die Informationsermittlung

Ein wesentlicher Ansatzpunkt für die Gestaltung intelligenter Unterstützung sind die Gründe für die Informationsbeschaffung. Diese Gründe geben weiterhin Hinweise darauf, wie wichtig die jeweilige Information ist und ob die Gründe eher intrinsischer Natur sind oder durch die Organisation vorgegeben sind.

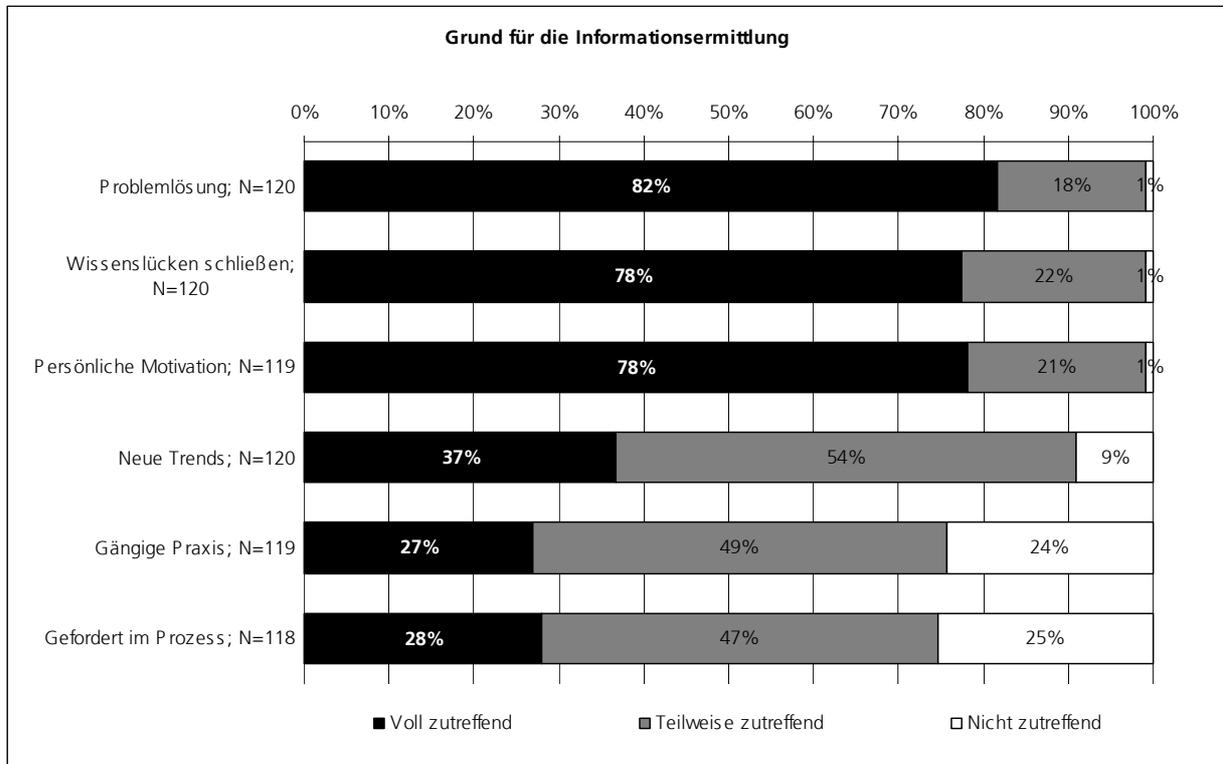


Abbildung 11: Warum ermitteln Sie Informationen?

Die drei Hauptgründe für die Ermittlung von Informationen sind die Lösung konkreter Probleme, das Schließen aktueller Wissenslücken und die persönliche Motivation (Abbildung 11). Diese Ergebnisse bestärken die Vermutung aus dem vorigen Abschnitt, dass intelligente Assistenz in erster Linie konkrete, schnell zu lösende Probleme adressieren sollte. Weiterhin lassen diese Ergebnisse vermuten, dass die Hauptgründe für die Ermittlung von Informationen eher intrinsischer Natur sind. Zudem liegt die höchste Motivation für die Informationsermittlung in der Befriedigung aktueller Informationsbedürfnisse. Längerfristige Informationsbedürfnisse wie Information über neue Trends sind eher untergeordneter Natur. Auch haben die gängige Praxis und organisatorische Gründe für die Motivation zur Informationsbeschaffung im Vergleich zu den anderen Gründen geringeren Einfluss. Intelligente Unterstützung sollte sich daher auf die Erkennung und Unterstützung konkreter Informationsbedürfnisse konzentrieren, die sich aus der Arbeit des unterstützten Individuums ergeben.

3.1.3 Lernspezifische Assistenz

Auch bei der Frage „Welche lernspezifischen Aspekte sollte die Assistenz verbessern?“ ergibt sich ein ähnliches Bild wie in den beiden vorigen Abschnit-

ten (siehe Abbildung 5). Hier gaben 82 Prozent der Teilnehmer erneut die Lösung von konkreten Problemen als „voll zutreffend an“. Die anderen Gründe wurden verhältnismäßig gering bewertet. Daher liegt die Interpretation nahe, dass die Benutzerakzeptanz dadurch gesteigert werden kann, dass der langfristige Aufbau von Kompetenzen in eine Folge von Problemlösungen integriert wird.

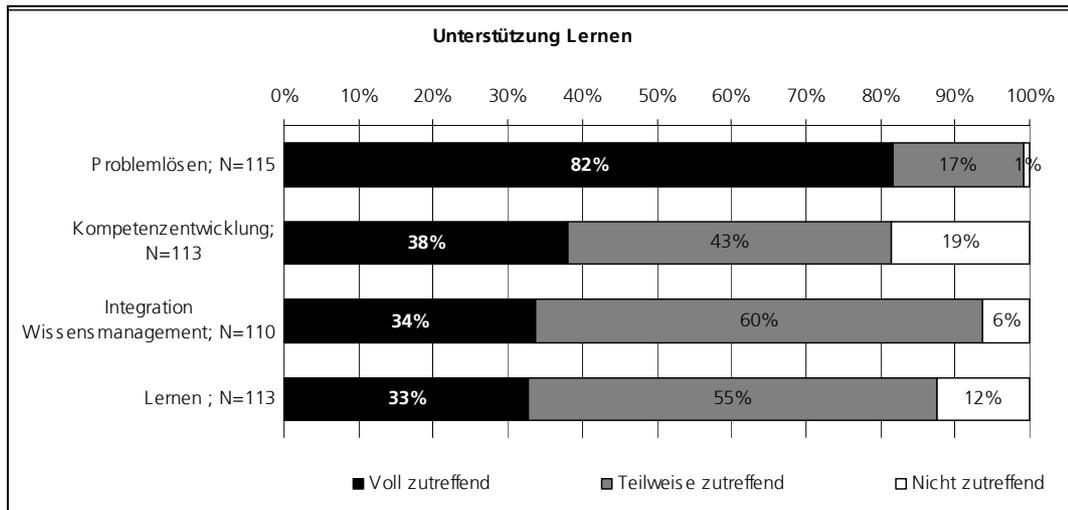


Abbildung 12: Bedarf an intelligenter Assistenz für die Unterstützung des Lernens

3.1.4 Eigenschaften und Umgang mit Information

Die Eigenschaften von Informationen in Bezug auf Qualitätskriterien wie Korrektheit und Nützlichkeit (siehe Abbildung 13) sind für die Umsetzung von intelligenter Assistenz in zweierlei Hinsicht relevant: Zum einen beschreiben diese Qualitätskriterien die Daten, auf welche eine intelligente Unterstützung aufbauen kann. Zum anderen bieten Defizite bei diesen Qualitätskriterien Hinweise darauf, welche Aspekte der Qualität von Information unterstützt werden können.

