KOOPERATIVE STEUERUNG VON MODELLVERSUCHSFAHRZEUGEN

ENTWICKLUNG EINES INTELLIGENTEN FAHRENTSCHEIDERS

Softwareentwicklungspraktikum
Sommersemester 2008

Pflichtenheft zum System

E C A R

Auftraggeber
Technische Universität Braunschweig
Institut für Betriebssysteme und Rechnerverbund
Prof. Dr.-Ing. Lars Wolf
Mühlenpfordstrasse 23
38106 Braunschweig

Betreuer: Kai Homeier, Carina Flämig

Auftragnehmer

Name	E - Mail
Wira Kakar	wirak@web.de
Chen Zhiwei	czwnii@hotmail.com
Rayan Merched El Masri	rayan_masri@hotmail.com
Ekrem Enes Duman	dumanenes@hotmail.de
Günter Dusch	g.dusch@Googlemail.com

Braunschweig, 18.04.2008

Versionsübersicht

Version	Datum	Autor	Status	Kommentar
1	17.04.2008	Auftragnehmer	akzeptiert	Erste Version abgeschlossen
2	23.04.2008	Auftragnehmer	akzeptiert	Layout, Rechtschreibung, Use-Case-Diagramm, Geschäftsprozesse, Abgrenzungskriterien

I n h a l t s v e r z e i c h n i s

1 Zielbestimmung

In Anbetracht der aktuellen Bedeutung und weltweiten Forschung und Entwicklung in der autonomen Robotertechnologie, die sich noch in den Anfängen befindet, ist es von großer Wichtigkeit für den Innovationsstandort Deutschland auch in diesem mit hohem Potential versehenem Forschungsbereich mitzuhalten und Akzente zu setzen.

Hinzu kommt, dass es seit Menschengedenken ein Wunschtraum ist autonom arbeitende Maschinen zu erschaffen, die Arbeiten und Leistungen erbringen, die zu gefährlich, unzugänglich, mühsam oder undurchführbar für den Menschen sind. Hier wollen wir einen Beitrag leisten.

Ziel dieses Projektes ist die Erschaffung eines Softwaresystems, das einem sich fortbewegendem Roboter mit diversen Sensoren ermöglicht ein gesuchtes Objekt - in diesem Fall einen Ball - in einem vorher unbekannten Labyrinth zu finden. Die zusammengebaute fertige Hardware wird vom Institut bereitgestellt, sodass es unsere Aufgabe ist diese leere Hülle mit „Leben“ zu füllen, damit die Problematik des Suchens und Findens möglichst effizient gelöst werden kann.

1.1 Musskriterien

Zu den Musskriterien gehören die Identifikation von Baken zur Navigation, eine kamerabasierte Identifikation des gesuchten Objekts, eine adäquate Fortbewegung innerhalb des Labyrinths, eine Speicherung der bereits abgesuchten Wege und Kreuzungen, das Auffinden des gesuchten Objekts und die Positionsberechnung anhand der aufgestellten Baken.

1.2 Wunschkriterien

Zu den Wunschkriterien zählt die Erstellung einer Karte des Labyrinths, das Auffinden des gesuchten Objektes in minimaler Zeit und eine Effizienzsteigerung durch den Einsatz mehrerer autonomer Fahrzeuge, die miteinander kommunizieren und so Synergien bilden.

2 Produkteinsatz

Der Einsatzbereich dieses Systems ist das Modellversuchsfahrzeug des IBR, wobei der Aufgabenbereich dieses Fahrzeugs ein willkürliches Labyrinth ist. Dieses Projekt stellt unter Anderem ein Grundlagensystem in der Forschung der Roboterprogrammierung dar, das für zukünftige Systeme für unterschiedliche Anforderungen die Basis bildet.

2.1 Anwendungsbereiche

Das Suchen und Finden von Elementen/Objekten auf fremden Planeten ist ein möglicher Anwendungsbereich, wobei ein autonomes Fahrzeug eine Notwendigkeit darstellt, da eine direkte Steuerung aufgrund der enormen Distanz ungeeignet ist (Aufgabenbereich in der Raumfahrt). Eine weitere Möglichkeit sind Erkundungsmissionen in für den Menschen unzugänglichen Gebieten z.B. die Erforschung von einsturzgefährdeten Höhlen oder extrem kleinen unerforschten Hohlräumen in antiken Bauten wie Pyramide (Aufgabenbereich in Forschung). Hinzu kommt das Suchen und Finden von Verstopfungen und starken Verschmutzungen in Kanalisationen (wirtschaftlicher Aufgabenbereich). Eine denkbare Option ist auch das Suchen- und Finden von Landminen in für den Menschen zu gefährlichen Gebieten (militärischer Aufgabenbereich). Theoretisch mögliche Anwendungsgebiete erschließen sich zudem im Suchen und Finden von verletzten Soldaten auf dem Schlachtfeld (militärischer Aufgabenbereich).

