

Sperrvermerk

Die vorliegende Arbeit beinhaltet interne und vertrauliche Informationen der Firma dotSource GmbH. Die Weitergabe des Inhaltes der Arbeit und eventuell beiliegender Zeichnungen und Daten im Gesamten oder in Teilen ist grundsätzlich untersagt. Es dürfen keinerlei Kopien oder Abschriften – auch in digitaler Form – gefertigt werden. Ausnahmen bedürfen der schriftlichen Genehmigung der Firma dotSource GmbH. Die Arbeit ist nur Mitgliedern des Prüfungsausschusses zugänglich zu machen.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Problemstellung	3
1.2	Ziele und Thesen der Untersuchung	4
1.3	Abgrenzung der Themenstellung.....	4
1.4	dotSource GmbH.....	5
1.5	Betriebswirtschaftliche Bedeutung des Begriffs „Produktivität“	8
1.6	Messmethoden der Dienstleistungsbranche	10
2	Produktivität in der Softwareentwicklung	15
2.1	Ermittlung der Produktivität in der Softwareentwicklung	18
2.1.1	Function Point Analyse	22
2.1.2	Weitere Messmethoden	23
2.1.3	Zwischenfazit	24
2.2	Kennzahlen für Produktivität und Qualität.....	25
2.3	Ursachen für Produktivitätseinbußen.....	28
2.4	Produktivitätsmethode „Pomodoro“	31
2.5	Vorstellung produktivitätsoptimierender Maßnahmen.....	33
2.5.1	Zielvereinbarungssysteme	33
2.5.2	Mitarbeiter-Aktienindex	35
2.5.3	Partizipatives Produktivitätsmanagement	36
2.5.4	Business-Dashboards	37
2.6	Fazit These 1	39

3	Analyse der Produktivität	40
3.1	Struktur eines Projektteams.....	40
3.2	Joblevel	41
3.3	Team Intershop 2	42
3.4	Methodisches Vorgehen bei der Untersuchung.....	42
3.4.1	Analyse bestehender Arbeitsabläufe mittels Aufwandsbuchungen	44
3.4.2	Aufbau der Untersuchung	46
3.4.3	Auswahl der Konkurrenten.....	46
3.5	Auswertung der Arbeitszeit-Produktivität in der dotSource GmbH	49
3.5.1	Gesamtbetrachtung	49
3.5.2	Sprintbetrachtung.....	49
3.5.3	Tagesbetrachtung	53
3.5.4	Joblevelbetrachtung	57
3.5.5	Profession im Wettbewerbsvergleich	59
3.6	Fazit These 2.....	61
4	Fazit und Ausblick.....	64
	Abbildungsverzeichnis	67
	Tabellenverzeichnis	69
	Abkürzungsverzeichnis	70
	Anlagenverzeichnis.....	71
	Literaturverzeichnis	85
	Interne geschützte Quellen.....	91

1 Einleitung

„Remember that time is money.“¹

Benjamin Franklin (1706 - 1790)

So lautet ein Ratschlag in Benjamin Franklins Schrift „*Advice to Young Tradesmen*“ aus dem Jahr 1748. Rund 30 Jahre später beschreibt der Moralphilosoph Adam Smith, wie sich der „*Wohlstand der Nationen*“ durch Spezialisierung, Arbeitsteilung, und einen freien Markt steigern lässt. Der Mensch handelt, nach Smith, eigennützig und ist bestrebt, seine Bedürfnisse bestmöglich zu befriedigen.²

Jedoch ist nicht jeder dazu in der Lage, seinen Bedarf an Nahrung, Unterkunft oder Kleidung selbständig zu erfüllen. Nicht jeder Mensch kann jagen, Häuser bauen oder Textilien verarbeiten. Wenn der Schneider bspw. seine Nahrung selbst erlegt, benötigt er mehr Zeit als der Jäger. Zeit, in welcher der Schneider mehr Textilwaren hätte produzieren, als Wild erlegen können. Ein solches Vorgehen wäre folglich, für eine bestmögliche Bedürfnisbefriedigung, ineffektiv. Die Arbeit wird also so verteilt, dass jedermann Güter produziert, auf die er spezialisiert ist und anderen eigennützig, also im Austausch gegen Geld oder Waren, zur Verfügung stellt.

Mit der Arbeitsteilung wurde eine effektive Methode gefunden, Zeit und damit Geld zu sparen. Doch wie effizient wurde die Arbeitsteilung betrieben?

In Großbritannien, Mitte des 18. Jahrhunderts, bestand die Textilindustrie aus tausenden Familienbetrieben. Die Herstellung von Textilien wurde von ausgebildeten Spinnern und Webern in mühsamer Handarbeit durchgeführt. Durch ein stetiges Bevölkerungswachstum und einen damit steigenden Bedarf an Textilien in jener Zeit, war die Arbeit per Hand nicht mehr effizient genug. So fanden in den 1760er Jahren Wettbewerbe über Innovationen statt, welche die Produktion beschleunigten. Einen Höhepunkt der gerade stattfindenden Industrialisierung stellt James Hargreaves' Erfindung „Spinning Jenny“ dar (1770). Mit dieser Spinnmaschine wurde es möglich,

¹ *Franklin*, *Advice to a Young Tradesman*, [21 July 1748].

² *Dahrendorf*, *Adam Smith: Der Wohlstand der Nationen*; *Maslow*, *Psychol. Rev.* 1943.

bei Einsatz von weniger Zeit, Arbeitskraft und auch Wissen den Output eines gelernten Spinners zu erreichen bzw. zu übertreffen.³

Wie von Smith erläutert nimmt Arbeitsteilung und Spezialisierung in der Produktion zu. Maschinen wurden stets weiterentwickelt und ersetzt oder unterstützen die Arbeiter und Fabriken verdrängten traditionell arbeitende Manufakturen.⁴ Mit Beginn der Industrialisierung erhielt die Produktivität, Einzug in das Denken der Kaufleute. Kosten für Ressourcen wurden eingespart, Fachpersonal konnte zu einem Großteil durch Hilfskräfte ersetzt und die Produktionsmenge pro Zeiteinheit potenziert⁵ werden. Das Unternehmertum optimierte stets seine Produktionsprozesse und wirtschaftete immer effizienter, sodass der hohe Output der Fabriken Großbritanniens Handelsbeziehungen u. a. mit Deutschland⁶ ermöglichte.

Um 1800 lebten in Deutschland etwa 80% der Menschen auf dem Land und das Verkehrswesen war vergleichsweise schwach entwickelt. In den folgenden vier Jahrzehnten wurden Landstraßen verbessert und die erste deutsche Eisenbahn in Betrieb genommen. Die entstehende Verkehrsinfrastruktur verknüpfte Städte mit Dörfern, Vertreter konnten ihr Vertriebsgebiet vergrößern und die Verteilung von Gütern wurde erheblich verbessert.⁷ Im Jahr 1886 ließ der Unternehmer Ernst Mey seine Textilprodukte in bebilderten Katalogen abdrucken.⁸ Mey war somit in der Lage, Vertreter bzw. Verkaufsgespräche einzusparen und zeitgleich seinen Absatzmarkt um die ländlichen Regionen zu erweitern.

Nach den beiden Weltkriegen griff der Hamburger Kaufmann Werner Otto das Prinzip des Versandhandels wieder auf. 1948 gründete dieser eine Firma um Schuhe zu verkaufen. Drei Jahre später erweiterte Otto seine Produktpalette um Textilien, die über einen Katalog angeboten wurden.⁹ Der Otto-Versandhandel wuchs über die

³ *Walter*, Wirtschaftsgeschichte, S. 61ff.

⁴ *Walter*, Wirtschaftsgeschichte, S. 48.

⁵ *Walter*, Wirtschaftsgeschichte, S. 63.

⁶ *Kreus u. a.*, Industrie/Dienstleistungen, S. 29.

⁷ *Walter*, Wirtschaftsgeschichte, S. 73f, 93.

⁸ *Deutsche Post AG*, Einkaufen 4.0 - Der Einfluss von E-Commerce auf Lebensqualität und Einkaufsverhalten, S. 18.

⁹ *Hasse*, Otto - mit Versandhandel an die Weltspitze (Seite 1).

nächsten vier Jahrzehnte und war, als Vorreiter der Branche, bereits zu Beginn des E-Commerce 1995 mit einem Webauftritt im Internet vertreten.¹⁰

1.1 Problemstellung

Die E-Commerce-Branche hat in den letzten 20 Jahren eine rasante Entwicklung erlebt. Die Anzahl der Internetnutzer betrug im Jahr 1995 ca. 18 Mio.¹¹ und ist bis heute auf etwa 2,9 Mrd.¹² gestiegen. Folglich handelt es sich bei den potentiellen Kunden von Webshops um rund 40 % der Erdbevölkerung.

„[...]The demand for new software is increasing faster than our ability to supply it, using traditional approaches.“¹³

So, wie die Fertigungsanlagen der Textilindustrie immer leistungsfähiger und die Ansprüche an jene komplexer wurden, ist diese Tendenz auch in der Entwicklung neuer Software für Webshops zu erkennen. Durch das Internet ist der freie Markt nach Smith Realität geworden. Um die Kaufkraft der Internetnutzer effizient abschöpfen zu können, verlangen die Unternehmer nach Alleinstellungsmerkmalen in Form von bspw. neuen Webshop-Funktionalitäten oder -innovationen. Daraus resultiert, dass die E-Commerce-Branche sehr schnelllebig, die Entwicklung neuer Software jedoch so zeitintensiv ist, dass diese zum Releasezeitpunkt teilweise bereits veraltet ist.¹⁴

Die dotSource GmbH hat als E-Commerce-Agentur viele Kunden mit Webshops. Steigend werden neue Kunden gewonnen, für die qualitativ hochwertige Lösungen umgesetzt werden.

Die Entwicklung neuer Software bedarf einer Vielzahl kreativer Prozesse. Vom Erdenken eines Konzeptes über die Erstellung der Designs bis hin zur eigentlichen Programmierung und Fertigstellung eines Projektes, werden viele Arbeitsstunden geleistet. Fraglich ist jedoch, wie sich die eingesetzte Zeit der Mitarbeiter zum End-

¹⁰ Hasse, Otto - mit Versandhandel an die Weltspitze (Seite 1).

¹¹ ECC Köln - Online Experts, 10 Jahre E-Commerce – Rückblick des elektronischen Handels.

¹² Statista GmbH, Internetnutzer - Internetnutzung an der Gesamtbevölkerung weltweit | 2014.

¹³ BOEHM, BARRY W.: Software engineering economics, Prentice-Hall advances in computing science and technology series, Englewood Cliffs, N.J: Prentice-Hall 1981, S. 643

¹⁴ SchlöBer, Die 7++ Todsünden der Programmierer typische Fehler in der Software-Entwicklung vermeiden, S. 15f; Thaller, Software-Anforderungen für Webprojekte, S. 28.

produkt bzw. dem verkauften Aufwand verhält. Während in einem Textilverarbeitungsbetrieb, bspw. die Ausbringungsmenge von Hemden pro Mitarbeiter in der Stunde spielend leicht gemessen werden kann, stellt sich in der dotSource GmbH die Frage: Wie kann die Produktivität eines kreativen Prozesses gemessen werden?

1.2 Ziele und Thesen der Untersuchung

THESE 1: Die Produktivität in der Softwareentwicklung kann mit vertretbarem Aufwand und dennoch aussagekräftig gemessen werden.

Ziel dieser Bachelorarbeit ist es, einen Überblick über das Thema Produktivität zu geben und auf die Besonderheiten der Produktivität eines Dienstleistungsunternehmens einzugehen. Weiterhin werden mögliche Messverfahren und Steigerungsmaßnahmen der Produktivität in der Softwareentwicklung aufgezeigt und kritisch gewürdigt.

THESE 2: Die Arbeitsabläufe der dotSource GmbH sind suboptimal. Die Softwareentwickler unterliegen zu häufig Ablenkungen. Zudem ist der Anteil der Berufseinsteiger zu hoch.

Im Praxisteil dieser Arbeit wird der Fragestellung: *Wie ist der Stand der Arbeitszeit-Produktivität in der dotSource GmbH?* nachgegangen. Um diese Frage diskutieren zu können, wird eine zeitliche Erfassung der Mitarbeiteraktivitäten der dotSource GmbH durchgeführt. Im Anschluss erfolgen eine Untersuchung der Firmenzugehörigkeit der Mitarbeiter der dotSource GmbH sowie eine Wettbewerbsanalyse.

1.3 Abgrenzung der Themenstellung

Aufgrund der zeitlichen Limitierung der Bearbeitungsphase und des Umfangs des Themas Produktivität, muss für diese Arbeit ein realistischer Rahmen gesetzt werden.