Mehr als 50 Prozent der Teilnehmer bewerten den Umgang in Bezug auf Korrektheit (60 Prozent) und Nützlichkeit (53 Prozent) mit „sehr gut / gut“. Daher kann mit einer hohen Wahrscheinlichkeit die intelligente Assistenz auf korrekte wie auch nützliche Daten einer Organisation aufsetzen. Im Mittelfeld liegen Verfügbarkeit (46 Prozent), Verbreitung (45 Prozent), Konsistenz (43 Prozent), Auffindbarkeit (43 Prozent) und Nutzung (39 Prozent). In diesem Feld wird die Verbreitung von Information mit 22 Prozent am häufigsten mit „schlecht / sehr schlecht“ bewertet. Dieses Ergebnis ist insofern interessant, da mit der vergleichsweise guten Bewertung der Verfügbarkeit und Auffindbarkeit die Informationen eigentlich gut zu erreichen ist. Eine mögliche Interpretation ist, dass

für den Benutzer die vorhandenen Quellen nicht derart erschlossen werden, dass die jeweils relevanten Informationen die Aufmerksamkeit des Benutzers finden. Zur Verbesserung der Verbreitung sollte eine intelligente Unterstützung die Interessen des Benutzers ermitteln und die vorhandenen Quellen passend aufbereiten. Das Schlusslicht bilden die Qualitätseigenschaften Verständlichkeit, Vollständigkeit und Konformität mit organisationalen Regelungen. Intelligente Assistenz kann durch das Vorschlagen geeigneter Vorlagen und durch Unterstützung beim Ausfüllen dieser Vorlagen die Probleme bei diesen drei Qualitätseigenschaften adressieren.

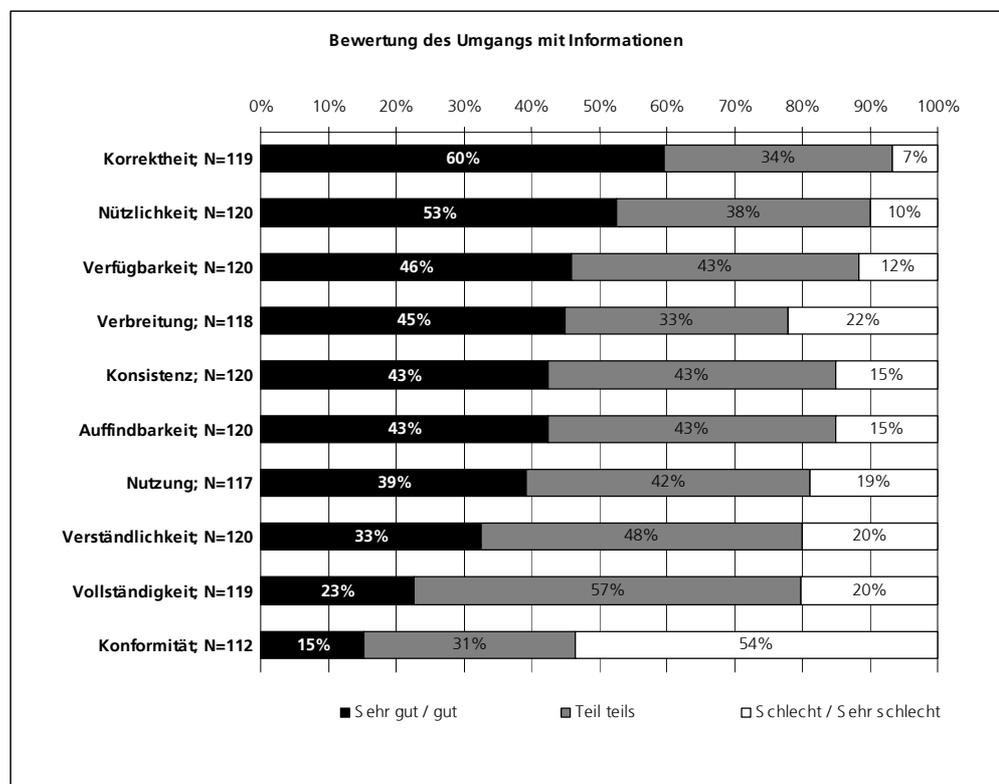


Abbildung 13: Wie bewerten Sie die Eigenschaften und den Umgang mit Informationen in ihrem Unternehmen?

3.2 Durchführung von Softwareprojekten

Als weiterer Anhaltspunkt für die intelligente Unterstützung von Software-Engineering-Aktivitäten dient die Durchführung von Projekten und die im Rahmen dieser Projekte genutzten Dokumente.

Bei den Vorgehensmodellen liegen mit Adaptive Software Engineering (36 Nennungen), Crystal (32) und Extreme Programming (25) drei agile Entwicklungsmethoden unter den ersten fünf Vorgehensmodellen. Der Unified Process ist in

diesem Feld als einziger "klassischer" Entwicklungsprozess vertreten (33). Individuelle prozessgetriebene Vorgehensmodelle (25) sind auf Platz fünf. Eine intelligente Unterstützung muss daher flexibel auf den jeweiligen Status und insbesondere auf die Übergänge zwischen Prozessschritten eingehen.

Bei den Arbeitsprodukten (Software-Artefakten) liegt mit 104 Nennungen der Quellcode vorne. Danach folgen Anforderungsdokumente (81) und Datenbankschemata (76). Auf Platz vier und fünf liegen Entwurfsdokumente und Projektpläne.

3.3 Softwareentwicklungsinfrastruktur

Damit sich intelligente Assistenz möglichst nahtlos in vorhandene Tool-Landschaften integriert, muss man wissen, welche Tools für welche Aufgaben im Rahmen der Softwareentwicklung eingesetzt werden. Insbesondere kann damit ermittelt werden, in welchem Format für die Assistenz relevante Daten vorliegen.

Bei der Präsentation der Ergebnisse im Rahmen dieser Kurzfassung werden für die jeweiligen Tool-Kategorien jeweils die drei am häufigsten genannten Werkzeuge aufgeführt. Diese Ergebnisse lassen Schlussfolgerungen darüber zu, in welchem Format die jeweiligen Software-Artefakte vorliegen. Weiterhin wird dargestellt, wie viele Teilnehmer ein zwei oder mehr Werkzeuge aus der jeweiligen Tool-Kategorie einsetzen. Damit sind Rückschlüsse auf die Heterogenität der Tool-Infrastruktur für den jeweiligen Anwendungszweck möglich.

Tabelle 1: Übersicht über Tool-Infrastruktur

Tool-Kategorie	Werkzeugeinsatz			Anzahl gleichzeitig eingesetzter Tools		
	Erster Platz	Zweiter Platz	Dritter Platz	Eines	Zwei	Mehr
Anforderungen	MS Office (67)	Keine Tools (38)	Star / Open Office (30)	45	33	5
Entwurf	Kein Tool (44)	MS Visio (43)	Andere (14)	44	20	9
IDE	Eclipse (57)	Visual Studio (56)	Andere (37)	43	43	28
Programmiersprachen	Java (66)	C/C++ (62)	PhP (39)	28	30	62
Refaktorisierung	Kein Tool (61)	Eclipse (44)	Andere (8)	43	11	3
Messen	Kein Tool (90)	Andere (11)	Purify (7)	20	4	2
Testen	Kein Tool (52)	JUnit (33)	Andere (31)	43	12	13
Issue Management	Kein Tool (42)	Andere (32)	Bugzilla (24)	62	10	3
Versionierung	CVS (48)	Subversion (39)	SourceSafe (27)	66	25	9
Kommunikation	Email (90)	Groupware (54)	Wiki (40)	24	33	55
Projekt-Management	Kein Tool (55)	MS Project Server (30)	Andere (27)	57	5	0

Im Rahmen der *Anforderungsbeschreibung* sowie beim *Design* überwiegen Produkte aus dem Office-Bereich. Bei dieser Art der Erfassung liegen die jeweiligen Informationen als eigenständige Dokumente vor. Daher ist zu vermuten, dass diese Dokumente nicht technisch mit anderen Softwaredokumenten integriert sind. Damit können sich Probleme, zum Beispiel bei der Nachverfolgbarkeit von Änderungen, ergeben. Auch setzen hier verhältnismäßig viele Teilnehmer mehr als eines der jeweiligen Tools ein.