2.2 Zielgruppen

Das Softwaresystem ist primär für Studenten an der technischen Universität Braunschweig als Lernmöglichkeit gedacht, in dem praktisch das erworbene Wissen der ersten vier Semester angewendet werden kann. Dieses System ist sekundär für Programmierer und Entwickler als Ausgangssystem zu verstehen, um nach den jeweiligen Bedürfnissen und Anforderungen einen erfolgreichen Suchen und Finden Vorgang zu ermöglichen und so ausbauen zu können, dass es in anderen Anwendungsbereichen Lösungen liefert.

2.3 Betriebsbedingungen

Die physikalische Umgebung des Systems ist ein Labyrinth in einem beliebigen Raum. Die Betriebszeit beläuft sich auf mehrere Minuten bis Stunden je nach dem wie komplex und groß das zu bearbeitende Gebiet des Systems ist und die Stromversorgung gewährleistet wird. Abhängig vom zukünftigen Einsatzgebiet soll auch ein Dauerbetrieb, jedoch nur als Erweiterungsmöglichkeit für externe Programmierer und Entwickler, ermöglicht werden. Eine

dauerhafte Beaufsichtigung des Systems ist nicht zwingend erforderlich, aber durch die Anforderungen von speziellen Einsatzmöglichkeiten möglich.