Den größten Anteil der Mitarbeiter der dotSource GmbH bilden die Softwareentwickler. Um einen möglichst großen Nutzen aus dieser Arbeit zu ziehen und diese in der gegebenen Zeit durchführen zu können, werden diese in der Untersuchung fokussiert. Sämtliche Analysen werden unter Ausschluss der Mitarbeiter aus Marketing, Verwaltung, Vertrieb und IT durchgeführt.

1.4 dotSource GmbH

Das im E-Commerce angesiedelte Unternehmen dotSource GmbH wurde im Jahr 2006 gegründet. Die Agentur bietet ihren Kunden ein umfassendes Leistungsspektrum von der Strategieberatung und E-Commerce-Workshops, Beratung über das passende Shopsystem¹⁵ sowie der Entwicklung und Umsetzung des Onlineshops bis zur weiterführenden Betreuung des Kunden nach dem Live-Gang und ergänzenden Online-Marketing-Konzepten.¹⁶

Der größte Geschäftsanteil der E-Commerce-Agentur dotSource GmbH ist das Projektgeschäft, also die Entwicklung von Softwarelösungen für Onlineversandhäuser, und das Consulting.¹⁷ Als Basis für die Umsetzung von Onlineshops oder einzelnen Features¹⁸ für Onlineshops werden u. a. Standard Shop-Systeme von der Intershop Communications AG, Magento Inc. und hybris AG eingesetzt.

Das Unternehmen erwirtschaftete 2014 5 Mio. € Umsatz. Davon entfielen % auf das Projektgeschäft und % auf das Consulting (vgl. Abbildung 1 und 2).¹⁹

¹⁵ Ein Shopsystem ist ein Programm, das darauf ausgelegt ist, mehrere tausend Produkte und Bestellungen am Tag zu verarbeiten. Aktuelle Shopsysteme sind modular aufgebaut und ermöglichen eine getrennte Weiterentwicklung von Komponenten. Vgl. *Hennig/Köhler*, Ratg. Für Den Online-Händler 2013, S. 12–13.

¹⁶ *dotSource GmbH*, E-Commerce Leistungen | dotSource – die E-Commerce Agentur.

¹⁷ *dotSource GmbH*, Referenzen | dotSource – die E-Commerce Agentur.

¹⁸ Als Feature wird eine Funktionalität einer Software bezeichnet, beispielsweise eine Suche nach Produkten oder eine Bewertungsmöglichkeit für Produkte.

¹⁹ *Bohnenberg*, dotSource Internes Dokument - BCS Unternehmensinformationen.

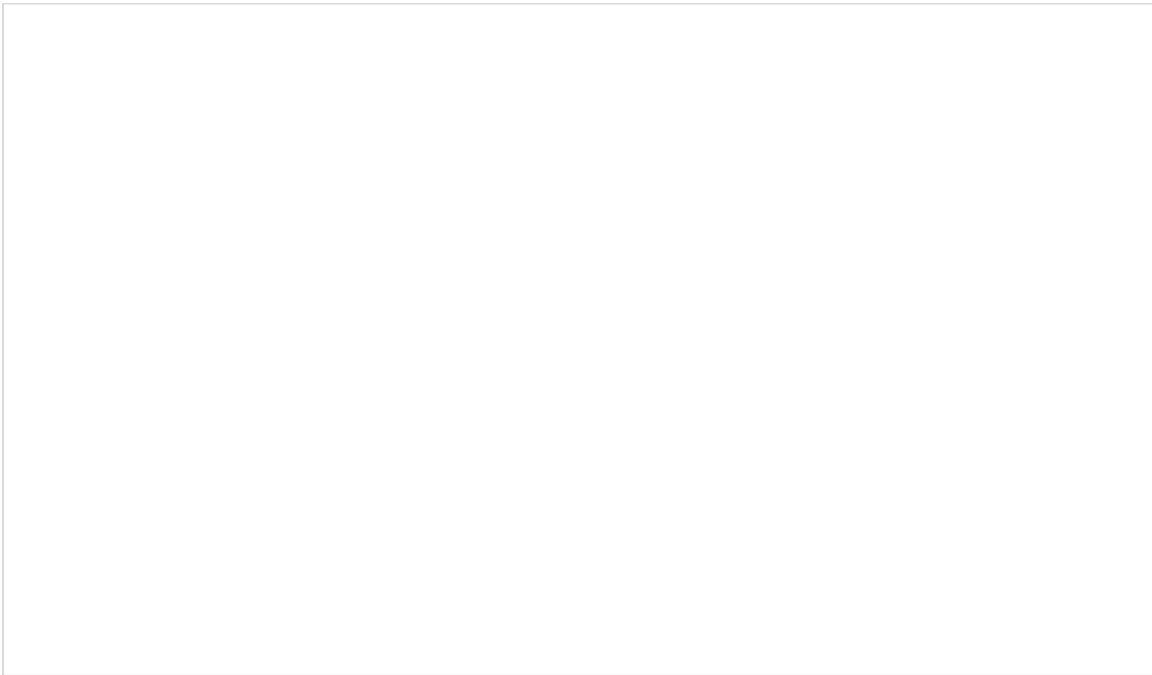


Abbildung 1: Umsatz 2014 nach Geschäftsfeld

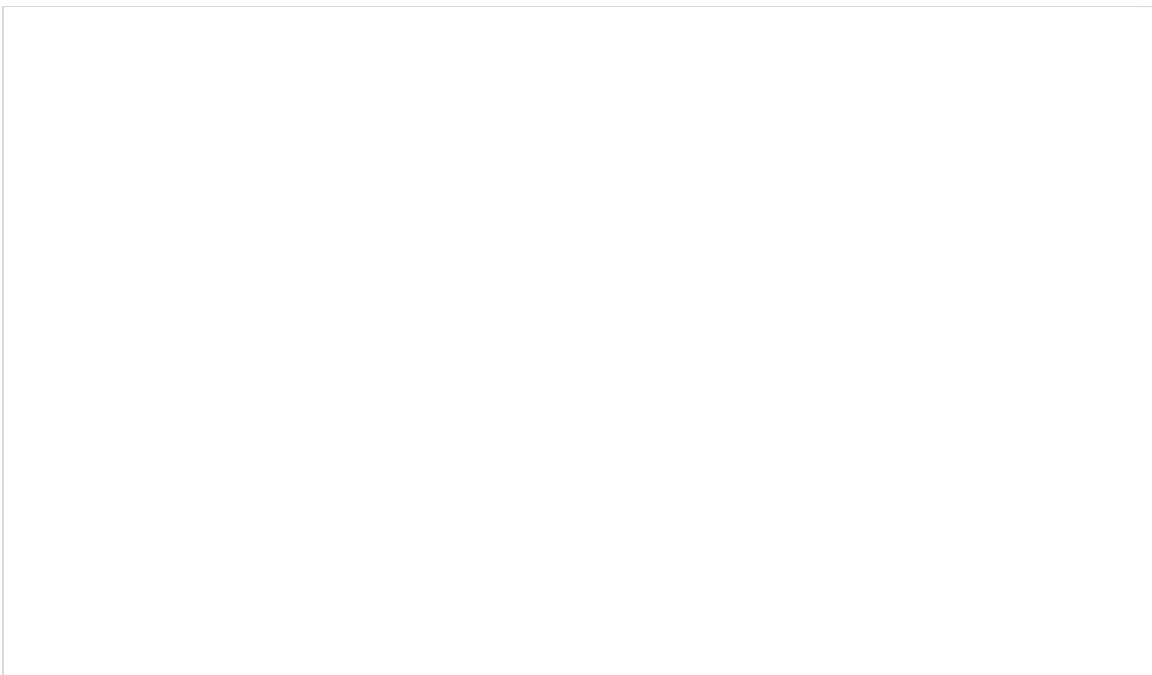


Abbildung 2: Umsatz 2014 nach Shopsystem

Die dotSource GmbH beschäftigt mehr als 100 Mitarbeiter. Mit einer durchschnittlichen Umsatzsteigerung von rund 45 % und einem durchschnittlichen Anstieg der

Mitarbeiterzahl von rund 35 % p. a., ist die dotSource GmbH zu einer der führenden E-Commerce-Agenturen im deutschen Sprachraum gewachsen.²⁰

Organisation der dotSource GmbH

Die Organisationsmethodik einer Unternehmung kann Auswirkungen auf die Produktivität haben. Durch eine komplizierte Organisierung können bspw. Kommunikationswege länger werden und die Produktivität einzelner Mitarbeiter oder ganzer Projektteams können dadurch beeinträchtigt werden.²¹

Die Abteilungen in der dotSource GmbH sind in zwei Bereiche, die kaufmännische (CEO) und die technische Leitung (CTO), gegliedert. Beide Bereiche sind nach einer Linienorganisation strukturiert (vgl. Abbildung 3).

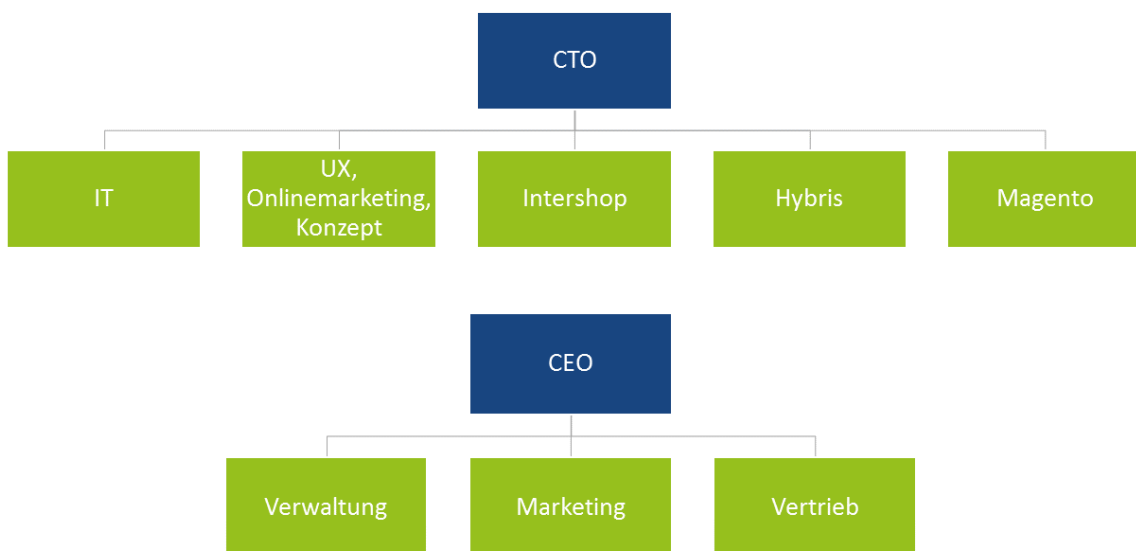


Abbildung 3: Organisationsstruktur dotSource GmbH

Da nur eine Organisationsebene unter der Geschäftsführung existiert, greifen die Vorteile einer Linienorganisation nach Scheld, wie z. B. schnelle Informationsflüsse

²⁰ dotSource GmbH, Zahlen & Fakten | dotSource – die E-Commerce Agentur.

²¹ Scheld, Grundlagen und Informationsmanagement mit Fragen, Aufgaben, Antworten und Lösungen, S. 66–86; Vogenschow u. a., Soft Skills für Softwareentwickler Fragetechniken, Konfliktmanagement, Kommunikationstypen und -modelle, S. V.

und eine klare Abgrenzung von Verantwortlichkeiten, Kompetenzen, Aufgaben und Weisungs- oder Berichtswegen noch besser.²²

Die Softwareentwicklung findet maßgeblich in den Abteilungen Intershop, Hybris und Magento statt. Die jeweiligen Abteilungen sind nochmals in zwei interdisziplinäre Teams mit unterschiedlich hoher Berufserfahrung aufgeteilt. Eine detaillierte Vorstellung eines Softwareentwicklungsteams folgt im Abschnitt 3.1 „Struktur eines Projektteams“ am Beispiel des Teams Intershop 2.

1.5 Betriebswirtschaftliche Bedeutung des Begriffs „Produktivität“

Produktivität ist ein Maß zur Steuerung und Beurteilung von Unternehmungen.²³ Dieser Erfolgsfaktor leitet sich aus dem Wirtschaftlichkeitsprinzip ab²⁴ und wird durch das Verhältnis von Input zu Output (Minimalprinzip) oder Output zu Input (Maximalprinzip) bestimmt. Alleinstehend sind diese Werte nur wenig aussagefähig. Es bedarf eines Vergleichs mit bspw. Vergangenheitswerten der eigenen oder einer Konkurrenzunternehmung.²⁵

Tabelle 1: Begriffe der Produktivität²⁶

	Begriffe	Formel
Gesamtproduktivität	Aufwandsproduktivität	$\frac{\text{Aufwendungen}}{\text{Erträge}}$
	Ertragsproduktivität	$\frac{\text{Erträge}}{\text{Aufwendungen}}$
Teilproduktivitäten	Arbeitsproduktivität	$\frac{\text{Produktionswert}}{\text{Personalaufwand}}$
	Maschinenproduktivität	$\frac{\text{Produktionswert}}{\text{Abschreibungen}}$
	Materialproduktivität	$\frac{\text{Produktionswert}}{\text{Materialaufwand}}$

²² Scheld, Grundlagen und Informationsmanagement mit Fragen, Aufgaben, Antworten und Lösungen, S. 71–72.