Bei den Entwicklungsumgebungen oder *IDEs* entfällt der überwiegende Anteil auf Eclipse und Visual Studio. Insbesondere Eclipse mit seinen offenen Schnittstellen bietet sich daher an, Intelligente Assistenz im Rahmen der Programmierung anzubieten. Bei den *Programmiersprachen* liegen Java und C/C++ klar vorne. Da diese Sprachen auch durch Eclipse unterstützt werden, ist dies ein weiterer Anhaltspunkt, intelligente Unterstützung basierend auf dieser Plattform anzubieten. Dabei sollte die intelligente Unterstützung sich nach Möglichkeit nicht nur auf eine Programmiersprache beschränken, da oft mehrere Programmiersprachen in einer Organisation eingesetzt werden.

Bei der *Refaktorisierung*, dem *Messen*, dem *Testen* und dem *Issue Management* werden in der Regel keine spezifischen Tools eingesetzt. Hier kann intelligente Unterstützung helfen, mögliche Hemmschwellen für den Einsatz dieser Werkzeuge abzubauen. Der Einsatz von Eclipse im Rahmen der Refaktorisierung bietet eine weitere Grundlage dafür, mögliche intelligente Unterstützung auf dieser Plattform aufzubauen. Das Test-Framework JUnit wiederum ist eine vielversprechende Grundlage für intelligente Unterstützung im Bereich des Testens. Bei allen vier Tool-Kategorien wird darüber hinaus eher ein Tool für den jeweiligen Bereich eingesetzt, so dass die Konzentration auf den jeweiligen Marktführer eine gute Basis für die Umsetzung von intelligenter Unterstützung bietet.

Der Tool-Einsatz bei der *Versionierung* verteilt sich relativ gleichmäßig auf CVS, Subversion und SourceSafe. Auch hier wird innerhalb einer Organisation eher nur ein Tool im Rahmen der Softwareentwicklung eingesetzt. Daher muss eine intelligente Unterstützung idealerweise diese drei Versionierungssysteme unterstützen.

Bei den *Kommunikationsumgebungen* ist Email klar auf Platz eins. Dies bedeutet, dass Kommunikation innerhalb von Projekten in erster Linie über dieses Medium abgewickelt wird. Daher sollten Lösungen zur intelligenten Unterstützung dieses Medium mit einbeziehen. Bei eher teamorientierten Ansätzen ist Groupware auf Platz zwei. Wikis auf Platz drei bieten einen weiteren Ansatzpunkt bei der Unterstützung zur Erstellung und zum Management von Projekt- und Software-Artefakten. Dabei sollte eine intelligente Unterstützung mehrere Kommunikationstools mit einbeziehen, da mehrere Kommunikationsumgebungen parallel in Organisationen genutzt werden.

Dedizierte *Projektmanagementumgebungen* werden eher selten eingesetzt. Allerdings wird der Projektmanagement-Server von Microsoft öfters eingesetzt, so dass eine intelligente Unterstützung in diesem Bereich auf diesem Tool aufbauen könnte. Zudem wird in den Projektmanagementumgebungen in der Regel nur ein Tool verwendet, so dass eine intelligente Unterstützung in einer konkreten Installation nur die Anbindung an ein Tool unterstützen muss.

3.4 Bekanntheit und Gestaltungsformen von Assistenz

In diesem Abschnitt sind einige Fragen darüber zusammengefasst, inwieweit Assistenz an sich und welche Formen von Assistenz bekannt sind. Dabei wird der Bekanntheitsgrad, die Nützlichkeit der Assistenz, das Vorgehen eines Assistenzsystems, sowie der Grad an gewünschter Assistenz untersucht.

3.4.1 Bekanntheitsgrad und Akzeptanz von Formen der Assistenz

Intelligente Unterstützung ist in hohem Maße abhängig von der Akzeptanz des Benutzers. Um das Adjektiv „intelligent“ zu Recht zu beanspruchen, muss sich die Unterstützung dem Benutzer in einer Form anbieten, welche die Arbeit des Benutzers möglichst gut unterstützt. Um diese Daten zu ermitteln, wurde gefragt, ob bestimmte Formen der Assistenz bekannt sind und wie stark diese genutzt werden. Da die Assistenz in der Regel nicht zwingend nötig für die Bearbeitung einer Aufgabe ist, kann aufgrund der Angaben über die Benutzung auf die Akzeptanz geschlossen werden.

Das Ergebnis zu der entsprechenden Frage ist in Abbildung 14 dargestellt. Am häufigsten wird mit 81 Prozent „sehr häufiger / häufiger“ Nutzung die Vervollständigung von Texten genutzt (wie z. B. Code Completion in Eclipse). Auf

Platz zwei und drei befinden sich Erklärungen (z.B. Beschreibungen aus Java-Doc) und die Recherche von Nutzungsorten (z.B. Deklaration und Verwendung einer Funktion) mit 69 Prozent und 57 Prozent. Am wenigsten genutzt wird die Transformation des Dokumentenaufbaus mit 29 Prozent „sehr häufiger / häufiger“ Nutzung. Diese Form der Assistenz ist mit 12 Prozent der Antworten zudem verhältnismäßig unbekannt, während die anderen Formen zwischen 1 Prozent und 6 Prozent der Teilnehmer unbekannt waren.

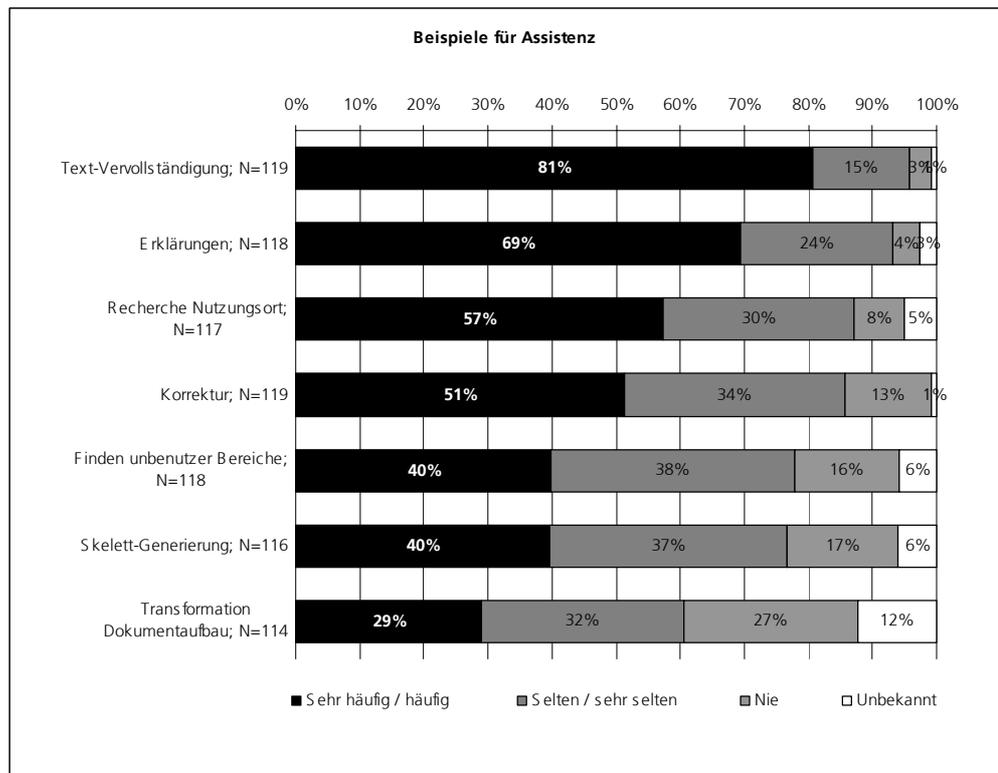


Abbildung 14: Welche Beispiele für Assistenz sind Ihnen bekannt und wie häufig nutzen Sie diese?