3 Produktleistungen

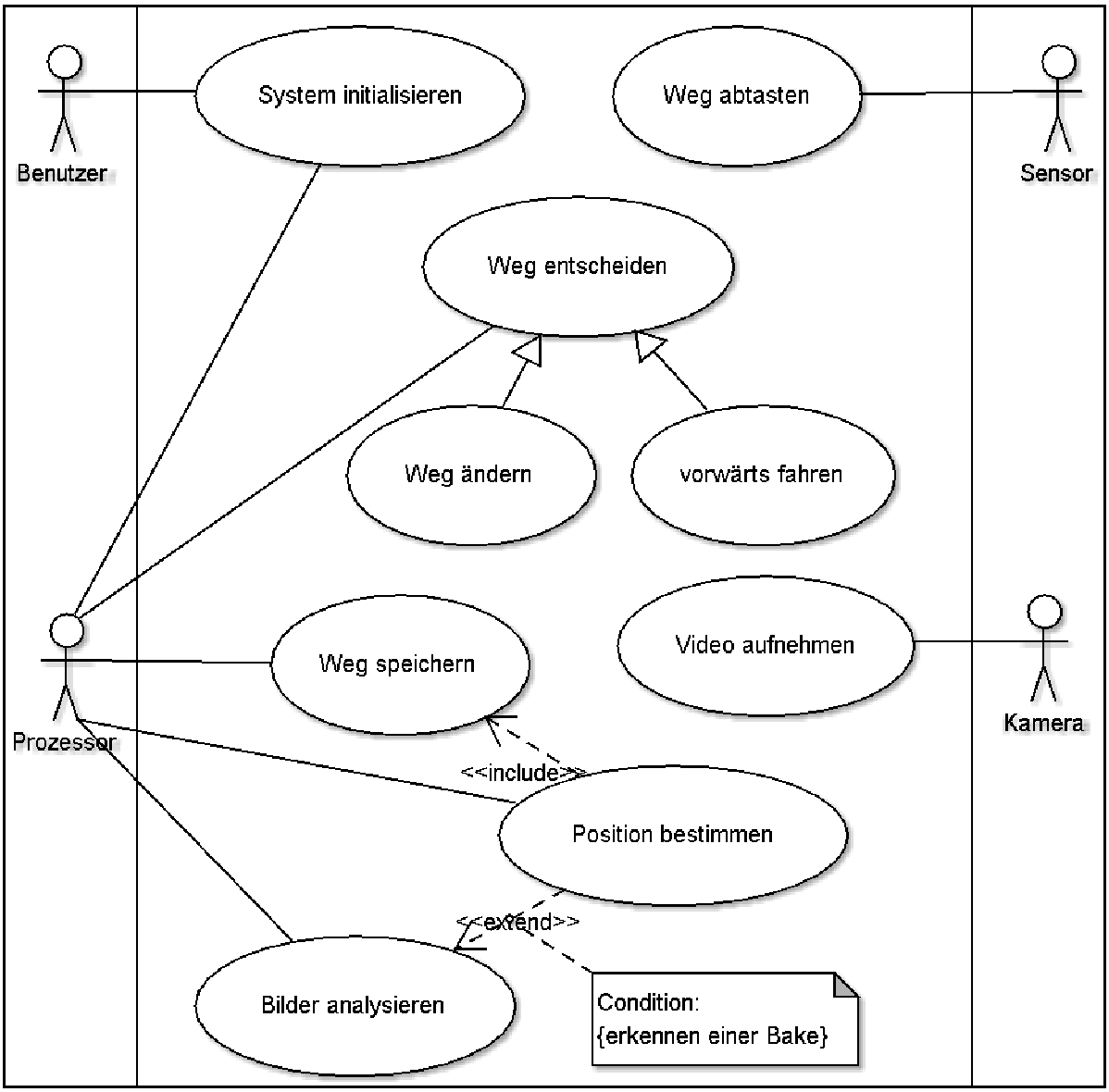

Abbildung 1: Use-Case-Diagramm des autonomen Modellfahrzeugs

4 Produktfunktionen

/ **F10**/

Geschäftsprozess: System initialisieren

Ziel: Das System starten

Vorbedingung: Die Stromversorgung wird gewährleistet und das Programm steht auf der Hardware gespeichert zur Verfügung

Nachbedingung Erfolg: Das System startet erfolgreich

Nachbedingung Fehlschlag*:* Das System kann nicht starten

Akteure: Benutzer, Prozessor

Auslösendes Ereignis: Auf Wunsch des Benutzers wird das System gestartet

Beschreibung:

1. Der Schalter wird eingeschaltet
2. Der Compiler kompiliert das das Programm

/**F20**/

Geschäftsprozess: Weg abtasten

Ziel: Das Abtasten des Weges findet durch die Sensoren statt

Vorbedingung: Die Sensoren müssen eingeschaltet und das Gesamtsystem muss schon initialisiert worden sein

Nachbedingung Erfolg: Die Sensoren können klar und deutlich die Hindernisse erkennen

Nachbedingung Fehlschlag*:* Die Sensoren können keine Hindernisse erkennen

Akteure: Sensor

Auslösendes Ereignis: Der Start und die Initialisierung führen zur Einschaltung der Sensoren und ermöglichen die Reaktion auf Hindernisse

Beschreibung:

1. Durch den Schalter wird das Modellversuchsfahrzeug aktiviert
2. Die Sensoren tasten ab, ob es Hindernisse gibt

/**F30**/

Geschäftsprozess: Weg entscheiden

Ziel: Der richtige, noch nicht besuchte Weg wird gewählt und die Kollision mit den Hindernissen verhindert

Vorbedingung: Das Gesamtsystem muss gestartet worden sein und die Sensoren müssen vorhanden sein

Nachbedingung Erfolg: Der noch nicht besuchte Weg wird gewählt. Kein Kollidieren mit den Hindernissen

Nachbedingung Fehlschlag*:* Der Weg, der schon besucht wurde, wird wieder abgefahren oder es findet eine Kollision mit einem Hindernis statt
Akteure: Prozessor
Auslösendes Ereignis: Das erfolgreiche Starten des Modellversuchsfahrzeugs
Beschreibung:
1. Der Weg wird Mit Hilfe von Sensoren abgetastet.
2. Die Entscheidung für eine Richtung wird getroffen

/**F31**/
Geschäftsprozess: Weg ändern
Ziel: Nachdem ein Hindernis gefunden wurde, wird die Fahrtrichtung geändert und ein neuer Weg gefunden
Vorbedingung: Es muss dem Prozessor mitgeteilt werden, dass es ein Hindernis existiert
Nachbedingung Erfolg: Die Fahrtrichtung wird geändert und das Modellversuchsfahrzeug bewegt sich in eine andere Richtung
Nachbedingung Fehlschlag*:* Die Fahrtrichtung wird nicht geändert und das Modellversuchsfahrzeug fährt gegen das Hindernis
Akteure: Prozessor
Auslösendes Ereignis: Wenn die Sensoren ein Hindernis finden
Beschreibung:
1. Hindernis wird gefunden
2. Das Modellversuchsfahrzeug ändert seine Richtung in einen noch nicht besuchten Weg

