FORUM Technologie & Gesellschaft

Eine Initiative des FORUM46 – Interdisziplinäres Forum für Europa e.V.

„Chancen und Risiken in der Wagnisgesellschaft“

Dokumentation der Veranstaltung am 15. Oktober 2014 in der Bundesanstalt für Materialforschung und -prüfung (BAM), Berlin

Die Tagung „Chancen und Risiken in der Wagnisgesellschaft“
wurde ermöglicht durch die freundliche Unterstützung von:

VDI BAM spaceopal

INHALT

Das **FORUM Technologie & Gesellschaft** ist eine Initiative getragen vom
FORUM46 – Interdisziplinäres Forum für Europa e. V.
Kontakt: Dr. Bernd Schulz-Forberg
bernd.schulz-forberg@forum46.eu
Dokumentation: Tiemo Ehmke
Fotos: Stefan Hertzke
Illustration S. 3:
© Karl-Heinz Höppner, Dipl.-Grafik-Designer AGD, Nordfriesland

Postfach 640237
D-10048 Berlin
www.forum46.eu

PROGRAMM

„Chancen und Risiken in der Wagnisgesellschaft“
Mittwoch, 15. Oktober 2014, 19:00
Bundesanstalt für Materialforschung und -prüfung (BAM)

Begrüssung und Impulsvortrag

Einleitung von **Bernd Schulz-Forberg,**
Leiter des FORUM Technologie & Gesellschaft Berlin

„Die technologische Zivilisation am Scheideweg.
Wieviel darf man wagen, um zu gewinnen?“
Walther Ch. Zimmerli, Technikphilosoph, Berlin

Der Blick über den Tellerrand – Methodische Ansätze

Die Rolle von Ingenieuren in der „brave new world“
der Technologien
Aleksandar Jovanovic – Steinbeis Advanced Risk Technologies,
Universität Stuttgart ZIRIUS

Risikodiskussion bei technischen Risiken in der Schweiz
Raymond Dumont – Amt für Verbraucherschutz,
Kanton Aargau, CH

Der Aufbau einer Risiko-Sektion in den Niederlanden
Jan Meissen – Risicobeheer en Techniek,
Koninklijk Instituut Van Ingenieurs, NL

Wer wagt, gewinnt? Berichte aus Praxis und Gesellschaft

Chancen & Risiken der Energiewende
Michael Limburg, Europäisches Institut für Klima und Energie,
Potsdam

Raumfahrt zwischen Höhenflug und Risiko
Christian Langenbach, Spaceopal GmbH, München

Zur Rolle des Finanzsektors in der Wagnisgesellschaft
Thomas Meyer, Deutsche Bank Research, Frankfurt/Main

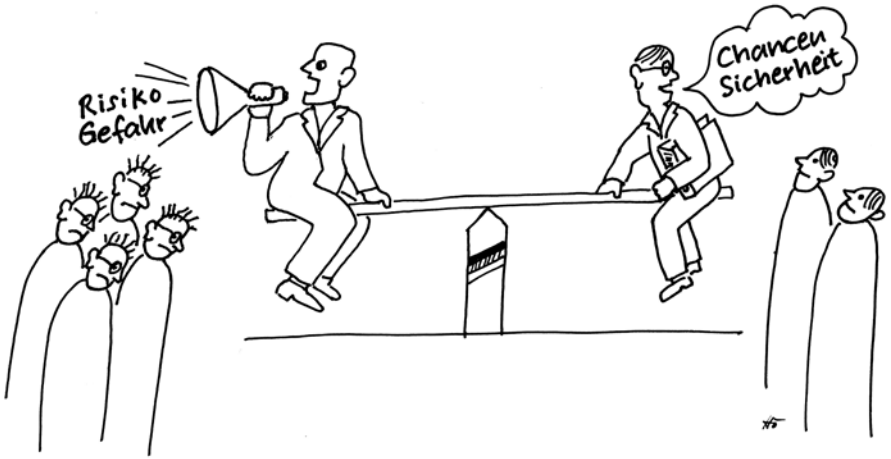
Gesellschaftliche Relevanz der Informatik als Strukturtechnologie
Hubert B. Keller, Institut für Angewandte Informatik, KIT,
Karlsruhe

No risk no fun? – better safe than sorry!

Grusswort des Präsidenten der Bundesanstalt für Materialforschung und -prüfung (BAM)

Abschlusstalk

Anschliessend Empfang



BEGRÜSSUNG

Ulrich Panne, Präsident der Bundesanstalt
für Materialforschung und -prüfung (BAM)

Meine sehr geehrten Damen und Herren, ich freue mich, Sie an der Bundesanstalt für Materialforschung und –prüfung, der BAM, begrüßen zu dürfen. Die BAM ist eine Bundesressortforschungseinrichtung im Geschäftsbereich des BMWi mit der Mission „Sicherheit und Technik und Chemie“. Für eine Diskussion zu Chancen und Risiken in der Wagnisgesellschaft mithin ein passender und inspirierender Ort.

Gestatten Sie mir, Ihnen die BAM kurz vorzustellen. Die BAM forscht, berät und prüft für konkrete gesetzliche Aufgaben und allgemeine Aufgaben im gesellschaftspolitischen Sinn ihrer Mission. Diese Arbeiten lassen sich in drei Bedarfsfeldern darstellen: Energie, Infrastruktur und Umwelt. Ermöglicht wird die Arbeit in diesen Feldern durch zwei tradierte Themenfelder, die Schlüsseltechnologien für den Auftrag der BAM sind und zu ihrer Alleinstellung beitragen: Material und Analytical Sciences.