Diese Ergebnisse lassen eine Tendenz hin zu einfachen, schnell ausführbaren Formen von Assistenz vermuten. Komplexere Formen der Assistenz wie das Generieren von Code-Skeletten und die Transformation von Dokumenten werden eher selten genutzt oder sind unbekannt. Daher sollte bei diesen Formen besonderes Augenmerk darauf verwendet werden, wie die entsprechende Unterstützung dem Benutzer angeboten wird. Ein Ansatzpunkt ist die Vermutung, dass einfachere Formen der Unterstützung eher verstanden und leichter rückgängig gemacht werden können. Daher sollten komplexe Formen der Unterstützung dem Benutzer das Verständnis für die Ergebnisse und die Kontrolle über die Veränderungen ermöglichen. Allerdings muss hier ebenfalls ermittelt werden, ob komplexe Formen der Assistenz generell weniger genutzt werden, da ihre Ergebnisse per se seltener im Rahmen der Arbeit benötigt werden. Ein

Beispiel ist die Skelett-Generierung von Code, die am Anfang einer Klasseerstellung häufig benutzt wird - während die Textvervollständigung auch während der restlichen Entwicklung sehr häufig verwendet wird.

3.4.2 Nützlichkeit von Darstellungsformen

Um weitere Hinweise darüber zu erhalten, welche Formen der Darstellung akzeptiert werden, wurde direkt nach der Nützlichkeit von Darstellungsformen von Assistenzergebnissen gefragt.

Die Ergebnisse zu dieser Frage finden sich in Abbildung 15. Tooltips, Listen und Bilder bilden mit 67 Prozent und mehr Antworten „sehr gut / gut“ die Gruppe der Spitzenreiter. Video, Audio und animierte Assistenz bilden die andere Gruppe, die mit 59 Prozent und mehr mit „schlecht / sehr schlecht“ bewertet wurden.

Damit zeigt sich auch hier die Tendenz zu einfachen Formen der Assistenz, die zudem visuell schnell erfassbar sind. Animierte oder hörbare Formen der Assistenz scheinen als störend empfunden zu werden. Eine mögliche Erklärung für diese Bewertung ist, dass diese Darstellungsformen die volle Aufmerksamkeit eines Benutzers verlangen und damit von der Arbeit abhalten. Dies bedeutet für intelligente Unterstützung, dass sinnvolle, textuelle Darstellungsformen ausreichend sind. Dadurch wird der Aufwand für mögliche Implementierungen intelligenter Unterstützung reduziert.

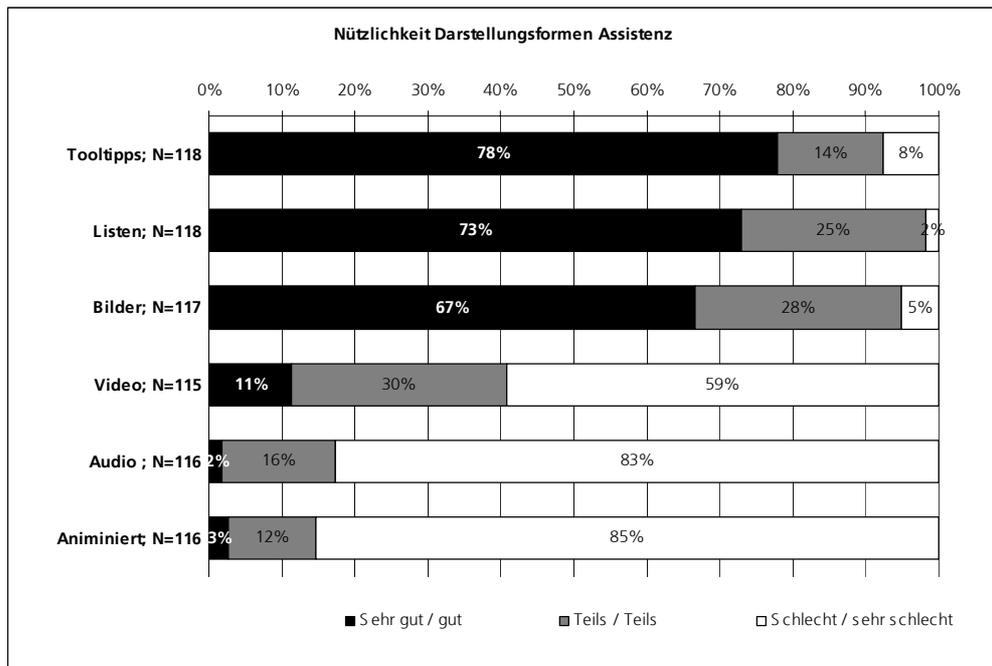


Abbildung 15: Wie bewerten Sie die Nützlichkeit der folgenden Darstellungsformen der Assistenz?

3.4.3 Vorgehen bei Assistenz

Die dritte Frage zu möglichen Darstellungsformen von Assistenz ist das Vorgehen bei der Assistenz.

Wie in Abbildung 16 gezeigt, bevorzugen 47 Prozent der Teilnehmer eine reaktive Assistenz, d.h. dass der Benutzer die Assistenz explizit nachfragt oder anstößt. Ein Drittel der Teilnehmer (33 Prozent) bevorzugen hingegen proaktive Assistenz, bei der die Vorschläge durch das Assistenzsystem unterbreitet werden. 7 Prozent bevorzugen andere Formen, 13 Prozent machten keine weiteren Angaben zu dieser Frage.

Aufgrund dieser Ergebnisse kann keine klare Entscheidung darüber abgeleitet werden, ob eine Assistenz reaktiv oder proaktiv gestaltet werden soll. Daher muss diese Entscheidung für den jeweiligen Anwendungsfall getroffen werden.