²³ Weber, Rentabilität, Produktivität und Liquidität, S. 15.

²⁴ Wildemann, Qualität und Produktivität, S. 15.

²⁵ Weber, Rentabilität, Produktivität und Liquidität, S. 90.

²⁶ In Anlehnung an: Weber, Rentabilität, Produktivität und Liquidität, S. 89.

Die in der Tabelle 1 dargestellten Kennzahlen dienen zur Beurteilung der Ergebnisse eines Fertigungsbetriebes, bspw. der Textilindustrie. Das Ziel solcher Analysen besteht darin, den Wirkungsgrad eingesetzter Produktionsfaktoren zu verbessern, Verschwendungen oder Störungen in Prozessketten zu lokalisieren und produktivitätssteigernde Maßnahmen abzuleiten.²⁷ Die Abbildung 4 verdeutlicht, dass sich die Produktivität aus weiteren Parametern zusammensetzt (die vollständige Mindmap befindet sich im Anhang).



Abbildung 4: Mindmap Produktivität, kompakt²⁸

Neben diesen Parametern sind die Faktoren Qualität und Zeit von komplementärer Bedeutung. Die Verbesserung der Produktqualität wirkt bspw. durch die Reduktion von Nacharbeiten oder Ausschussware produktivitätssteigernd.²⁹ Diese Wirkung lässt sich auch auf die Dienstleistungsbranche (DL) und die Softwareentwicklung übertragen. Eine verschnittene Frisur muss ebenso nachgebessert werden wie ein Fehler im Quellcode. Mit der Folge einer Verschwendung der wichtigsten Ressource: dem Humankapital.

²⁷ Wildemann, Qualität und Produktivität, S. 15.

²⁸ Springer Gabler Verlag, Mindmap Produktivität.

²⁹ Wildemann, Qualität und Produktivität, S. 16.

1.6 Messmethoden der Dienstleistungsbranche

Konzepte zur Produktivitätsbestimmung wurden ursprünglich für den Sachgütermarkt entwickelt und lassen sich nicht ohne weiteres auf die DL-Branche anwenden.³⁰ Ursächlich dafür sind die Immaterialität und Integrativität, also die Integration des Kunden in den Leistungserstellungsprozess,³¹ von Dienstleistungen. Als Voraussetzung, um erbrachte Dienstleistungen messen und bewerten zu können, sind die Größen Input und Output zu spezifizieren. Zur Bestimmung jener Größen gibt es unterschiedliche Messmethoden. Nach Bruhn können u. a. die Formen

- statische und dynamische Messung,
- unternehmens-, kunden- und mitarbeiterbezogene Messung,
- sowie die ereignisorientierte und problemorientierte Messung

unterschieden werden.

Die *statischen und dynamischen Messungen* basieren auf einem Zeit- und Wertorientierten Ansatz. Statische Messungen entsprechen einer Zeitpunkt Betrachtung, d. h. es werden Vergangenheitswerte miteinander verglichen, während dynamische Verfahren auch prognostizierte Werte einbeziehen.

Innerhalb von *unternehmens-, kunden-, und mitarbeiterbezogenen Erhebungen* gilt es, zwischen ökonomischen und psychologischen Maßen zu unterscheiden. Ökonomische Input-Größen können z. B. die Arbeitsstunden pro MA oder die Anzahl der Beschäftigten sein. Wird als Output bspw. der Deckungsbeitrag (DB) herangezogen, kann die Produktivitätskennzahl Deckungsbeitrag pro Mitarbeiter berechnet werden. Bei Produktivitätsmessungen in der DL-Branche sind diese, nur schwer quantifizierbaren, psychologischen Größen von hoher Bedeutung. Bspw. könnte eine hohe Mitarbeiterzufriedenheit (anhand von Fluktuation und Fehlzeiten messfähig) produktionsfördernd und somit produktivitätssteigernd wirken.³²

³⁰ Bruhn/Hadwich, in: Dienstleistungsproduktivität - Einführung in die theoretischen und praktischen Problemstellungen (Band 1), S. 6.

³¹ Diller, Vahlens großes Marketinglexikon [in zwei Bänden], S. 285 f.

³² Deede/Mühlhan, Arbeitsmotivation & Arbeitszufriedenheit, S. 19 ff; Bruhn/Hadwich, in: Dienstleistungsproduktivität - Einführung in die theoretischen und praktischen Problemstellungen (Band 1), S. 15.

Marketing- und Vertriebsmaßnahmen können bspw. zu einer erhöhten Kundenakquise führen, welche die Produktivität in einem Zeitraum steigern kann. Solche periodischen Gegebenheiten werden in den *ereignisorientierten Messansätzen* berücksichtigt. Bei den *problemorientierten Messungen* werden diejenigen Abschnitte in den Leistungserstellungsprozessen analysiert, bei welchen es bspw. zu Kundenbeschwerden kam. Gelingt es einem Unternehmen diese Felder zu verbessern, werden die Fehlerquote und Nacharbeiten reduziert und die DL-Produktivität steigt.³³

Die Ermittlung von Input- und Output-Größen erfolgt mittels interner und externer Informationsquellen. Intern eignen sich das betriebliche Rechnungswesen sowie ein ERP-System zur Informationsbeschaffung und für die externe Informationsbeschaffung werden auf u. a. Kundenbefragungen³⁴, Branchenberichte und Wirtschaftsdatenbanken³⁵, wie GBI-Genios oder Hoppenstedt, zurück gegriffen. Weitere Beispiele für Maßgrößen und Quellen sind der Arbeit angehängt (vgl. Abbildung 29).

Unternehmen der Dienstleistungsbranche unterliegen den drei Wettbewerbsdimensionen, Qualitäts-, Kosten- und Zeitwettbewerb. Diese drei Dimensionen stehen in einem konfliktären Verhältnis zueinander. Eine hohe Qualität benötigt mehr Zeit und mehr Zeit verursacht höhere Kosten. Dennoch stellen diese Faktoren nach Bruhn und Hadwich den „zentralen Treiber für die Produktivität eines Unternehmens dar“³⁶ und lassen sich als Kategorien für Produktivitätskennzahlen nutzen (vgl. Tabelle 2).

³³ Bruhn/Hadwich, in: Dienstleistungsproduktivität - Einführung in die theoretischen und praktischen Problemstellungen (Band 1), S. 16.

³⁴ Bruhn/Hadwich, in: Dienstleistungsproduktivität - Einführung in die theoretischen und praktischen Problemstellungen (Band 1), S. 16.

³⁵ Stoetzer, Erfolgreich recherchieren, S. 115 ff.

³⁶ Bruhn/Hadwich, in: Dienstleistungsproduktivität - Einführung in die theoretischen und praktischen Problemstellungen (Band 1), S. 18.

Tabelle 2: Produktivitätskennzahlen Dienstleistungsbranche³⁷

Qualitätsindizes	Kostenindizes	Zeitindizes
$\frac{\text{Anzahl Neukunden}}{\text{Anzahl Mitarbeiter}}$	$\frac{\text{Kundenakquisitionserträge}}{\text{Kundenakquisitionsaufwände}}$	$\frac{\text{Anzahl Kundenberatungen}}{\text{Arbeitsstunden MA}}$
$\frac{\text{Anzahl Aufträge}}{\text{Anzahl Angebote}}$	$\frac{\text{Kundenbindungsgewinn}}{\text{Kundenbindungskosten}}$	$\frac{\text{Anzahl Neukunden}}{\text{Dauer Akquise}}$
$\frac{\text{Umsatz mit zufriedenen Kunden}}{\text{Anzahl Kunden gesamt}}$	$\frac{\text{Umsatz}}{\text{Anzahl Mitarbeiter}}$	$\frac{\text{Vertragsabschluss}}{\text{Dauer Beratungsgespräch}}$
$\frac{\text{Anzahl zufriedener Kunden}}{\text{Anzahl Kunden gesamt}}$	$\frac{\text{DB eines Kunden (-segments)}}{\text{Aufwand Kundenbetreuung}}$	$\frac{\text{Break Even eines Kunden}}{\text{Gesamtzeit Betreuung}}$

Die Produktivität von (Dienst-)Leistungserstellungsprozessen wird von zahlreichen Determinanten beeinflusst. Durch die Integrativität von Dienstleistungen beeinflusst der Kunde die Produktivität in hohem Maße. Der Kunde erbringt Zuarbeiten, z. B. unternehmensinterne Daten, die in ein neues System integriert werden sollen. Er wird in Folge dessen aktiv in den Leistungserstellungsprozess eingebunden und hat direkten Einfluss auf die Effizienz und Produktivität des DL-Unternehmens.³⁸ In der Abbildung 5 sind weitere Beispiele für Einflussfaktoren, welche die Produktivität tangieren, dargestellt.

³⁷ Bruhn/Hadwich, in: Dienstleistungsproduktivität - Einführung in die theoretischen und praktischen Problemstellungen (Band 1), S. 19.

³⁸ Bruhn/Hadwich, in: Dienstleistungsproduktivität - Einführung in die theoretischen und praktischen Problemstellungen (Band 1), S. 21 f.

Determinanten der Dienstleistungsproduktivität					
Unternehmens-bezogen	Produkt-bezogen	Innovations-bezogen	Kunden-bezogen	Mitarbeiter-bezogen	Prozess-bezogen
<ul style="list-style-type: none"> ■ Unternehmensgröße ■ Unternehmensimage/Reputation ■ Internationalisierungsstrategie ■ Marktstellung ■ Kostenstruktur ■ Vertriebsorganisation ■ u.a.m. 	<ul style="list-style-type: none"> ■ Leistungsprogramm ■ Qualitätsniveau ■ Preisniveau ■ Komplexität ■ Attraktivität (USP) ■ Innovationsgehalt ■ Lebenszyklus ■ u.a.m. 	<ul style="list-style-type: none"> ■ Leistungsverbesserung ■ Komponentisierung ■ Innovationsprozess ■ Kundenintegration ■ Hybride Problemlösungen ■ u.a.m. 	<ul style="list-style-type: none"> ■ Kundenstruktur ■ Kundenintegration ■ Anzahl der Kunden ■ Heterogenität der Kundengruppen ■ Kundenerwartung ■ Informationsverhalten ■ Serviceanforderungen ■ u.a.m. 	<ul style="list-style-type: none"> ■ Anzahl Mitarbeiter ■ Mitarbeiterqualifikation ■ Personalmanagement/Personalentwicklung ■ Erwartungshaltung der Mitarbeitenden ■ Mitarbeiterzufriedenheit ■ u.a.m. 	<ul style="list-style-type: none"> ■ Zeitdauer des Dienstleistungserstellungsprozesses ■ Anzahl der Beteiligten des Prozesses ■ Umfang der Kundenintegration ■ Grad der Interaktion ■ Erwartungen der Prozessbeteiligten ■ u.a.m.

Abbildung 5: Determinanten der Dienstleistungsproduktivität³⁹

Nach Bergmann und Daub ist das Ziel einer Produktivitätssteigerung, die Zukunftsfähigkeit von DL-Unternehmen zu sichern. Die Mitarbeiter sollen von „mühsamer und stupider“⁴⁰ Arbeit befreit werden, um somit Kapazitäten für die Kompetenzentwicklung zu schaffen (vgl. auch Abbildung 5, Mitarbeiterqualifikation). Solche Entlastungen der Mitarbeiter finden durch Prozessautomatisierungen statt. Ein aktuelles Beispiel ist der Einsatz von RFID⁴¹-Armbändern auf dem Hurricane-Festival. Dieses wurde als Zahlungsmittel eingesetzt und entlastete somit die Catering-Dienstleister um den Zahlungsprozess.⁴² Ein weiteres Beispiel stellt ein Feature des, sich aktuell in der Entwicklung befindlichen, Music Store Onlineshops dar. Der sogenannte Shop im Shop ist eine Softwarelösung⁴³, welche die Funktionen eines Verkäufers digital abbildet und Fragen zu Bedürfnissen und Produkten automatisiert klärt. Somit werden wiederum Mitarbeiter im Callcenter und evtl. auch im stationären Geschäft entlastet. Wei-

³⁹ Bruhn/Hadwich, in: Dienstleistungsproduktivität - Einführung in die theoretischen und praktischen Problemstellungen (Band 1), S. 21.

⁴⁰ Bergmann/Daub, in: Produktivität von Dienstleistungen - mehr des Selben oder mehr des Anderen?, S. 97.