/**F32**/
Geschäftsprozess: Vorwärts fahren
Ziel: Problemloser Fortbewegungsprozess in gerader Richtung
Vorbedingung: Auf dem Weg ist kein Hindernis vorhanden
Nachbedingung Erfolg: Das Modellversuchs fährt geradeaus
Nachbedingung Fehlschlag*:* Aus technischen Gründen ändert das Modellversuchsfahrzeug seine Fahrtrichtung oder fährt nicht
Akteure: Prozessor
Auslösendes Ereignis: Wenn die Sensoren kein Hindernis auf ihrem Weg finden, fährt das Modellversuchsfahrzeug vorwärts
Beschreibung:

1. Der Prozessor bekommt keine Meldung über ein Hindernis
2. Modellversuchsfahrzeug fährt vorwärts

/**F40**/

Geschäftsprozess: Position bestimmen

Ziel: Wenn die Erkennung der Baken durch das Modellversuchsfahrzeug durchgeführt wurde, wird die Position bestimmt

Vorbedingung: Die Bildanalyse muss erfolgreich beendet worden sein, damit die Baken erkannt werden

Nachbedingung Erfolg: Das Modellversuchsfahrzeug kann nach dem Finden der Baken seine Position bestimmen

Nachbedingung Fehlschlag*:* Aus technischen Gründen kann das Modellversuchsfahrzeug seine Position entweder nicht bestimmen oder falsch bestimmen

Akteure: Prozessor

Auslösendes Ereignis: Das Finden der Baken

Beschreibung:

1. Die Kamera sendet die Bilder an den Prozessor
2. Der Prozessor analysiert die Bilder und erkennt die Baken darin
3. Das Modellversuchsfahrzeug bestimmt seine Position

/**F50**/

Geschäftsprozess: Bilder analysieren

Ziel: Die Bilder der Objekte müssen analysiert und mit den Bildern der Baken und des gesuchten Objekts verglichen werden

Vorbedingung: Dem Prozessor muss die Form und Farbe der Baken bzw. des gesuchten Objekts bereits bekannt sein

Nachbedingung Erfolg: Die aufgenommenen Bilder werden analysiert und darin werden die Baken bzw. das gesuchte Objekt gefunden

Nachbedingung Fehlschlag*:* Aus technischen Gründen können die Bilder nicht analysiert werden oder die aufgenommenen Bilder stimmen mit den Bildern, die dem Prozessor bereits bekannt sind, nicht überein, weil es möglicherweise in dem Labyrinth keine Baken gibt bzw. das gesuchte Objekt nicht existiert

Akteure: Prozessor

Auslösendes Ereignis: Starten des Gesamtsystems

Beschreibung:

1. Bilder werden aufgenommen.
2. Farbe und Form der gefundenen Objekte werden mit Hilfe von OpenCV interpretiert.

3. Ergebnis der Interpretation wird mit den Daten des gesuchten Objekts oder der Baken verglichen.

/**F60**/

Geschäftsprozess: Weg speichern

Ziel: Der Weg wird gespeichert um zu vermeiden, bereits besuchte Wege wieder zu besuchen

Vorbedingung: Ein einwandfreier erfolgreicher Suchvorgang, sowie die Funktionsfähigkeit der Kamera, muss gewährleitet sein

Nachbedingung Erfolg: Bereits besuchter Weg wird gespeichert

Nachbedingung Fehlschlag*:* Ein bereits besuchter Weg wird nicht gespeichert oder ein noch nicht besuchter Weg wird gespeichert

Akteure: Prozessor

Auslösendes Ereignis: Das Modellversuchsfahrzeug startet den Fortbewegungsprozess

Beschreibung:

1. Initialisierung des Modellversuchsfahrzeugs
2. Modellversuchsfahrzeug fährt und legt eine Strecke zurück
3: Diese zurückgelegte Strecke des Modellversuchsfahrzeugs wird gespeichert

/**F70**/

Geschäftsprozess: Video aufnehmen

Ziel: Es werden Aufnahmen mit der Videokamera gemacht, um die Baken und mögliche andere Objekte zu finden