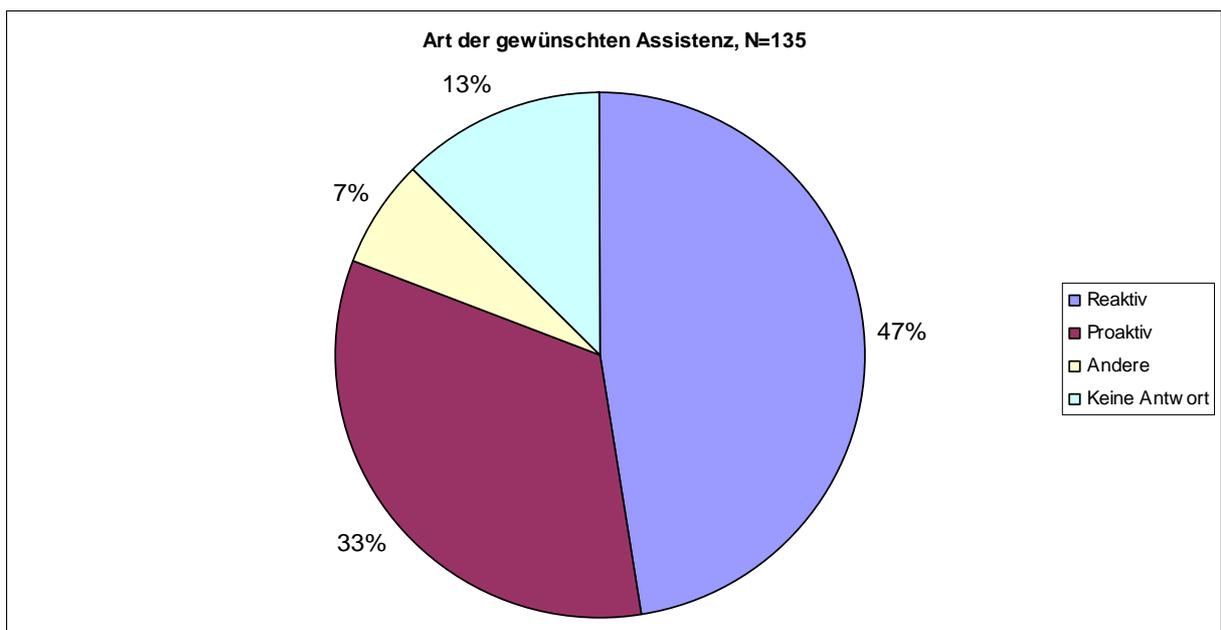


Abbildung 16: Welches Vorgehen der Assistenz bevorzugen Sie?

3.4.4 Grad von Assistenz

Einen weiteren Hinweis auf mögliche Gestaltungsformen von Assistenz liefert die Frage nach dem Grad von Assistenz. Hier konnten die Teilnehmer unter den verschiedenen vorgeschlagenen Assistenzgraden angeben, welcher Grad von Assistenz bevorzugt. Dabei waren mehrfache Antworten möglich.

In Abbildung 17 sind die Ergebnisse für diese Frage angegeben. Etwa die Hälfte der Teilnehmer bevorzugten einen Überblick über die aktuell möglichen Alternativen (69 Antworten) und den Vorschlag von möglichen Aktivitäten (Ausführung auf Wunsch, 59 Antworten). Die zweite Gruppe bildet die Auswahl einer Möglichkeit von verschiedenen Alternativen (39 Antworten) und der Information des Benutzers nach der Anwendung einer Aktivität durch die intelligente Assistenz (26 Antworten). Die anderen Grade der Assistenz werden nicht bevorzugt: die Präsentation nur eines einzelnen Vorschlags aus einer Liste von Möglichkeiten, die sofortige Ausführung dieses Vorschlags, die Entscheidung der Assistenz über die Information des Benutzers sowie die Ausführung nach einer Zeitspanne und der autonomen Ausführung.

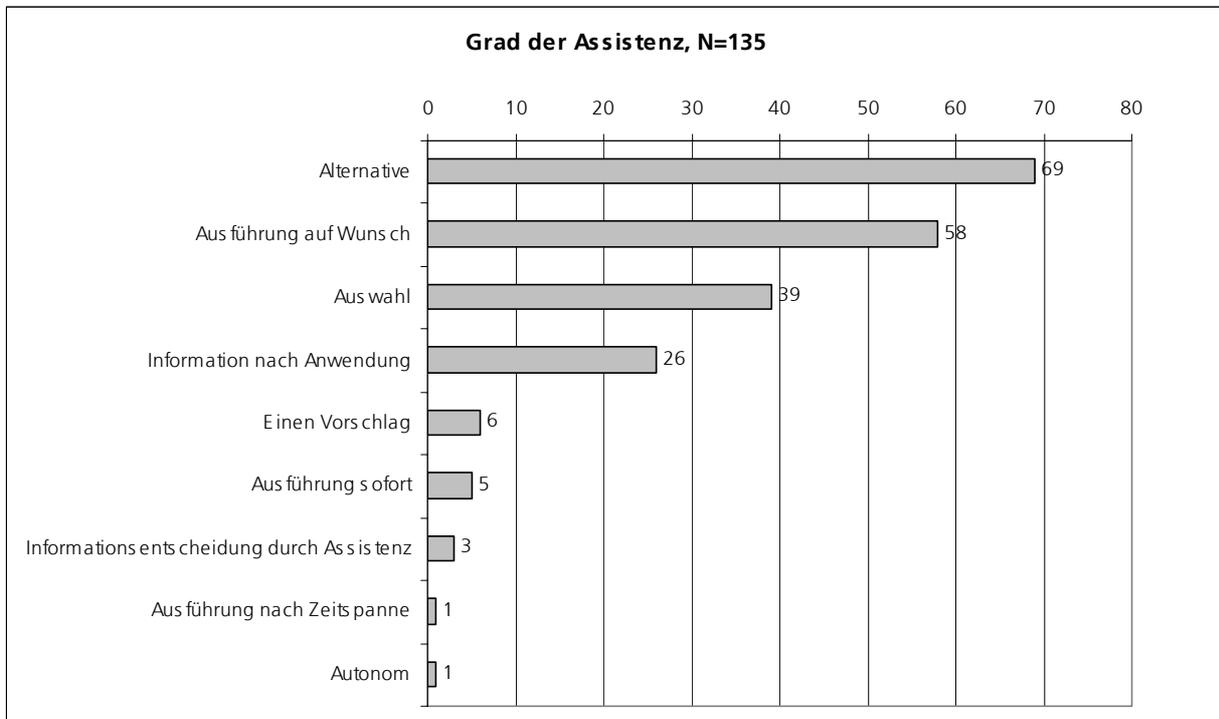


Abbildung 17: Welchen Grad von Assistenz bevorzugen Sie?

Diese Ergebnisse geben einen weiteren Hinweis darauf, dass der Benutzer bei der intelligenten Assistenz die Kontrolle und die Übersicht über die jeweiligen Aktivitäten behalten möchte. Durch den Vorschlag von möglichen Alternativen, gefolgt von einer Auswahl einer Aktivität, kann der Benutzer diese Kontrolle behalten.

3.5 Potenziale für den Einsatz von Assistenz in der Softwareentwicklung

Als weiteren Aspekt für das Design von intelligenter Assistenz wird in diesem Abschnitt untersucht, in welchen Kontexten ein Bedarf an Assistenz vorliegt. Hinsichtlich dieses Bedarfs werden zunächst die Phasen und die Produkte im Rahmen der Softwareentwicklung untersucht. Daran schließt sich die Analyse des Bedarfs an Assistenz im Rahmen der Erstellung und Wiederverwendung von Dokumenten an.

3.5.1 Assistenz in welchen Phasen

Die Ergebnisse darüber, in welchen Phasen Bedarf an Assistenz vorliegt, sind in Abbildung 18 dargestellt. Mehr als die Hälfte der Teilnehmer hat Bedarf an Assistenz im Rahmen der Programmierung. 50 Teilnehmer oder mehr haben Bedarf im Rahmen der Anforderungsentwicklung, beim Design und bei der Projektplanung. Bei den anderen Kategorien bewegten sich die Anzahl der Teilnehmer, die ein entsprechendes Bedürfnis haben, zwischen 14 und 33.