⁴¹ Abk. für Radio Frequency Identification. RFID bezeichnet eine drahtlose Schnittstelle zur Kommunikation mit IT-Systemen. Springer Gabler Verlag, Stichwort: Software Engineering, online im Internet:

⁴² Krulle, Protest wegen Bezahl-Chips beim Hurricane-Festival.

⁴³ Arnold, dotSource Internes Dokument - BCS Spezifikation Shop in Shop.

tere mögliche Nebeneffekte sind u. a. eine Minderung der Transaktionskosten⁴⁴, steigende Absätze durch die erhöhte Ablaufschnelligkeit sowie eine stabile Qualität der Prozesse⁴⁵. Die mitarbeiterbezogenen Faktoren sind in der Konsequenz als sehr wichtige Determinanten der Dienstleistungsproduktivität anzusehen.

⁴⁴ *Bergmann/Daub*, in: Produktivität von Dienstleistungen - mehr des Selben oder mehr des Anderen?, S. 89.

⁴⁵ *Oswald*, in: Deutscher Dialogmarketing Verband e.V., Automatisierung von Kundenmanagementprozessen in der Dienstleistungsbranche, S. 150.

2 Produktivität in der Softwareentwicklung

„99 Bottles of Beer“ ist amerikanischer Folksong mit 100 Strophen. Die Ausgabe des Textes wurde in über 1500 Programmiersprachen umgesetzt.⁴⁶ Wenn bei Google nach dem Schlagwort „Software Engineering“ gesucht wird, fördert dies rund 222 Millionen Treffer zutage. Daraufhin stellt sich die Frage, wie die Angehörigen einer, sich innerhalb von 40 Jahren rasant entwickelten,⁴⁷ jungen Disziplin, den Überblick behalten und produktiv sein können. In den folgenden Kapiteln wird auf Aktivitäten und Modelle zur Vorgehensweise in der Softwareentwicklung eingegangen und die Probleme dabei beleuchtet.

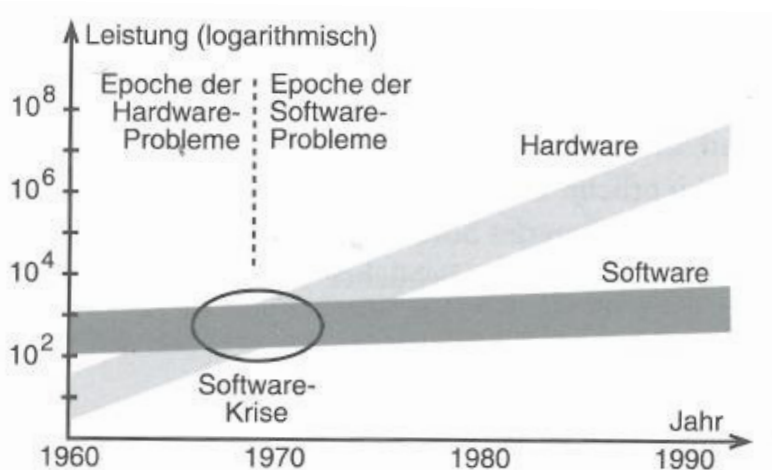


Abbildung 6: Entwicklung der Hard- und Software⁴⁸

Software-Engineering ist eine „wissenschaftliche Disziplin, die sich mit der Entwicklung, dem Einsatz und der Wartung von Software befasst.“⁴⁹ Mit der Einführung des Engineering-Begriffs sollte die Software-Entwicklung aus dem Künstlertum geführt werden. In Abbildung 6 ist die Leistungssteigerung von Hard- und Software darge-

⁴⁶ <http://www.99-bottles-of-beer.net/>

⁴⁷ Plewan/Poensgen, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 7.

⁴⁸ Ludewig/Lichter, Software Engineering, S. 44.

⁴⁹ Springer Gabler Verlag, Stichwort: Software Engineering, online im Internet:

stellt. Daraus lässt sich ableiten, dass sich die Ansprüche des Software-Engineering noch nicht durchgesetzt haben.⁵⁰

Die Softwareentwicklung wird häufig im Rahmen eines Projekts organisiert.⁵¹ Die fünf Basis-Aktivitäten in der Softwareentwicklung sind Anforderungsanalyse, Design, Implementierung, Test und Auslieferung (vgl. Tabelle 3). Weiterhin sind noch die Querschnittsaktivitäten Projekt-, Qualitäts- und Änderungs- / Konfigurationsmanagement zu nennen.⁵²

Tabelle 3: Typische Ergebnistypen der Basisaktivitäten⁵³

Basisaktivität	Typische Ergebnistypen
Anforderungsanalyse	<ul style="list-style-type: none"> • Anforderungsliste • Anforderungsspezifikation • GUI-Prototyp...
Design	<ul style="list-style-type: none"> • Architekturdokument • DV-Konzept...
Implementierung	<ul style="list-style-type: none"> • Code • Systemdokumentation • Benutzerhandbuch
Test	<ul style="list-style-type: none"> • Testberichte
Auslieferung	<ul style="list-style-type: none"> • Installation beim Kunden

Diese Aktivitäten finden sich in etablierten Vorgehensmodellen des Software-Engineerings wieder. Die Vorgehensmodelle werden in die vier Kategorien Wasserfallmodell, inkrementelle-, iterative- und agile Entwicklung eingeteilt.⁵⁴ Aus Relevanz-

⁵⁰ Ludewig/Lichter, Software Engineering, S. 47f.

⁵¹ Ludewig/Lichter, Software Engineering, S. 89.

⁵² Plewan/Poensgen, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 8 ff.

⁵³ Plewan/Poensgen, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 11.

⁵⁴ Plewan/Poensgen, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 13.

gründen wird nicht jedes Vorgehensmodell vorgestellt. Stattdessen wird am Beispiel der Methode Scrum (vgl. Abschnitt 3.4.) kurz auf die agile Entwicklung eingegangen. Ein Scrum-Projekt wird in mehrere Sprints eingeteilt, an deren Ende jeweils das Release einer neuen Softwareversion erfolgt.⁵⁵ Für jeden Sprint werden zu bearbeitende Features priorisiert und dem Team zugewiesen. Der wichtigste Unterschied zwischen klassischen und agilen Methoden sind die flexiblen Prozesse.⁵⁶ Durch die regelmäßige Planung, sind bei einem Scrum-Projekt, Anforderungsänderungen durch den Kunden weniger kritisch als bei klassischen Methoden.

Probleme in der Softwareentwicklung

„Aus Zeitnot sind mehr Software-Projekte schief gegangen als aus allen anderen Gründen zusammengenommen.“⁵⁷

DeMarco und Brooks sehen eine mangelnde Fähigkeit von Projektbeteiligten, die zeitlichen Aufwände für das Entwickeln einer Software richtig zu schätzen.⁵⁸ Das Schätzen ist ein wichtiger Bestandteil der Anforderungsanalyse und der tatsächlichen Implementation in einem Softwareprojekt. Basierend auf den Ergebnissen werden Angebote erstellt und der Zeitplan geprüft. Ein zu niedrig geschätzter Aufwand führt oft zu finanziellen Problemen oder Verlusten, während ein zu hoch geschätzter Aufwand das Zustandekommen eines Projektes verhindern kann.⁵⁹ Nach DeMarco liegen schlechte Schätzungen u. a. darin begründet, dass die Fähigkeit des Schätzens nicht gelernt wird.⁶⁰ Dafür können firmenpolitische Faktoren ursächlich sein. Ein solcher Faktor ist bspw. das Nichtdurchführen einer Folgeschätzung um keine Fehler eingestehen zu müssen oder die Anpassung der Schätzung an Erwartungen oder

⁵⁵ Ludewig/Lichter, Software Engineering, S. 228 ff.

⁵⁶ Ludewig/Lichter, Software Engineering, S. 183.

⁵⁷ Brooks, Vom Mythos des Mann-Monats, S. 14.

⁵⁸ DeMarco, Was man nicht messen kann, kann man nicht kontrollieren, S. 37 ff; Brooks, Vom Mythos des Mann-Monats, S. 14 f.

⁵⁹ Thaller, Software-Anforderungen für Webprojekte, S. 40 f; Plewan/Poensgen, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 43.

⁶⁰ DeMarco, Was man nicht messen kann, kann man nicht kontrollieren, S. 37 f.

erlaubte Verzögerungen.⁶¹ Durch das regelmäßige Einfordern und Controlling von Schätzungen ist es möglich, die Qualität von Schätzungen mittelfristig zu verbessern.

Plewan und Poensgen vermuten ein starkes Defizit hinsichtlich der Nutzung anerkannter Vorgehensmodelle in der deutschen Softwareindustrie. So schätzen sie, dass bis zu 50 % der Unternehmen keine oder eine Methode nur rudimentär anwenden.⁶² Als problematisch ist ebenfalls die Professionalität in der Softwareentwicklung anzusehen. Wesentliche Ausprägungen einer professionellen Disziplin sind u. a. ein eindeutiger Konsens von Methoden mit denen gearbeitet wird, zuverlässige Ergebnisse mit hoher Qualität sowie ein festgelegter Ausbildungsweg und somit die Existenz klarer und anerkannter Anforderungen an eine Fachperson.⁶³ Es bedarf folglich auf die Bedürfnisse der Unternehmen zugeschnittene, homogene Ausbildungen⁶⁴ im Software-Engineering. Zum eingehenderen Studium der Professionalität und Verbesserungsmöglichkeiten, sei auf die Werke „Clean Coder“⁶⁵ und „Der Weg zur professionellen IT“⁶⁶ verwiesen.

2.1 Ermittlung der Produktivität in der Softwareentwicklung

Im ersten Kapitel wurden die industrielle Produktivität sowie die Ansätze in der DL-Branche beschrieben und auch schon Verbindungen zur Softwareentwicklung hergestellt. In der Softwareentwicklung gibt es nach Plewan und Poensgen zwei Betrachtungsmöglichkeiten, die Wirtschaftlichkeit (Effektivität) und die Leistungsproduktivität (Effizienz).⁶⁷ Mit der Wirtschaftlichkeit ist das Verhältnis von Ertrag und Kosten eines Softwareprojektes gemeint. Es werden die Fragen nach der Effektivität beleuchtet. Demgemäß „*Was wird entwickelt?*“ und „*Kann das Projekt gewinnbringend abge-*

⁶¹ DeMarco, Was man nicht messen kann, kann man nicht kontrollieren, S. 38 – 43.

⁶² Plewan/Poensgen, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 19 f.

⁶³ Plewan/Poensgen, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 17 f.

⁶⁴ Plewan/Poensgen, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 25; Ludewig/Lichter, Software Engineering, S. 76.

⁶⁵ Martin, Clean Coder.

⁶⁶ Foegen u. a., Der Weg zur professionellen IT.

⁶⁷ Plewan/Poensgen, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 31 f.

geschlossen werden?“. Es geht somit um die Kennzahl Return on Investment (ROI). Die Leistungsproduktivität stellt sich der Frage „*Wie wird entwickelt?*“. Folglich wird die Effizienz eines Projektes, also wie „kostengünstig, termintreu und mit hoher Qualität entwickelt [...] werden kann“⁶⁸, gemessen.

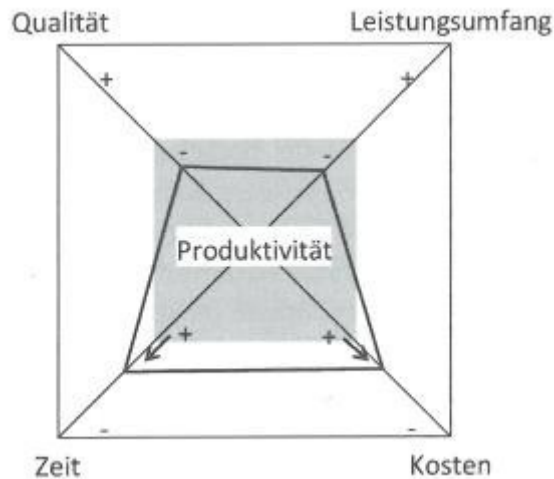


Abbildung 7: Teufelsquadrat nach Sneed⁶⁹

Im Abschnitt 1.5 wurde bereits der Zusammenhang zwischen dem Faktor Qualität und Kosten angeschnitten. In der Abbildung 7 sind diese Faktoren neben Leistungsumfang und Zeit wiederzufinden. Harry Sneed hat mit dem *Teufelsquadrat* die Abhängigkeiten dieser Optimierungsziele untereinander demonstriert. Bei konstanter⁷⁰ Produktivität (Quadrat in der Mitte), hat die Verschiebung zu einem Ziel direkten Einfluss auf die anderen Ziele.⁷¹ Es kann also festgehalten werden, dass das Quadrat in der Mitte die Leistungsfähigkeit eines Entwicklungsteams abbildet. Wenn ein Softwareprojekt in weniger Zeit und zu geringeren Kosten abgeschlossen werden soll,

⁶⁸ Plewan/Poensgen, *Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis*, S. 32 – 34.