Vorbedingung: Die Kamera muss eingeschaltet sein

Nachbedingung Erfolg: Video wird aufgenommen und die Daten werden dem Prozessor geliefert

Nachbedingung Fehlschlag*:* Aus technischen Gründen können keine Videos aufgenommen werden

Akteure: Kamera

Auslösendes Ereignis: Das Einschalten der Videokamera

Beschreibung:

1. Die Kamera wird eingeschaltet
2. Die Kamera fängt mit der Aufnahme an
3. Die aufgenommen Bilder dem Prozessor weitergeleitet

5 Produktdaten

/**D10**/ Daten der Ortung

Position des Modellversuchsfahrzeuges

Karte

Baken

/**D20**/ Daten der Bildverarbeitung

Kamerabilder

6 Produktleistungen

/**L10**/ Zwischen einem Event der Software und der Sensoren muss eine Reaktionszeit von max. 100 Millisekunden gewährleistet werden.

/**L20**/ Die Suche erfolgt effizient.

/**L30**/ Das Modellversuchsfahrzeug soll sich von einer beliebigen Startposition aus Orientieren können.

/**L40**/ Das Modellversuchsfahrzeug muss das gesuchte Objekt finden.

/**L50**/ Das Modellversuchsfahrzeug muss die Hindernisse erkennen und darf mit denen nicht kollidieren.

7 Qualitätsanforderungen

	sehr wichtig	Wichtig	normal	nicht relevant
Richtigkeit	X			
Zuverlässigkeit		X		
Robustheit		X		
Leistung			X	
Benutzerfreundlichkeit		X		
Überprüfbarkeit	X			
Wartbarkeit	X			
Reparierbarkeit	X			
Entwicklungsfähigkeit	X			
Wiederverwendbarkeit	X			
Portabilität	X			
Verständlichkeit		X		
Interoperabilität	X			
Produktivität		X		
Pünktlichkeit	X			
Sichtbarkeit	X			

8 Benutzeroberfläche

Es wird eine Benutzeroberfläche und eine Hardware-Schnittstelle definiert. Da es sich hier um eine Systemsoftware und nicht um eine Anwendungssoftware handelt, werden wir keine Benutzeroberflächen, im Sinne einer grafischen Benutzeroberfläche, erstellen.

/**B10**/ USB-Kernel-Schnittstelle die das Modellfahrzeug mit dem Entwicklungsrechner verbindet um die Übertragung von Java-Quellcode zu ermöglichen.

9 Nichtfunktionale Anforderungen

/**NF10**/ Das Produkt ist mit geringem Aufwand weiterentwickelbar und wartbar.

/**NF20**/ Das Produkt ist benutzerfreundlich.

/**NF30**/ Das Produkt ist plattformunabhängig.

10 Technische Produktumgebung

10.1 Software

Für den Benutzer ist nur ein installierter C++ Compiler für einen einwandfreien Betrieb des Roboters vorausgesetzt.

10.2 Hardware

Der Auftrag beschränkt sich auf die Produktion eines geeigneten Softwaresystems. Daher wird der Roboter mit sämtlicher Hardware vorausgesetzt. Diese besteht aus folgenden Komponenten:

- Mini-ITX Express Motherboard:

CPU	Support VIA Eden 1GHz Processor, Package type 400
Memory	1 * 240 -pin DIMM Sockets for unbuffered DDR2 533 SDRAM up to 1 GB
Chipset	VIA CN700 + 8237RP Chipset
PCI enhanced IDE	4 * Ultra DMA 133 / 100 / 66 IDE Devices Support .
Serial ATA Interface	2 Serial ATA HDDs with RAID 0.1 Function Support
VGA Interface	VIA Unichrome Pro Processor
Video Memory	Share Memory up to 128MB
Audio Interface	VIA VT1617A 6-Channel AC 97' Audio CODEC
LAN Interface	VIA VT6103CL 10 / 100 PCI LAN PHY
Extended Interface	1 * 32-bit PCI Slot
	1 * PCI 120-Pin Adapter Connector With Expansion Daughter Boards
Internal I/O Port	3 * High Speed USB Connectors @ 480 MBit / s for 6 USB 2.0 Ports
	CPU / Chassis Fan Connectors / 1 * 26-pin Parallel Connector
	9-Pin COM2 Connector / 1 * 24 -pin ATX Power Connector
External I / O Port	2 * High Speed USB Connectors @ 480 MBit/ s
	1 * Serial Port / 1 * D-Sub 15-pin VGA Connector
	1 * PS / 2 Mouse & 1 * PS/2 Keyboard / 1 * RJ45 Connector / 1 * Audio I / O
BIOS	Award 4 MB Flash ROM
CD / AUX Audio IN & SPEAK Out	
Power Requirement	Standard 24 -Pin ATX Power supply