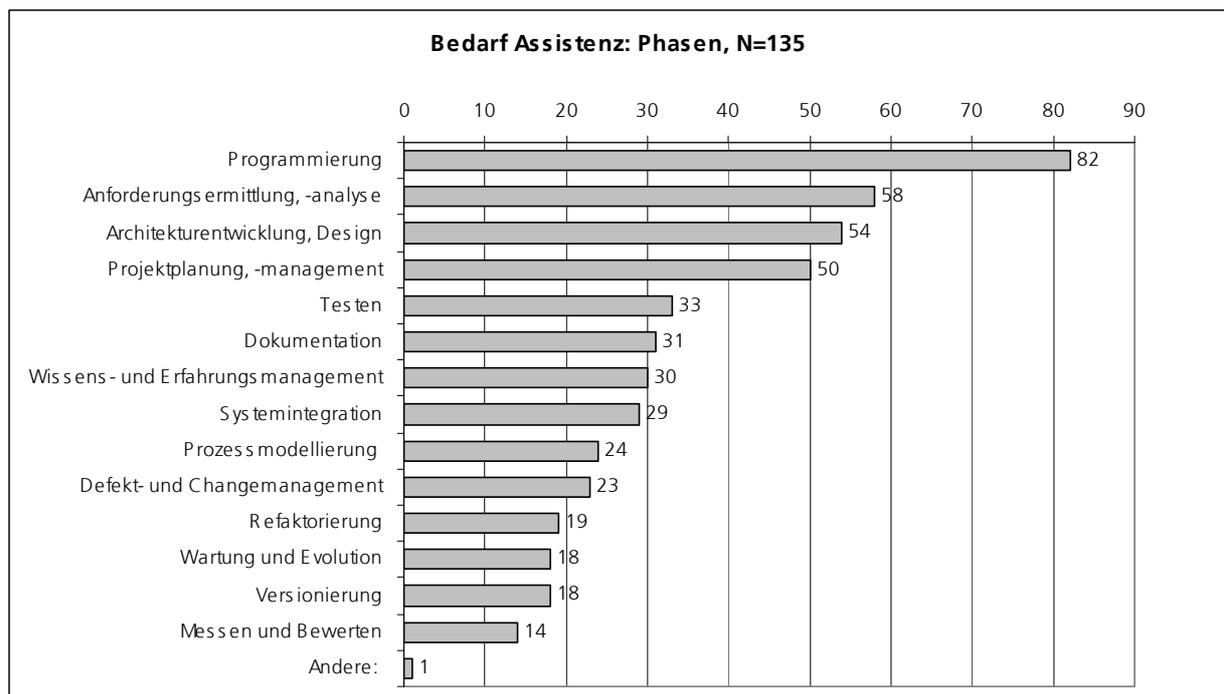


Abbildung 18: In welchen Phasen haben Sie häufig Bedarf für weitere Informationen (bspw. zur Entscheidungsfindung, Arbeitsunterstützung oder Kompetenzentwicklung)?

Generell scheint ein hoher Bedarf bei der Assistenz im Rahmen der Kernphasen und -aktivitäten (Programmierung, Anforderungsmanagement, Design und Pro-

jektmanagement) im Rahmen der Softwareentwicklung zu bestehen. Bei den anderen Phasen wird aktuell kein Bedarf an Assistenz gesehen. Eine mögliche Erklärung ist, dass diese Phasen und Aktivitäten eher von Spezialisten übernommen werden, die keine Unterstützung benötigen oder sich entsprechende Unterstützungswerkzeuge selbst entwickelt haben. Dem regulären Mitarbeiter präsentieren sich die in diesen Phasen eingesetzten Werkzeuge über ein einfaches und damit verständliches Interface (z.B. wie bei Versionierungs- und Defektmanagement-Werkzeugen). Dem gegenüber steht, dass Assistenz in den Phasen und Aktivitäten nachgefragt wird, in denen bereits eine hohe Anzahl an entsprechenden Werkzeugen verfügbar ist. Dies gibt Hinweise auf ein mögliches Marktpotenzial, da diese Werkzeuge zum einen weit verbreitet sind, andererseits hier aber auch ein entsprechender Bedarf vorliegt.

3.5.2 Arbeitsprodukte

Ein zweiter Aspekt des Einsatzes von intelligenter Assistenz sind die Produkte, bei deren Bearbeitung Bedarf für Unterstützung gesehen wird.

Wie in Abbildung 19 dargestellt, wird Bedarf an Unterstützung insbesondere bei der Erstellung von Quellcode angesehen: 68 Teilnehmer sahen hier Bedarf für intelligente Unterstützung. Anforderungsdokumente (46 Teilnehmer), Entwurfsdokumente (39 Teilnehmer) und Datenbankschemata (32 Teilnehmer), bilden die nächste Gruppe von Dokumenten. Projektpläne liegen mit 29 Teilnehmern, die hier Bedarf sehen, mit Testdokumenten und Benutzerdokumentation gleich auf. 26 Teilnehmer nutzen Erfahrungsdokumentation. Andere Dokumente werden eher selten genutzt.

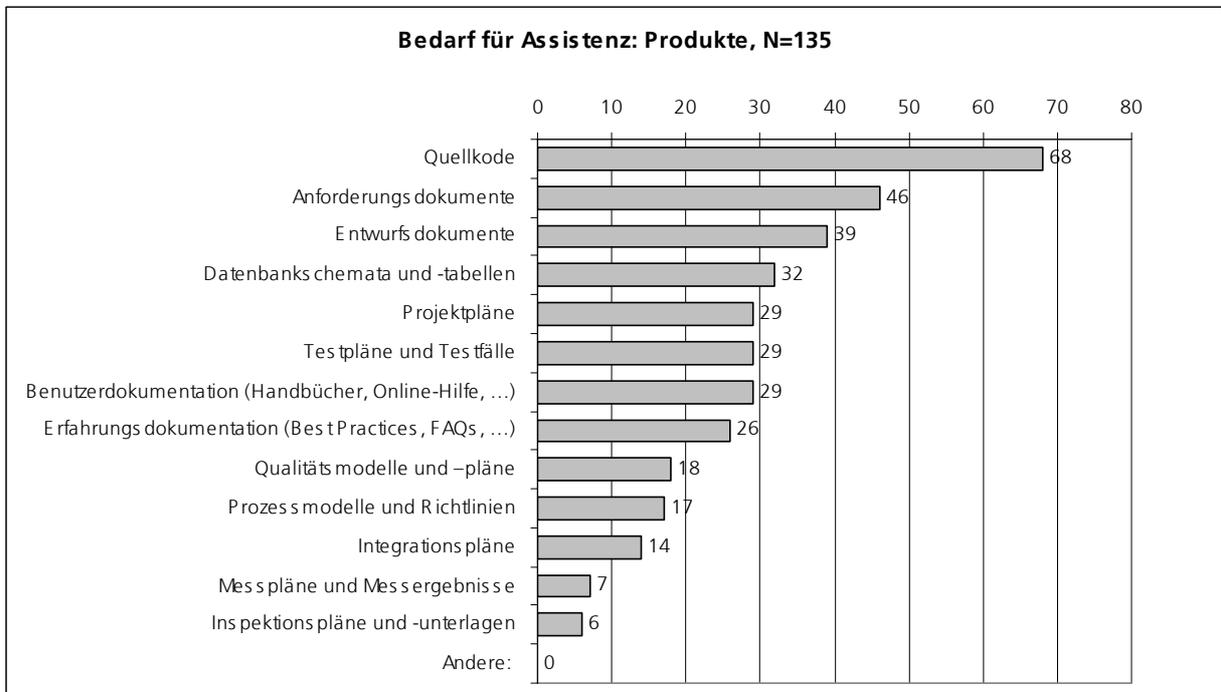


Abbildung 19: Bei welchen Arbeitsprodukten haben Sie häufig Bedarf für weitere Informationen (bspw. für Vorlagen, Glossare, optimierte Algorithmen oder wiederverwendbare Textbausteine)?

Wenn man die Datenbankschematas mit zu den Entwurfsdokumenten zählt, ergibt sich aus den Antworten zu dieser Frage das gleiche Bild wie im vorigen Abschnitt: Erneut stehen Dokumente aus den Kernaktivitäten der Softwareentwicklung im Vordergrund. Dies bekräftigt die Vermutung, dass ein Markt für intelligente Unterstützung bei diesen häufig genutzten Dokumenten vorhanden ist.