⁶⁹ Plewan/Poensgen, *Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis*, S. 35.

⁷⁰ Nach Sneed ist die Produktivität eines Projektes stets als konstant anzusehen, da Steigerungen nicht kurzfristig geschehen können und evtl. Motivationsschübe durch -defizite wieder ausgeglichen werden. *Sneed u. a., Software in Zahlen*, S. 230.

⁷¹ Plewan/Poensgen, *Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis*, S. 35.

kann das, ohne eine hinreichend hohe Leistungsproduktivität,⁷² nur zu Lasten der Qualität und des Leistungsumfanges passieren.⁷³ Plewan und Poensgen weisen auf die Möglichkeit der Steigerung einer „imaginären“ Produktivität hin. Damit ist eine einseitige Steigerung der Arbeitsleistung durch Überstunden bei Terminverzug gemeint.⁷⁴ „Personen unter Zeitdruck arbeiten nicht besser; sie arbeiten nur schneller.“⁷⁵ Eine exorbitante Nutzung von Überstunden wird negative Seiteneffekte auf die Produktqualität, Kundenzufriedenheit und Kosten (Gehälter, Bugfixes⁷⁶) haben.⁷⁷

Das statistische Amt Eurostat ermittelte einen Zusammenhang zwischen den geleisteten Arbeitsstunden und der damit erreichten Arbeitsproduktivität im Ländervergleich. Demnach besteht eine mittlere negative Korrelation (-0,58) zwischen beiden Faktoren, d. h., dass in Ländern mit niedriger Produktivität eher länger gearbeitet wird und umgekehrt (siehe Abbildung 30 im Anhang).⁷⁸ Das wesentliche Maß der Arbeitsproduktivität ist das Bruttoinlandsprodukt.⁷⁹ Dieses ist die Summe der Bruttowertschöpfung einer Volkswirtschaft, also die Produktionswerte abzgl. der Vorleistungen.⁸⁰ Eine hohe Arbeitsproduktivität entsteht demnach bei der Herstellung von vielen Produkten hoher Qualität, in kurzer Zeit und zu niedrigen Kosten. Es besteht folglich eine Verbindung zwischen der Produktivität und Qualität.

⁷² *Plewan/Poensgen*, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 35.

⁷³ vgl. auch *DeMarco/Lister*, Wien wartet auf Dich! produktive Projekte und Teams ; [mit sechs neuen Kapiteln], S. 14 ff; *Plewan/Poensgen*, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 40 ff.

⁷⁴ *Plewan/Poensgen*, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 40; *DeMarco/Lister*, Wien wartet auf Dich! produktive Projekte und Teams ; [mit sechs neuen Kapiteln], S. 13 ff.

⁷⁵ *DeMarco/Lister*, Wien wartet auf Dich! produktive Projekte und Teams ; [mit sechs neuen Kapiteln], S. 18.

⁷⁶ BUG: Bezeichnung für einen Softwarefehler, also einer Abweichung des Softwareverhaltens von der Anforderung. vgl. *Plewan/Poensgen*, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 91.

⁷⁷ *Ebert/Dumke*, Software measurement, S. 294.

⁷⁸ *Eurostat*, Regionen: Statistisches Jahrbuch 2006, S. 76.

⁷⁹ Die Arbeitsproduktivität (€ pro Stunde) errechnet sich aus BIP / (Beschäftigte x Wochenarbeitszeit x jährliche Arbeitswochen). vgl. *Eurostat*, Regionen: Statistisches Jahrbuch 2006, S. 77.

⁸⁰ *Springer Gabler Verlag*, Stichwort: Bruttoinlandsprodukt (BIP), online im Internet:

In der Software-Engineering Literatur werden überwiegend Messmethoden, wie die Function Point Analyse (FPA) und dem Quellcodeumfang „Lines of Code - Metrik“ (LoC) präsentiert.⁸¹ Auf diese wird im Folgenden genauer eingegangen.

Die Leistungsproduktivität (L_p) wird, anal. zur industriellen Produktivität, aus dem Verhältnis $L_p = \frac{Output}{Input} = \frac{Projektergebnis}{Projektaufwand}$ berechnet. Als Projektaufwand finden i.d.R. Größen wie Personentage, -monate oder Kosten Anwendung. Der Output eines Softwareprojektes, also der Leistungsumfang, wird entweder über konstruktive oder funktionale Messgrößen definiert.⁸² Unabhängig von der gewählten Methode ist es nötig, standardisierte Vorlagen und Regeln aufzustellen, damit Messungen effizient durchgeführt werden und valide Vergleiche mit anderen Projekten stattfinden können.⁸³

Die LoC-Metrik ist eine konstruktive Messgröße, welche die geschriebenen Zeilen Quelltext pro Aufwandseinheit beschreibt. Wie in Kapitel 2 beschrieben, gibt es zahlreiche Programmiersprachen. Soll die L_p anhand der LoC bestimmt werden, können beträchtliche Schwierigkeiten bei der Definition einer Code-Zeile aufkommen, da diese Metrik mit allen, für den Urheber relevanten Sprachen anwendbar sein muss.⁸⁴ Die wichtigsten funktionalen Messgrößen sind die Function Points (fp) und die Use-Case Points (ucp). Sie beschreiben die „fachlich funktionalen Anforderungen“ des Softwareprojektes, bspw. die Zahlungsmöglichkeiten eines Onlineshops. Das Projekt ist umso produktiver, je mehr Function Points in einem Personenmonat umgesetzt wurden. Die Aussage über die L_p ist je nach Wahl einer konstruktiven oder funktionalen Methode eine andere. Man kann dies mit dem Häuserbau vergleichen. Die konstruktive Methode würde bewerten, wie viele Ziegelsteine pro Zeiteinheit verbaut wurden

⁸¹ vgl. u. a. *Ebert/Dumke*, Software measurement, S. 183 ff; *Ludewig/Lichter*, Software Engineering, S. 128 ff; *Plewan/Poensgen*, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 51 ff.

⁸² *Plewan/Poensgen*, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 33.

⁸³ *Ebert/Dumke*, Software measurement, S. 120.

⁸⁴ *Ludewig/Lichter*, Software Engineering, S. 303.

während die funktionale bspw. die entstandene Wohnfläche beschreiben würde.⁸⁵ An diesem Beispiel kann gefolgert werden, dass die Anzahl der geschriebenen Codezeilen zu wenig über den eigentlichen Leistungsumfang aussagt. In der professionellen Softwareentwicklung gilt es zudem, den Quelltext so kurz, einfach und verständlich wie möglich zu halten.⁸⁶ Aus diesen beiden Gründen sind funktionale Messgrößen für die Gesamtbewertung eines Projektes vorzuziehen.

2.1.1 Function Point Analyse

Die International Organization for Standardization (ISO) beschreibt im Standard ISO/IEC 14143 das grundsätzliche Reglement zur Messung des Leistungsumfanges einer Software.⁸⁷ Anwendungsbezogene Vorgehensweisen sind in weiteren Standards geregelt, darunter die gemäß Plewan und Poensgen in Deutschland sowie international am meisten verbreitete Methode Function Point Analyse nach IFPUG (ISO/IEC 20926:2009).⁸⁸ Diese Methode wurde 1979 von Allan Albrecht (IBM) publiziert und wird von der International Function Point Users Group (IFPUG) weitergeführt.⁸⁹

Bei der FPA wird Software durch die Quantifizierung der Nutzeranforderungen gemessen. Es werden somit die Funktionen berücksichtigt, die für den Anwender sichtbar sind und die Geschäftsprozesse des Nutzers unterstützen.⁹⁰ Ein durch die Software abgebildeter Geschäftsprozess wird in dessen Einzelaktivitäten, den sog. *Elementarprozessen*, zerlegt und gruppiert. Als Elementarprozess wird die „kleinste sinnvolle und [...] in sich abgeschlossene Funktion“⁹¹ einer Software bezeichnet,

⁸⁵ *Plewan/Poensgen*, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 33.

⁸⁶ *Ludewig/Lichter*, Software Engineering, S. 456 ff.

⁸⁷ *International Organization for Standardization*, ISO/IEC 14143-1:2007 - Information technology -- Software measurement -- Functional size measurement.

⁸⁸ *Plewan/Poensgen*, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 51 f.

⁸⁹ *IFPUG International Function Point Users Group*, About Function Point Analysis - IFPUG.

⁹⁰ *Plewan/Poensgen*, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 52.

⁹¹ *Plewan/Poensgen*, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 53.

bspw. die Eingabe einer Lieferadresse, die Berechnung des Warenkorbs oder die Ausgabe eines Suchergebnisses. Ein Elementarprozess wird einer der Gruppen *Transaktionen (Ein- und Ausgaben, Abfragen)* oder *Verwaltung von Datenbeständen* zugewiesen und nach Komplexität gewichtet.⁹² Daraus wird der Function Point ermittelt. Bspw. die Ausgabe eines Suchergebnisses kann je nach Komplexität der Nutzeroberfläche, bzw. Schnittstelle über 4 bis 7 fp verfügen.⁹³ Die Summe aller fp ergibt die Functional Size, also den Leistungsumfang einer Software.

Zusammenfassend ist festzuhalten, dass die FPA grundsätzlich nicht von der technischen Realisierung abhängt und eine konsistente Vergleichsmöglichkeit zwischen verschiedenen Projekten bietet.⁹⁴ Die Anwendung bringt auch zusätzlichen Nutzen, bspw. können FPAs dem Review von Anforderungsdokumenten dienlich sein. Durch die Aufspaltung der Funktionen in Elementarprozesse sind Inkonsistenzen und Lücken mit hoher Wahrscheinlichkeit zu identifizieren.⁹⁵ Ferner sind mit der FPA verschiedene Metriken umsetzungsfähig. Beispielsweise kann die Produktivität (Aufwand / fp), Qualität (Bugs / fp) oder die Effizienz (Kosten / fp) berechnet und ausgewertet werden.⁹⁶

2.1.2 Weitere Messmethoden

Basierend auf die FPA wurden noch weitere Messverfahren entwickelt. Der funktionale Umfang von Software kann auch durch die Anzahl und Komplexität von Use Cases⁹⁷ beschrieben und bewertet werden. Die *Use Case Point Methode* wurde urspr. für Aufwandsschätzungen entwickelt und um Ansätze aus der FPA und dem COCOMO Modell erweitert. Diese Methode ist bei Projekten, die anhand von Use

⁹² Plewan/Poensgen, *Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis*, S. 53.

⁹³ Sneed u. a., *Software in Zahlen*, S. 39.

⁹⁴ *International Organization for Standardization, ISO/IEC 20926:2009 - Software and systems engineering -- Software measurement -- IFPUG functional size measurement method 2009; Hürten, Function-Point-Analysis - Theorie und Praxis*, S. 12.

⁹⁵ Plewan/Poensgen, *Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis*, S. 54.

⁹⁶ *IFPUG International Function Point Users Group, What Are Function Points – Fact Sheet - IFPUG.*

⁹⁷ *USE CASE: Konzept zur Anforderungsbeschreibung einer Software, mittels Anwendungsfällen aus der Perspektive des Endnutzers. vgl. Ludewig/Lichter, Software Engineering*, S. 536.

Cases beschrieben sind, gut anzuwenden und die Ergebnisse sind mit der Functional Size vergleichbar. Jedoch ist die Methode nicht nach ISO 14143 normiert und es muss sichergestellt sein, dass die Use Cases stets über denselben Detailgrad verfügen.⁹⁸

Besonders in agilen Projektorganisationen findet die *Story Point Methode* Anwendung. Story Points sind für die unternehmens- / wettbewerbsweite Produktivitätsanalyse ungeeignet, da ein Story Point eine relative Größe ist. Sie bezeichnen den Umfang eines Arbeitspaketes im Vergleich zu anderen innerhalb eines Projektes.⁹⁹

Ein weiteres erwähnenswertes Modell sind die *Testcase Points*. Testfälle lassen sich u. a. aus der Anforderungsdokumentation oder dem Quelltext ableiten. Als Testfall wird die Abfolge von Aktionen entlang eines einmaligen Weges bezeichnet. Daraus folgt, dass an jeder Bedingung und jeder Schleife im Code ein neuer Weg bzw. Testfall entsteht. Die Erstellung von Testcase Points ergibt ein relativ genaues Maß für den Softwareumfang.¹⁰⁰ Weiterhin sind, bei Einsatz der Methode, Synergieeffekte hinsichtlich der Zusammenarbeit zwischen QA- und Entwicklungsabteilungen sowie eine Senkung von Fehlerquoten denkbar.¹⁰¹

2.1.3 Zwischenfazit

Um Softwareprojekte zu bewerten, eignen sich sowohl konstruktive als auch funktionale Messmethoden. Ungeachtet eines Disputs, welche Art besser ist, sollte der Einsatz einer bestimmten Metrik immer dann erfolgen, wenn eine ausreichend hohe „Verwendbarkeit, [...] Güte und [...] Aussagekraft“ besteht.¹⁰² Für die Anfangsphase eines Projektes eignen sich funktionale Methoden am ehesten, gesetzt dem Fall, es

⁹⁸ Ludewig/Lichter, Software Engineering, S. 132; Plewan/Poensgen, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 56 f.