- Abstandssensor
- Restliche nichtelektronische Bestandteile des Fahrzeugs
- Videosensor Webcam

10.3 Produktschnittstellen

Kamera-Prozessor-Schnittstelle: Über diese Schnittstelle werden die Bilder von der Kamera auf den Prozessor übertragen.

Sensor-Prozessor-Schnittstelle: Über diese Schnittstelle werden die gemessenen Werte von den Sensoren auf den Prozessor übertragen

Motor-Prozessor-Schnittstelle: Über diese Schnittstelle werden die Befehle vom Prozessor auf den Motor übertragen, um vorwärts zu fahren bzw. stehen zu bleiben.

Räder-Prozessor-Schnittstelle: Über diese Schnittstelle werden die Befehle vom Prozessor auf die Räder übertragen, um links oder rechts abzubiegen oder geradeaus zu fahren. Weitere Details werden zu einer späteren Phase des Projekts spezifiziert.

11 Glossar

Richtigkeit: [1]: Das Programm sei richtig, wenn es nach der Spezifikation der Funktionen verhält, die es unterstützen soll.

Zuverlässigkeit [1]**:** Eine Software sei zuverlässig, wenn der Benutzer sich auf sie verlassen kann. Oder auch die Wahrscheinlichkeit dass die Software über eine bestimmte Zeitspanne wie erwartet funktioniert.

Robustheit [1]**:** Die Software sei robust, wenn sie sich vernünftig, auch bei unerwarteten Situationen, verhält.

Leistung [1]**:** Eine Software ist dann effizient, wenn sie Rechenressourcen, wie z.B. Rechenzeit oder Speicher, sparsam verwendet.

Benutzerfreundlichkeit [1]**:** Eine Software sei benutzerfreundlich, wenn der Benutzer sie leicht bedienbar findet.

Überprüfbarkeit [1]**:** Eine Software sei überprüfbar, wenn deren Eigenschaften leicht überprüft werden können.

Wartbarkeit [1]**:** wird verwendet, um auf die Änderungen, die an einem Software-System nach dem ersten Release, zu beziehen.

Reparierbarkeit [1]**:** Eine Software sei reparierbar, wenn sie die Korrektur von Fehlern innerhalb einer begrenzten Menge an Arbeit zulässt.

Entwicklungsfähigkeit [1]**:** Eine Software muss in der Lage sein, neue Verwaltungs- und Organisationstechniken unterzubringen.

Wiederverwendbarkeit [1]**:** Ist das Wiederverwenden von Komponenten an verschiedenen Stellen im Projekt.

Portabilität [1]**:** Ist die Fähigkeit auf unterschiedliche Hardwareplattformen bzw. Betriebssysteme zu laufen.

Verständlichkeit [1]**:** Eine Software sei verständlich falls sie ein vorhersagbares Verhalten hat.

Interoperabilität [1]**:** Beschreibt die Fähigkeit eines Systems, das mit anderen Systemen koexistiert und kooperiert.

Produktivität [1]**:** Misst die Effizienz des Prozesses.

Pünktlichkeit [1]**:** Die pünktliche Lieferung eines Produkts.

Sichtbarkeit [1]**:** Eine Software sei sichtbar, falls alle Schritte und aktuellen Status deutlich dokumentiert sind.

mini-ITX [2]**:** Ein Formfaktor für Computer-Mainboards, der kleiner als Mainboards aus der herkömmlichen ATX-Reihe.

OpenCV[3] ist eine kostenlose Programmbibliothek von Intel. Sie ist für die Programmiersprachen C und C++ geschrieben und enthält Algorithmen für die Bildverarbeitung.

[1] Fundamentals of Software Engineering [1991]. Ghezzi, Jazayeri, Mandrioli.

[2] http://de.wikipedia.org/wiki/Mini-ITX

[3] http://de.wikipedia.org/wiki/Opencv