3.5.3 Bedürfnisse an Intelligenter Assistenz bei der Erstellung von Dokumenten

Als drittes werden die Bedürfnisse für intelligente Unterstützung im Rahmen der Erstellung von Dokumenten beschrieben.

Die Antworten der Teilnehmer sind in Abbildung 20 dargestellt. Mindestens die Hälfte der Teilnehmer möchte Unterstützung bei der Anzeige von Fehlern, Vervollständigung von Ausdrücken, Erklärung über die aktuelle Situation des Dokuments und die Entdeckung unbenutzter Elemente im Dokument. Die anderen Kategorien (z.B. Erzeugung von Skeletten, Ermittlung von Experten) werden mindestens von einem Viertel der Teilnehmer als nötig erachtet.

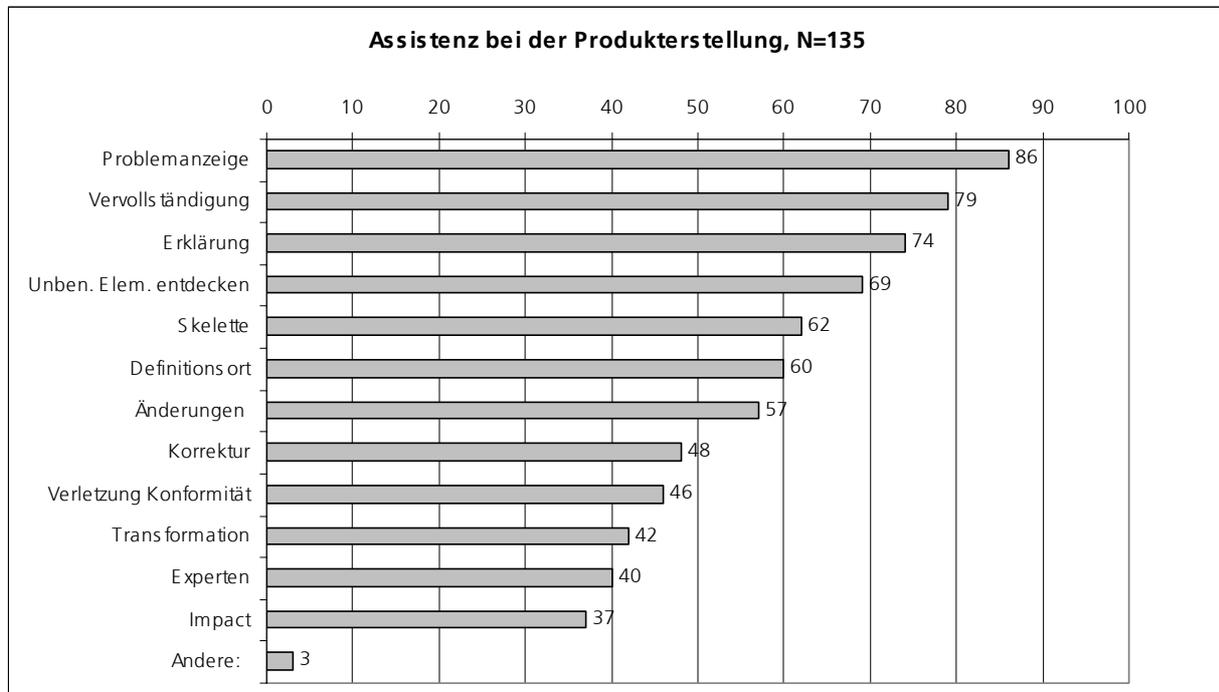


Abbildung 20: Welche Art von Assistenz bei der Erstellung und Änderung von Arbeitsdokumenten wünschen Sie sich?

Dieses Ergebnis bestätigt den Trend zu einfachen, überschaubaren Formen der Assistenz aus 3.4, Bekanntheit und Gestaltungsformen von Assistenz. Zu dem implizieren die beiden erstgenannten Formen der Unterstützung (Problemanzeige, Vervollständigung), dass eine Repräsentation des internen Aufbaus und der Beziehungen zwischen Software-Artefakten vorhanden sein müssen. Hier stellt sich die Frage, wie diese Informationen über Software-Artefakte in einer Form repräsentiert werden können, dass alle Tools innerhalb einer Entwicklungs-Infrastruktur diese Aufbau- und Beziehungsinformationen verwenden können.

3.5.4 Bedürfnisse an intelligente Assistenz bei der Wiederverwendung von Dokumenten

Assistenz sollte sich nicht nur auf die Erstellung von neuen Artefakten beschränken, sondern auch Unterstützung bei der Wiederverwendung geben. Durch diese Unterstützung werden Redundanzen vermieden und somit die Effizienz der Softwareentwicklung gesteigert.

Abbildung 21 zeigt die Ergebnisse für Assistenz im Rahmen der Wiederverwendung. 72 der Teilnehmer haben Bedarf an Unterstützung bei der Auswahl eines oder mehrerer Artefakte aus einer Menge von möglichen Kandidaten. Ebenfalls mehr als die Hälfte der Teilnehmer wünschten sich Unterstützung bei der Suche

nach derartigen Kandidaten. Bedarf bei der Anpassung von vorhandenen Artefakten sehen 51 Teilnehmer; Erfahrungen aus der Anwendung des wiederverwendeten Artefaktes wünschen sich 33 Teilnehmer.

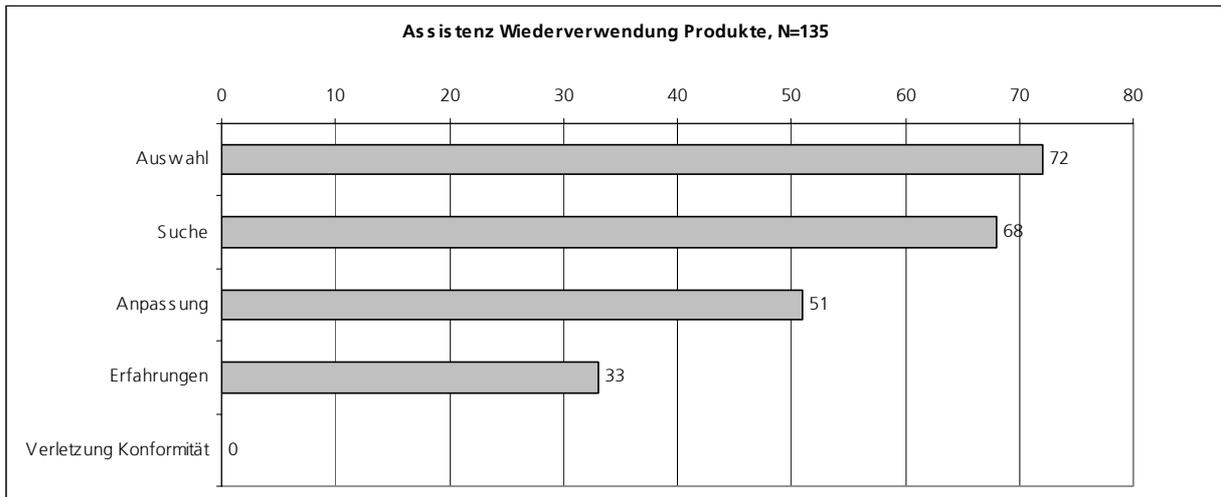


Abbildung 21: In welchem Schritt bei der Wiederverwendung von Arbeitsdokumenten wünschen Sie sich Assistenz?