⁹⁹ Plewan/Poensgen, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 57.

¹⁰⁰ Sneed u. a., Software in Zahlen, S. 42 f.

¹⁰¹ vgl. auch Vigneschow u. a., Soft Skills für Softwareentwickler Fragetechniken, Konfliktmanagement, Kommunikationstypen und -modelle, S. 263 f.

¹⁰² Plewan/Poensgen, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 153.

sind funktionale Anforderungen bekannt. Am Ende eines Projektes sind LoC sinnvoll zu erheben. Aus den LoC kann bspw. ein Verhältnis zu den Function Points erhoben werden (LoC / fp). Durch die gewonnen Erfahrungswerte können der Umfang des Quellcodes und die Kosten der Folgeprojekte noch genauer bestimmt werden.¹⁰³ Die oben behandelten Metriken stellen einen Auszug dar. In der Literatur werden noch mehr vorgestellt und unternehmensintern (weiter-)entwickelte Methoden sind ebenso existent. Die eigene Erwartungshaltung an eine Messmethode ist folglich sehr entscheidend. Sollen wettbewerbs- oder unternehmensweite Vergleiche angestellt werden, ist die FPA u. a. aufgrund der Verbreitung, Anwendungsmöglichkeit auf unterschiedliche Projekte und vorhandener Vergleichsdaten und Anwendungsrichtlinien empfehlenswert.¹⁰⁴

Zur Vertiefung in das Thema der Messmethoden, eignen sich die zitierten Quellen insb. die Werke von *Ebert und Dumke*, *Sneed* sowie von *Plewan und Poensgen*.

2.2 Kennzahlen für Produktivität und Qualität

Gemäß DeMarco zeichnet sich eine „brauchbare“ Kennzahl dadurch aus, dass sie messbar, unabhängig, verlässlich und genau ist.¹⁰⁵ Plewan und Poensgen haben mit der FPA zusammenhängende Kennzahlen zusammengestellt, die „einerseits mit vertretbarem Aufwand messbar und andererseits relevante Erkenntnisse und Aussagen liefern.“¹⁰⁶ Anhand dessen können effektiv aussagekräftige Benchmarks¹⁰⁷ für die Produktivität und Qualität eines Softwareprojektes erstellt werden.

Die wichtigste Kennzahl Leistungsproduktivität wurde bereits in Kapitel 2.1 als $L_p = \text{Projektergebnis} / \text{Projektaufwand}$ eingeführt. Als Projektergebnis sehen Plewan und Poensgen die Functional Size (fs) des Projektes vor und als Projektaufwand die Personenmonate (pm). Da die Zeit in der Praxis häufig in Stunden gemessen wird, wird

¹⁰³ *Poensgen/Bock*, Function-Point-Analyse, S. 153.

¹⁰⁴ *Poensgen/Bock*, Function-Point-Analyse, S. 156 f; *Plewan/Poensgen*, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 61.

¹⁰⁵ *DeMarco*, Was man nicht messen kann, kann man nicht kontrollieren, S. 101.

¹⁰⁶ *Plewan/Poensgen*, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 63.

¹⁰⁷ Benchmarks: Methode zum internen Vergleich von Abteilungen oder den Vergleich mit Wettbewerbern. vgl. *Scheld/Scheld*, Istkostenrechnung, S. 22.

eine standardisierte Umrechnung (zwischen 130 und 152 Stunden je pm) empfohlen.¹⁰⁸ Aus dem reziproken Wert der L_p kann die Project Delivery Rate $PDR = \text{Aufwand (h)} / \text{Functional Size (fs)}$ berechnet werden. Diese veranschaulicht den voraussichtlichen Aufwand für den Anforderungsumfang einer Software und wird für Aufwandsschätzungen benutzt.¹⁰⁹ Für Software, die zu einem Festpreis entwickelt wird, ist die Kostenproduktivität $KP = fs / \text{Projektkosten}$ geeignet. Reziprok dazu können die Kosten pro Function Point ermittelt werden. Eine klassische Kennzahl in der Softwaremessung ist der Gearing-Factor (GF). Im Deutschen bedeutet das „Übersetzung im Sinne eines Getriebes“. Der GF wird aus der Relation von LoC / Functional Size gebildet. Dieser Faktor sagt aus, wie effizient z. B. durch Nutzung von Bibliotheken programmiert wurde.¹¹⁰

Die Qualität der Software wird mit der Anzahl der aufgetretenen Bugs gemessen. In folgender Tabelle 4 sind die wichtigsten Qualitätskennzahlen aufgeführt.

¹⁰⁸ *Plewan/Poensgen*, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 64.

¹⁰⁹ *Plewan/Poensgen*, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 65.

¹¹⁰ *Plewan/Poensgen*, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 77.

Tabelle 4: Qualitätskennzahlen nach Plewan und Poensgen¹¹¹

Kennzahl	Formel	Aussage
Fehlerrate (FR)	$\frac{\text{Anzahl Fehler}}{\text{Functional Size (kfs)}}$	Anzahl gefundener Fehler je 1000 fps
Projektfehlerrate (pFR)	$\sum \frac{\text{Fehler während}}{\text{Projektdauer}}$	Summe aller Fehler die im Laufe des Projektes gefunden wurden
Restfehlerrate (rFR)	$\sum \frac{\text{Fehler nach}}{\text{Livegang}}$	Summe aller Fehler im ersten Monat nach Produktivstart
Gesamtfehlerrate (gFR)	$\frac{pFR + rFR}{\text{Functional Size (kfs)}}$	Anzahl aller Fehler je 1000 fps
Fehlerbehebungsrate (FBR)	$\frac{pFR}{gFR} * 100$	Anteil der gefundenen Fehler von der Gesamtfehlerrate

Die Ergebnisse dieser Kennzahlen sind als Näherungswerte zu sehen, denn die genaue Zahl der Fehler im Code kann nicht gemessen werden. Es können lediglich die aufgetretenen Fehler gezählt werden.¹¹² Anhand der letzten Kennzahl FBR kann abgeschätzt werden, mit welcher Qualität die Software ausgeliefert wurde. Diese und die anderen vorgestellten Kennzahlen beschreiben also, ob die gesetzten Ziele erreicht wurden, bzw. weisen auf Probleme hin.¹¹³ Für die Ursachenanalyse bspw. für eine abnehmende Leistungsproduktivität oder eine steigende pFR sind zusätzliche weiche Kennzahlen hilfreich. Kennzahlen wie die Teamgröße, Testfallabdeckung, Stabilität der Anforderungen, Berufserfahrung der Projektmitglieder sowie Kontinuität des Projektteams¹¹⁴ können dabei helfen, Zusammenhänge herzustellen.

¹¹¹ Plewan/Poensgen, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 62 – 65.

¹¹² Plewan/Poensgen, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 65.

¹¹³ Plewan/Poensgen, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 68.

¹¹⁴ Plewan/Poensgen, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 70 – 76.

Die Studie „*A Structured Review of Productivity Factors in Software Development*“ des Instituts für Informatik der Technischen Universität München bildet 50 verschiedene Produktivitätsfaktoren als Kennzahlen ab. Es handelt sich sowohl um weiche als auch um harte Kennzahlen. Die weichen Kennzahlen sind als nicht technische Einflussfaktoren, wie die Arbeitsumgebung, definiert, während harte Kennzahlen rein technischer Natur sind.¹¹⁵ Die Tatsache, dass mehr weiche als harte Kennzahlen in der Studie beschrieben werden, unterstreicht nochmal deren Relevanz für die Produktivität.

Die komplette Studie steht zum kostenfreien Download zur Verfügung (vgl. dazu Quellenverzeichnis).

2.3 Ursachen für Produktivitätseinbußen

„[...]Controlling productivity is only possible if the influencing factors are known.“¹¹⁶

Bevor Methoden zur Produktivitätsoptimierung beleuchtet werden, wird der Frage nachgegangen, welche Aktivitäten oder Arbeitsabläufe der Produktivität im Wege stehen oder dieser schaden. Mit der Analyse dieser Fragestellung ergeben sich ebenfalls erste Hinweise auf mögliche Produktivitätsfaktoren.

Nach Plewan und Poensgen zählen u. a. die folgenden Punkte zu den *Bad Practices* in der Softwareentwicklung:¹¹⁷

- Unklarheiten über Ziele eines Projektes
- unklare / instabile Anforderungen
- Nacharbeiten durch schlechte Qualität
- „echte Zeitverschwendungen“
- Brook'sches Gesetz

¹¹⁵ Wagner/Ruhe, *A Structured Review of Productivity Factors in Software Development*, S. 12.

¹¹⁶ Wagner/Ruhe, *A Structured Review of Productivity Factors in Software Development*, S. 18f.

¹¹⁷ Plewan/Poensgen, *Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis*, S. 117ff.

Durch Störungen der Kommunikationskanäle zwischen den Projektbeteiligten, Auftraggebern und dem Projektmanagement können *Unklarheiten über die Ziele* entstehen. Bspw. muss jedes Teammitglied über Änderungen des Leistungsumfanges (neue oder gestrichene Features) informiert sein. Andernfalls werden durch die Arbeit an hinfalligen Features oder das Reporten von Bugs, die eigentlich neue Features sind, Zeit bzw. Aufwände verschwendet.¹¹⁸ Diese Störungen sind auch Hauptveranlasser für *Nacharbeiten durch schlechte Qualität*. In Projekten werden hinreichend oft *unklare Anforderungen* gestellt, die durch den Entwickler falsch interpretiert werden und durch mangelnde Rücksprachen zu Fehlerwirkungen führen können. Dieses Problem ist seit Beginn der Softwareentwicklung zu beobachten. Durch die hohe Relevanz wurde daraus eine eigene Disziplin das *Requirements Engineering* entwickelt.¹¹⁹ Unter *echte Zeitverschwendungen* fallen u. a. unnötige oder zu lange Meetings sowie Reise- und Wartezeiten.¹²⁰ Martin schreibt von zwei Wahrheiten über Meetings, welche oft dasselbe Meeting beschreiben. „Meetings sind notwendig. Meetings sind eine Riesenzeitverschwendung.“¹²¹ Ein Problem sind die anfallenden Kosten, die auch für Teilnehmer anfallen, für die das Meeting keinen Nutzen hat. Ein Schritt Richtung Professionalisierung ist es, eine solche Gegebenheit offen zu kommunizieren und das Meeting zu verlassen.¹²² Gemäß Plewan und Poensgen besteht ein Projekttag für einen Entwickler zu 30 bis 40 % aus Präsentations-Vor- und Nachbereitung sowie aus Meetings (vgl. Kapitel 3.5).¹²³ Durch das „Chinesenprinzip“, also dem Hinzufügen von Mitarbeitern zu einem Projekt in Zeitnot, greift das *Brook'sche Gesetz* „Der Einsatz zusätzlicher Arbeitskräfte bei bereits verzögerten Software-Projekten verzögert sie noch mehr“.¹²⁴ Denn durch die Zuführung

¹¹⁸ Plewan/Poensgen, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 17 f.

¹¹⁹ Plewan/Poensgen, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 123 f.

¹²⁰ Plewan/Poensgen, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 121.

¹²¹ Martin, Clean Coder, S. 160.

¹²² Martin, Clean Coder, S. 160.

¹²³ Plewan/Poensgen, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 122.

¹²⁴ Brooks, Vom Mythos des Mann-Monats, S. 260.

neuer Mitarbeiter werden zeitgleich die Kommunikations- und Einarbeitungsaufwände erhöht.¹²⁵

Neben den besprochenen Produktivitätsfaktoren wird die Leistungsfähigkeit der Softwareentwickler von vielen weiteren Faktoren beeinflusst.¹²⁶ Ludewig und Lichter schlagen, in Anlehnung an Fairley und Boehm, die in Abbildung 8 dargestellten Produktivitätsfaktoren vor.

Faktoren	Merkmale
Arbeitsbedingungen	<ul style="list-style-type: none"> – Technische Ausstattung (Hardware und Software) – Störungen – Vielfalt der Aufgaben (Zersplitterung der Arbeitsleistung)
Schwierigkeiten der Aufgabe	<ul style="list-style-type: none"> – Neuigkeitsgrad der Aufgabe, Verfügbarkeit von Lösungen – Komplexität des Zielsystems – Spezielle Anforderungen wie hohe Zuverlässigkeit, extreme Anforderungen zur Datensicherheit oder zur Rechengenauigkeit – Verfügbarkeit klarer und stabiler Anforderungen
Rahmenbedingungen des Projekts	<ul style="list-style-type: none"> – Personelle Kontinuität – Arbeitsklima – Verteilung der Aufgaben und Kompetenzen – Kommunikationswege und -hindernisse (inkl. räumlich verteilte Entwicklung, Kulturbrüche) – Stabilität und Zuverlässigkeit der Führung – Zeit- und Kostendruck
Eignung der Entwickler für das Projekt	<ul style="list-style-type: none"> – Verständnis für den Kunden, Kulturbarrieren – Erfahrung im Anwendungsgebiet, Problemverständnis – Erfahrung in der verwendeten Technologie (Vorgehensmodell, Werkzeuge, Programmierparadigma, Programmiersprache, Methoden)
Individuelle Faktoren	<ul style="list-style-type: none"> – Fähigkeiten als Entwickler – Fähigkeiten zur Arbeit im Team – Disziplin – Motivation

Abbildung 8: Produktivitätsfaktoren nach Ludewig und Lichter¹²⁷

¹²⁵ Brooks, Vom Mythos des Mann-Monats, S. 260; Plewan/Poensgen, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 126.

¹²⁶ Ludewig/Lichter, Software Engineering, S. 76; Plewan/Poensgen, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 117 – 128.

¹²⁷ Ludewig/Lichter, Software Engineering, S. 77.

2.4 Produktivitätsmethode „Pomodoro“

Neben den dem Umfeld betreffenden sind auch individuelle Faktoren von entscheidender Bedeutung. DeMarco und Lister führten 1984 bis 86 Produktivitätsstudien mit einer Grundgesamtheit von 600 Entwicklern durch. Der Untersuchungsaufbau bestand im Wesentlichen aus dem Lösen einer Programmieraufgabe durch jeweils zwei Softwareentwickler aus und in derselben Unternehmung.¹²⁸ Neben sehr großen Leistungsunterschieden zwischen den einzelnen Entwicklern, ergab die Untersuchung eine „deutliche Korrelation mit der Produktivität“ in Abhängigkeit zu dem Teampartner. Daraus folgt, dass die Entwickler eines Unternehmens stets ähnlich gut oder schlecht abgeschnitten haben.¹²⁹ Aus diesen Ergebnissen wird die Relevanz weicher Produktivitätsfaktoren, wie der Arbeitsumgebung, deutlich. Ungeachtet der starken Streuung der individuellen Leistungen in der Softwareentwicklung (vgl. Kapitel 2), passt sich die Leistungsproduktivität eines Entwicklers, sowohl positiv als auch negativ, dem Arbeitsumfeld an. In der Konsequenz bedeutet die Produktivitätssteigerung des Einzelnen eine Steigerung der Teamproduktivität.¹³⁰

Viele Softwareentwickler verfügen über ihre eigene Methodik um möglichst produktiv zu arbeiten. Dennoch empfiehlt Bonset den eigenen Workflow auf Optimierungsmöglichkeiten hin zu prüfen.¹³¹ In der Folge wird die produktivitätsfördernde Methode Pomodoro (*zu dt. die Tomate*) vorgestellt, da diese für Wissensarbeiter wie Softwareentwickler sehr gut geeignet ist. Wie im vorherigen Kapitel besprochen, können Unklarheiten über Ziele bestehen. Wenn zu Beginn eines Arbeitstages kein Plan feststeht, welche Aufgaben zu erledigen sind, besteht wieder die Gefahr der Ressourcenverschwendung. An dieser Stelle setzen nach Bonset die meisten Produktivitätstechniken an.¹³²

¹²⁸ DeMarco/Lister, *Wien wartet auf Dich! produktive Projekte und Teams* ; [mit sechs neuen Kapiteln], S. 43.

¹²⁹ DeMarco/Lister, *Wien wartet auf Dich! produktive Projekte und Teams* ; [mit sechs neuen Kapiteln], S. 44, 47.

¹³⁰ Ludewig/Lichter, *Software Engineering*, S. 78.

¹³¹ Bonset, *T3n - Digit. Pioneers 2015*, S. 29.

¹³² Bonset, *T3n - Digit. Pioneers 2015*, S. 26.

Die Pomodoro-Technik verfolgt den Ansatz, den Arbeitstag in 25-minütige Phasen¹³³ (auch Tomatenzeit genannt) der Konzentration einzuteilen. Es soll dabei helfen, sich während einer Phase auf genau eine Aufgabe konzentrieren zu können und Ablenkungen / Störungen höflich aber bestimmt auf die nächste Phase zu verschieben. Nach dem der Timer abgelaufen ist, ist die derzeitige Aufgabe sofort zu unterbrechen und die aufgeschobenen Aufgaben sind zu bearbeiten. Danach ist eine Pause von 3 bis 5 Minuten angedacht, um in der Folge die nächste „Tomatenzeit“ zu beginnen.¹³⁴

Nach einer Phase sollte das Geschaffte reflektiert und notiert werden.¹³⁵ Dadurch trainiert der Anwender täglich seine Fähigkeit, Aufwände zu schätzen. Für diese Technik bedarf es einer täglichen Planung sowie einer Zeiterfassung der erledigten Aufgaben.¹³⁶ I. d. R. werden Aufwände in der Softwareentwicklung in Stunden aufgezeichnet, sodass dieser Nachteil ausgeklammert werden kann und die Vorteile, wie ein strukturiertes effizientes Arbeiten sowie „auswertbare Daten zum eigenen Workflow“¹³⁷, überwiegen.

Für die tägliche Planung empfiehlt es sich, Slots für die E-Mail-Bearbeitung, Rückfragen an oder durch andere Entwickler und Pausen festzulegen. Durch die Eingrenzung von E-Mail oder Instant Messenger-Benachrichtigungen auf einen oder mehrere Zeitslots, werden Gewohnheiten wie Multitasking oder Prokrastination unterbunden bzw. die Ablenkungen dadurch minimiert.¹³⁸

Weitere Produktivitätstechniken sind Getting Things Done, Personal Kanban und die Action Method.¹³⁹

¹³³ Für die Zeiterfassung der 25 Minuten - Phasen existiert eine kostenlose Webapp. Zu finden unter <http://tomato-timer.com/>

¹³⁴ *Martin*, Clean Coder, S. 167.

¹³⁵ *Cirillo*, HOME - The Pomodoro Technique®.

¹³⁶ *Martin*, Clean Coder, S. 167; *Bonset*, T3n - Digit. Pioneers 2015, S. 26f.

¹³⁷ *Bonset*, T3n - Digit. Pioneers 2015, S. 27.

¹³⁸ *Cirillo*, HOME - The Pomodoro Technique®.

¹³⁹ *Bonset*, T3n - Digit. Pioneers 2015, S. 26 – 28.

2.5 Vorstellung produktivitätsoptimierender Maßnahmen

2.5.1 Zielvereinbarungssysteme

Bei Zielvereinbarungen handelt es sich um Absprachen zwischen Vorgesetzten und seinen Mitarbeitern oder Teams über zu erreichende Ziele.¹⁴⁰ Unternehmungen existieren um Ziele zu erreichen und die Aufgabe von Führungskräften ist es, die Erfüllung dieser Ziele sicher zu stellen. Festgelegte Ziele müssen bei den Mitarbeitern Akzeptanz finden und kooperativ verfolgt werden.¹⁴¹

In der dotSource GmbH finden zweimal im Jahr Mitarbeitergespräche statt. Während dieser Gespräche werden u. a. die Leistungen der Mitarbeiter besprochen. Basierend auf der Leistungseinschätzung können Ziele vereinbart werden. Ziele sind bspw. die Verminderung einer entwicklerspezifischen Bugrate je Feature, Verbesserung der Code-Qualität oder das Ablegen einer Zertifizierung.

Mittels Zielvereinbarungssystemen ist es möglich, die diskutierten Probleme hinsichtlich der heterogenen Ausbildung in der Softwareentwicklung entgegen zu wirken. Der Mitarbeiter kann durch individuelle (Lern-)Ziele dahingehend gefördert werden, als dass dieser den Anforderungen professioneller Softwareentwicklung und den Qualitätsanforderungen der dotSource GmbH gerecht wird.

Anreize für die Erfüllung von Zielen entstehen u. a. durch Aufgabenidentifikation, Bonuszahlungen oder dem Selbstverpflichtungseffekt.¹⁴²

Durch die ehrliche und gleichberechtigte Beteiligung eines Mitarbeiters an der Bestimmung eines Ziels, entsteht eine hohe Bindung an die Zielentscheidung. Es entsteht der sog. Selbstverpflichtungseffekt, durch welchen sich der Mitarbeiter stärker zur Zielerfüllung verpflichtet fühlt und eine höhere Leistungsbereitschaft zeigt.¹⁴³ Während der Zielvereinbarung werden Endtermine und Ressourceneinsatz (bspw. Arbeitszeit) festgelegt und dokumentiert. Weiterhin ist der Mitarbeiter selbst für die

¹⁴⁰ *Bhagwati*, Zielvereinbarungssysteme - das Wirtschaftslexikon .com.

¹⁴¹ *Watzka*, Zielvereinbarungen in Unternehmen, S. 15.

¹⁴² *Watzka*, Zielvereinbarungen in Unternehmen, S. 43 ff.

¹⁴³ *Watzka*, Zielvereinbarungen in Unternehmen, S. 47.

Wahl geeigneter Mittel und oder Maßnahmen zur Zielerfüllung verantwortlich. Somit ist eine Sensibilisierung für effizientes Handeln mit eingeschlossen.¹⁴⁴

Ziele, wie „Akquirierung 20 neuer Kunden“ oder „Produktivitätserhöhung in der Softwareentwicklung“, sind Mittel um ein übergeordnetes Ziel, wie eine Umsatzsteigerung, zu erreichen. D. h., dass diese (Unter-)Ziele durch Aufspaltung (Zielkaskadierung) eines strategischen Ziels entstanden sind und in einem Zweck-Mittel-Verhältnis stehen.¹⁴⁵ Folglich können Zielvereinbarungssysteme unternehmensweit und nicht nur bei den Softwareentwicklern eingesetzt werden.

In Abbildung 9 ist die Aufspaltung eines Oberziels in Teilziele in allgemeiner Form dargestellt und in der Abbildung 10 an einem Beispiel umgesetzt.

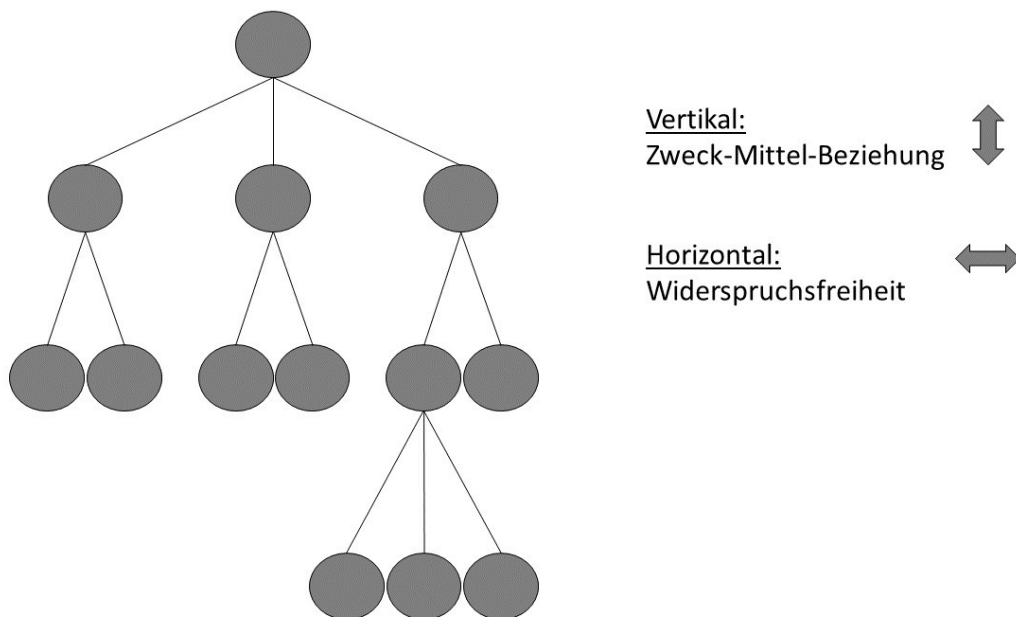


Abbildung 9: Zielkaskadierung¹⁴⁶

¹⁴⁴ Watzka, Zielvereinbarungen in Unternehmen, S. 68, 31.

¹⁴⁵ Watzka, Zielvereinbarungen in Unternehmen, S. 29.

¹⁴⁶ Watzka, Zielvereinbarungen in Unternehmen, S. 30.