Diese Ergebnisse legen die Vermutung nahe, dass das aktuell vielversprechendste Einsatzfeld für intelligente Assistenz die Erschließung vorhandener Software-Artefakte mittels Suche und anschließender Auswahl ist. Eine weitere Unterstützung bei der Bearbeitung von Artefakten ist nicht im Fokus des Interesses. Eine mögliche Erklärung ist, dass im Rahmen der Wiederverwendung Artefakte mit positiven Erfahrungen gefunden werden sollen, die möglichst ohne Modifikationen anwendbar sind. Damit wäre die Anpassung und Bewertung aufgrund von Erfahrungen implizit in der Suche und Auswahl enthalten. Eine andere Erklärung könnte sein, dass aufgrund der unzureichenden Erschließung vorhandener Quellen für wiederverwendbare Artefakte die Wiederverwendung generell gering ist. In diesem Fall bestünde kein Bedarf bei der Unterstützung der nachgelagerten Bearbeitungsschritte Anpassung und Eingabe von Erfahrungen.

4 Fazit

Die Softwareindustrie ist eine hoch innovative Branche, deren Endprodukte oft in Form von elektronischen Dokumenten (z.B. Quellcode, Anforderungsdokumente, etc.) vorliegen. Die Entwicklung von Software ist eine komplexe und wissensintensive Aufgabe, bei der viele Informationen durch die Mitarbeiter recherchiert, generiert und weitergegeben werden. Informationen über die Software wie z.B. Entwicklungsdokumente, Quellcode und dokumentierte Erfahrungen werden in den unterschiedlichsten Datenformaten erstellt und in verschiedenen Systemen gespeichert. Der Informationsbedarf eines Mitarbeiters in einem Softwareprojekt hängt von der aktuellen Arbeitstätigkeit, dem Arbeitsdokument und der eigenen Erfahrung ab und soll möglichst schnell und präzise befriedigt werden.

Intelligente Assistenzsysteme können an dieser Stelle die wissensintensiven Tätigkeiten wie die Erstellung, Modifikation oder Wartung dieser Arbeitsprodukte wesentlich unterstützen. Heutzutage gibt es bereits die unterschiedlichsten Arbeitsumgebungen, die teilweise mit mehr oder weniger intelligenten Assistenzsystemen ausgestattet sind. Diese Assistenzsysteme können dabei in den verschiedensten Formen auftreten und sehr unterschiedliche Arbeitsschritte unterstützen.

Die hierzu durchgeführte explorative Umfrage hatte zum Ziel, zu ermitteln, wo die intelligente Assistenz in der heutigen Werkzeuglandschaft in Softwareorganisationen am meisten benötigt und welche Art von Assistenz bevorzugt wird. Weiterhin sollten die Konzepte der intelligenten Assistenz geklärt, die Motivation für intelligente Assistenzsysteme aufgedeckt, einige Beispiele für intelligente Assistenz beschrieben und die Einstellung gegenüber intelligenter Assistenz in deutschen Softwareunternehmen sowie die Nachfrage danach ermittelt werden.

Der vorliegende Bericht fasst die wesentlichen Kerndaten der Umfrage zusammen, an der 135 Teilnehmer aus deutschen Unternehmen und Einrichtungen teilnahmen. Die Umfrage zeigte eine hohe Nachfrage nach und Akzeptanz für unaufdringliche, schnelle und reaktive Assistenz in den Kernphasen des Software Engineering. Weiterhin kristallisierten sich folgende Ergebnisse heraus:

- Assistenz, die dem Nutzer angeboten wird, sollte die Konformität, Vollständigkeit und Verständlichkeit der zu entwickelnden Dokumente (Informationen) verbessern (siehe Abschnitt 3.1.4).
- Die Nutzer benötigen meist wiederverwendbare Dokumente oder Vorlagen aber seltener Kurse oder Seminare (siehe Abschnitt 3.1.1).

- Es herrschen heterogene Entwicklungsumgebungen vor, die es zu integrieren gilt (siehe Abschnitt 3.3).
- Für intelligente Assistenz im Software Engineering werden insbesondere reaktive und textuelle Systeme nachgefragt (siehe Abschnitt 3.4).
- Intelligente Assistenz wird im Wesentlichen in den Kernphasen Programmierung, Entwurf und Anforderungsanalyse benötigt.
- Die Phasen, in denen Assistenz benötigt wird, verfügen über ausreichend formale Arbeitsprodukte, um dokumentspezifische Assistenzsysteme zu speisen.

Der ausführliche Bericht zu dieser Fallstudie beinhaltet die vollständige Beschreibung aller Ergebnisse der Umfrage sowie eine ausführliche Auswertung und Trendanalyse zum Thema „Intelligente Assistenz“ basierend auf den ermittelten Daten. Dabei wird insbesondere auch die typische Tool-Landschaft beschrieben, die in deutschen KMUs vorherrscht, und in welche intelligente Assistenzsysteme eingebettet werden können. Die Resultate sind insbesondere für Manager und Softwareentwickler interessant, die für die IT-Infrastruktur oder Weiterbildungsmaßnahmen in ihrer Softwareorganisation zuständig sind.

Literatur

- Winograd, T. (1973). Breaking the complexity barrier again. Paper presented at the Proceedings of the ACM SIGPLAN SIGIR Interface meeting on programming languages - information retrieval, 4-6 Nov. 1973, Gaithersburg, MD, USA.
- Teitelman W. (1972). "Automated programming - the programmer's assistant," Proceedings of the Fall Joint Computer Conference, AFIPS, 1972, page 915-921.
- Kaiser, G. E., Feiler, P. H., & Popovich, S. S. (1988). "Intelligent assistance for software development and maintenance." *Software, IEEE*, 5(3), 40-49.
- Rich, C., & Waters, R. C. (1988). The Programmer's Apprentice: a research overview. *Computer, USA* * vol 21 (Nov. 1988), no. 11, p. 10-25
- Reubenstein, H. B., & Waters, R. C. (1991). The Requirements Apprentice: automated assistance for requirements acquisition. *IEEE Transactions on Software Engineering, USA* * vol 17 (March 1991), no 3, p 226-240
- Waters, R. C., & Yang, M. T. (1991). Toward a Design Apprentice: supporting reuse and evolution in software design. *SIGSOFT Software Engineering Notes, USA* * vol 16 (April 1991), no 2, p 33-44
- Ng, K., Kramer, J., Magee, J., & Dulay, N. (1995). The Software Architect's Assistant-a visual environment for distributed programming. Proceedings of the Twenty-Eighth Hawaii International Conference on System Sciences, 1995.
- Robbins J. E. (1999). Cognitive Support Features for Software Development Tools. Ph.D. Thesis. University of California, Irvine
- Barnett, M., Rustan, K., Leino, M., & Schulte, W. (2005). The Spec programming system: an overview. International Workshop, CASSIS 2004. Marseille, France, 10-14 March 2004 * Berlin, Germany: Springer Verlag, 2005, p 49-69

Dokumenten Information

Titel: Intelligente Assistenz in der
Softwareentwicklung 2006: Zu-
sammenfassung der Ergebnisse

Datum: 29. Mai 2006
Report: IESE-045.06/D
Status: Final
Klassifikation: Öffentlich

Copyright 2006, Fraunhofer IESE.
Alle Rechte vorbehalten. Diese Veröffentlichung darf
für kommerzielle Zwecke ohne vorherige schriftliche
Erlaubnis des Herausgebers in keiner Weise, auch
nicht auszugsweise, insbesondere elektronisch oder
mechanisch, als Fotokopie oder als Aufnahme oder
sonstwie vervielfältigt, gespeichert oder übertragen
werden. Eine schriftliche Genehmigung ist nicht erfor-
derlich für die Vervielfältigung oder Verteilung der
Veröffentlichung von bzw. an Personen zu privaten
Zwecken.