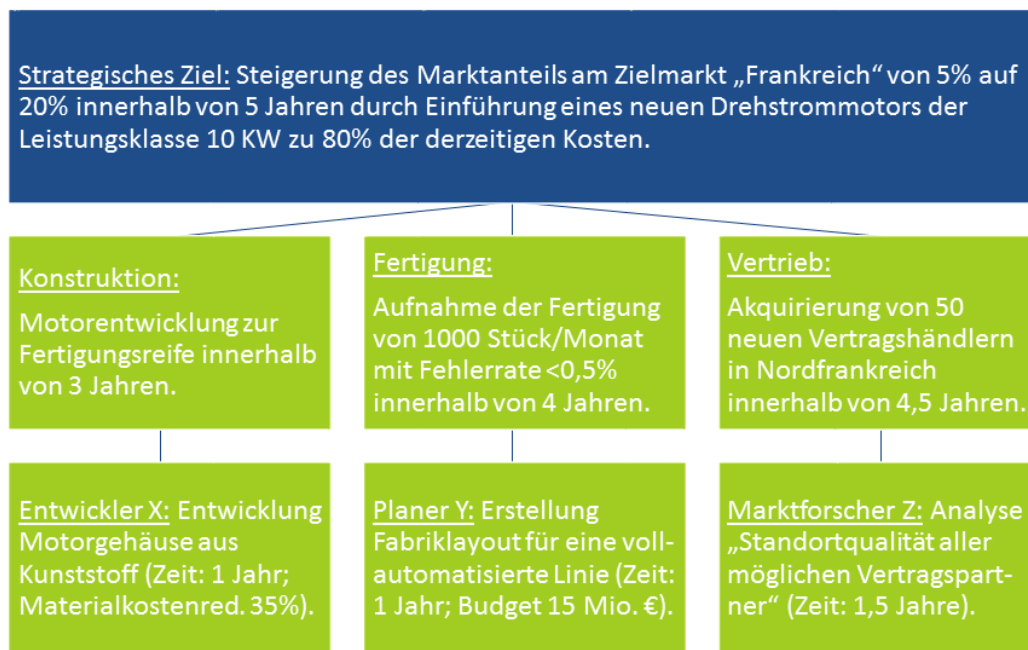


Abbildung 10: Beispiel Zielkaskadierung¹⁴⁷

2.5.2 Mitarbeiter-Aktienindex

Der Mitarbeiteraktienindex (MAX) ist ein System aus der DL-Branche (insb. der Hotellerie) zur Teamentwicklung und nachhaltigen Mitarbeitermotivation. Die Bezeichnung der Methode stellt eine Analogie zu dem DAX da und soll an den Finanzmarkt erinnern. Jeder neue Mitarbeiter startet, ähnlich einer Neuemission, mit einem Nennwert von 1000 Pixel¹⁴⁸ - dem Player-Index (PIX).¹⁴⁹

Der PIX wird aus der Selbstbewertung der einzelnen Mitarbeiter und der Einschätzungen des Vorgesetzten u. a. zu fachlichen und persönlichen Kompetenzen ermittelt.¹⁵⁰ Weiterhin haben Faktoren, wie das Einreichen von Verbesserungsvorschlägen, Pünktlichkeit, Weiterbildungen und gesundheitliche Aspekte (Body-Mass-Index, Raucher)¹⁵¹, einen direkten Einfluss auf den PIX-Wert. Neben dem PIX

¹⁴⁷ Watzka, Zielvereinbarungen in Unternehmen, S. 30.

¹⁴⁸ Es wurde auf eine Währungseinheit verzichtet, um einen „spielerischen Charakter“ zu vermitteln. Kobjoll u. a., MAX, S. 19.

¹⁴⁹ Kobjoll u. a., MAX, S. 18 f.

¹⁵⁰ Kobjoll u. a., MAX, S. 17 f.

¹⁵¹ Kobjoll u. a., MAX, S. 19 f, 64 f.

werden noch der Team- und der Community-Index (CIX) berechnet. Der Team-Index setzt sich aus Faktoren wie z. B. dem Erreichen von Umsatzzielen und dem Mittelwert der Player-Indizes zusammen.¹⁵² Diese Ratings werden monatlich erstellt und intern veröffentlicht. Ein Ziel ist es, das Ursachen für ein weniger gutes Rating, team-intern ergründet und behoben werden. Aus den Teambewertungen wird der CIX, der Index des Unternehmens, erhoben.¹⁵³

Durch den Wettbewerb unter den Teams kann es zu Leistungssteigerungen, durch das gemeinsame Ziel besser als die anderen Teams sein zu wollen, kommen.¹⁵⁴ Wie in Kapitel 2.4 beschrieben, konnten DeMarco und Lister feststellen, dass das Leistungsvermögen eines Teamkollegen einen deutlichen Einfluss auf das der anderen Teammitglieder hat. Durch das Wissen über die eigenen Stärken und Schwächen, mittels der Player-Indizes, und dem Teamranking, könnte das Streben nach Weiterbildung und Effizienzsteigerung des Einzelnen verstärkt werden. Jedoch sollten die Bewertungskriterien sorgsam gewählt werden. Faktoren wie Rauchen oder der BMI greifen tief in die Privatsphäre des Mitarbeiters ein. In der Folge könnten Maßnahmen zur Erzeugung von Zufriedenheit mit oder Vertrauen in den Arbeitgeber kannibalisiert werden.

2.5.3 Partizipatives Produktivitätsmanagement

Die PPM Methode ist mit dem Mitarbeiter-Aktienindex vergleichbar. Durch den erfolgreich Einsatz in diversen Dienstleistungssektoren abseits der Hotelbranche (bspw. ÖPNV-Unternehmen oder Marktforschungsinstituten)¹⁵⁵, wird die Methode kurz vorgestellt.

Der Einsatz des Partizipativen Produktivitätsmanagements (PPM) beinhaltet die Umsetzung von Leistungsmesssystemen. PPM muss dem Anspruch Genüge tun, dass

¹⁵² *Kobjoll u. a.*, MAX, S. 76 ff.

¹⁵³ *Kobjoll u. a.*, MAX, S. 21.

¹⁵⁴ *Katzenbach/Smith*, Teams, S. 15.

¹⁵⁵ *Schmidt u. a.*, in: Messung und Bewertung von Dienstleistungen als Grundlage für Produktivitätssteigerungen, S. 215 ff; *Roth*, Partizipatives Produktivitätsmanagement (PPM) bei Spitzentechnologie nutzenden und wissensintensiven Dienstleistungen.

sämtliche Messkriterien durch den Leistungserbringer beeinflussbar sind.¹⁵⁶ Die Installation des PPM-Ansatzes geschieht in den folgenden Schritten:¹⁵⁷

1. Aufgabenbereiche der Untersuchungsperson / -gruppe identifizieren
2. dafür geeignete Indikatoren bestimmen
3. Bewertungsfunktionen erstellen
4. Feedbacksystem aufsetzen

Diese Schritte werden mit den jeweiligen Mitarbeitern zusammen erarbeitet.¹⁵⁸ Daraus resultiert, neben validen Kennzahlen, Verständnis der Belegschaft sowie Vertrauen in den Arbeitgeber. Bei Anwendung der PPM erfolgt zunächst eine Nullmessung ohne, dass die Ergebnisse dem Leistungserbringer mitgeteilt werden, gefolgt von der ersten Erhebung mit anschließendem Feedback-Report.¹⁵⁹

2.5.4 Business-Dashboards

Wie in Kapitel 2.2 besprochen, sind Kennzahlen ideale Entscheidungshilfen und Indikatoren für aktuelle Leistungsanalysen. Die Erhebung von Kennzahlen liefert allein-stehende Messdaten. Um effizient Schlüsse aus den Erhebungen ziehen zu können, bedarf es einer Veredelung der Daten in Form von Diagrammen und Relationen zwischen den Kennzahlen.

¹⁵⁶ Schmidt u. a., in: Messung und Bewertung von Dienstleistungen als Grundlage für Produktivitätssteigerungen, S. 209.

¹⁵⁷ Schmidt u. a., in: Messung und Bewertung von Dienstleistungen als Grundlage für Produktivitätssteigerungen, S. 210 – 213.

¹⁵⁸ Schmidt u. a., in: Messung und Bewertung von Dienstleistungen als Grundlage für Produktivitätssteigerungen, S. 210.

¹⁵⁹ Schmidt u. a., in: Messung und Bewertung von Dienstleistungen als Grundlage für Produktivitätssteigerungen, S. 214.

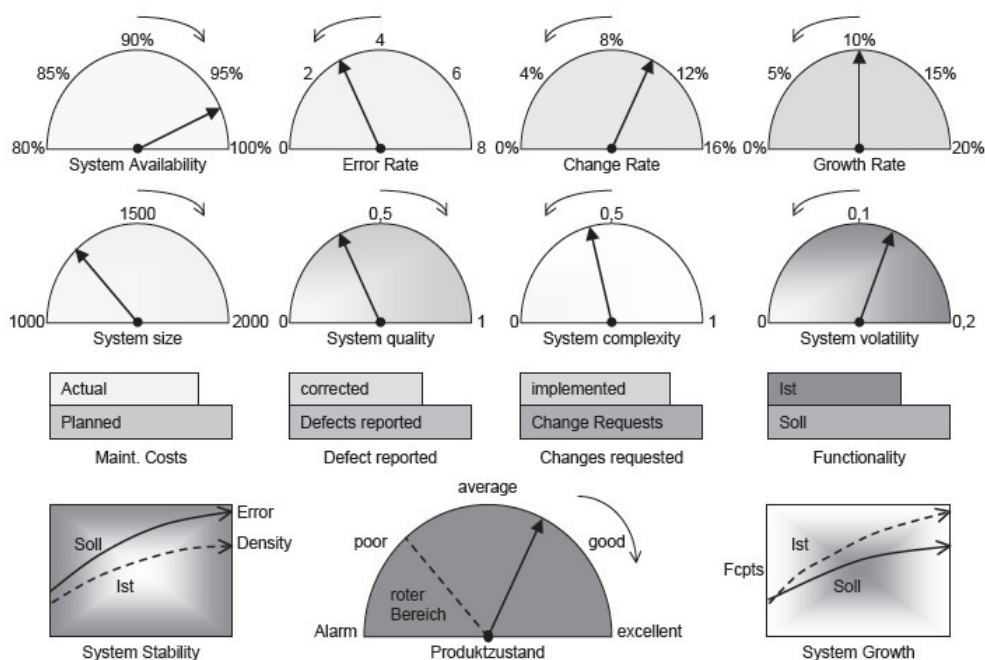


Abbildung 11: Dashboard mit Softwaremetriken¹⁶⁰

Sneed empfiehlt dafür die Darstellung als Tachometer. In der Abbildung 11 sind beispielhaft Tachometer und Balkendiagramme dargestellt, welche die Fortschritte des laufenden Projektes widerspiegeln. Die Diagramme können auch in farbige Bereiche eingeteilt werden, um noch effizienter für das operative Projektmanagement zu sein.¹⁶¹

Der größte Unterschied zwischen dynamischen Dashboards und statischen Reports, ist die Interaktionsmöglichkeit. Reports bieten nur den Informationsgehalt, den der Ersteller vorgesehen hat, während die Kennzahlen auf Dashboards auch ausgetauscht, weggelassen oder gefiltert werden können. Weiterhin sind Benachrichtigungen konfigurierbar, welche das Management bei Erreichen kritischer Werte informiert.¹⁶²

Mit der Einführung solcher Dashboards können, neben projektrelevanten Kennzahlen, ebenso die Team-Indizes und der Community-Index für alle Mitarbeiter sichtbar aufbereitet werden.

¹⁶⁰ Sneed u. a., Software in Zahlen, S. 324.

¹⁶¹ Sneed u. a., Software in Zahlen, S. 323 f.

¹⁶² Zunke, T3n - Digit. Pioneers 2015, S. 59.

2.6 Fazit These 1

In Kapitel 2 wurde die Produktivität aus industrieller sowie aus Sicht eines Dienstleistungsunternehmens vorgestellt. Weiterhin wurde der Produktivitätsgedanke auf die Softwareentwicklung übertragen und der Stand der Forschung evaluiert.

Es wurden die Besonderheiten der Produktivität in der DL-Branche und die Probleme in der Softwareentwicklung diskutiert. Ferner sind die wichtigsten Messmethoden und produktivitätssteigernde Maßnahmen präsentiert und kritisch gewürdigt worden.

Festzuhalten ist, dass die Produktivität aussagekräftig und mit vertretbarem Aufwand gemessen werden kann. Jedoch erst nachdem eine geeignete Methode oder ein geeigneter Methoden-Mix ermittelt und eingeführt wurde und dieser die Akzeptanz der Mitarbeiter erfährt.

Die Einführung einer entsprechenden Methode bedarf eines hohen Schulungsaufwands, die professionelle Anwendung kann bei Bedarf aber auch an einen Dienstleister abgegeben werden.¹⁶³

¹⁶³ *Plewan/Poensgen*, Produktive Softwareentwicklung - Bewertung und Verbesserung von Produktivität und Qualität in der Praxis, S. 79.