Grundgesetz, Artikel 2 – das Recht auf Leben und körperliche Unversehrtheit.

Wenn wir uns dem Auftrag der BAM ganz grundsätzlich nähern wollen, müssen wir eigentlich beim Grundgesetz, Artikel 2 – dem Recht auf Leben und körperliche Unversehrtheit – beginnen, auch wenn er von den Verfassern sicherlich von einem anderen, heute nur historisch greifbaren Hintergrund geschrieben wurde. Viele gesellschaftliche Diskurse zu unterschiedlichen Technologien der vergangenen Jahrzehnte implizieren mit Bezug auf diesen Artikel, dass technische Sicherheit ein Bürgerrecht ist.

Wir müssen jedoch auch feststellen, dass im gesellschaftspolitischen Diskurs zur technologischen Basis unserer Gesellschaft der Verlust von Sicherheit für die Bürger allerdings selten existentieller Natur ist. Er reflektiert häufig eher einen Verlust an Lebensqualität und eine mögliche Fremdbestimmung.

Eine demokratisch legitimierte Industriepolitik umfasst heute auch den Auftrag Bürger vor Risiken aus natur- und ingenieurwissenschaftlicher Forschung und ihrer technischen Anwendung zu schützen und transparent die Risiken zu kommunizieren.

Die Einführung neuer Technologien in unsere Gesellschaft wird erst durch das Vertrauen der Bürger in ihre Sicherheit möglich. Die Geschwindigkeit der Umsetzung von naturwissenschaftlichen Erkenntnissen in Technologien und Produkte nimmt allerdings rasant zu.

Für Antworten auf entsprechende sicherheitstechnische Fragen ist die BAM in diesem Spannungsfeld der Zivilgesellschaft problemorientiert und praxisnah sichtbar positioniert. Sie leistet dafür an der Schnittstelle zwischen Politik, Verwaltung, Wirtschaft, Wissenschaft und Technik einen wichtigen Beitrag. Die Integration unterschiedlicher und anspruchsvoller Dienstleistungen verringert auch Transaktionskosten. Für die Unterstützung und Umsetzung politischer Entscheidungen sowie zur Wahrnehmung öffentlicher Aufgaben.

Die Einführung neuer Technologien wird erst durch das Vertrauen der Bürger in ihre Sicherheit möglich.

Gestatten Sie mir noch einige Gedanken zum Begriff „Wagnis". Ich habe im Vorfeld zu dieser Veranstaltung versucht festzustellen, welche Konnotationen dieses Wort für mich hat.

Um diesen Begriff abzuschmecken, hilft ein Blick ins Grimm'sche Wörterbuch. Der Begriff Wagnis taucht erst spät in der niederhochdeutschen Sprache auf, ab dem 16. Jahrhundert und hier vorrangig in der Kanzleisprache. In der Bedeutung als „gefährliche Sache" gerät das „Wagnis" aber schon in Blick staatlichen Handelns. Goethe hat das Wort seit 1803 gebraucht und hat später eine besondere Vorliebe dafür in der Bedeutung von „gefährliches Unternehmen". Alexander v. Humboldt schrieb über Kepler in seinem Kosmos beispielsweise „... ein solcher geist war ... geeignet, durch den Reichtum und die Beweglichkeit seiner Ideen, ja durch die Wagnisse kosmologischer Ahndungen, Leben um sich her zu verbreiten."

Wagnis war damals offensichtlich positiv besetzt. Das knüpft eigentlich mehr an das Verb an. Wagen gibt es hingegen schon viel länger und hat seinen Ursprung in der mittelhochdeutschen Dichtersprache. Wagen aus der Bedeutung von „in die Waage setzen", abwägen, taucht zuerst im Rolandslied auf, im Kontext „in das Leben wagen". Später fand dann eine Verschiebung der Bedeutung von einem kühnen Unternehmen zum Begriff „Gefahr laufen" statt.

Wagnis hat etwas mit Leben zu tun, mit einer Freiheit auch Fehler zu machen. Wagnis hat aber auch etwas mit Leistung zu tun, Wagnis ist ein Impulsgeber für Höchstleistungen auch für technische und wissenschaftliche. Wagnisverweigerung ist eine systemische oder individuelle Charakterschwäche. Die Freiheit, etwas zu wagen, ist also ein wichtiges Recht, ein Recht auf Selbstbestimmung und Würde. Und Verantwortungsfähigkeit kann man dabei nur beweisen, wenn sie überhaupt zugelassen wird. Unfälle sind bei Wagnissen nicht immer vermeidbar. Sie sind der – in gewissen Grenzen berechenbare – Risikoanteil bei jedem Wagnis. Wenn sie dann doch geschehen, sollten wir aber rückfragen, ob die erworbenen Kompetenzen und die Vorbereitung ausreichend waren.

Die Freiheit, etwas zu wagen, ist ein wichtiges Recht.

Meine Abschlussthese, die sicherlich nicht neu ist im Zusammenhang mit dem Begriff Wagnis, ist: Wagnis- oder risikoaverse Gesellschaften sind gefährdet. Eine gesunde Gesellschaft wächst im beherrschten Wagnis und Einrichtungen wie die BAM tragen zur Beherrschung von technischen und naturwissenschaftlichen Wagnisse bei.


„Chancen und Risiken
in der Wagnisgesellschaft"

VORWÄRTS GERICHTETE VERFAHREN UND DAMIT DAS DENKEN IN DIE ZUKUNFT SICHERN

Ein Fazit von Bernd Schulz-Forberg

Erfolge und Unfälle wurden jahrhundertelang als Schicksal begriffen. Heute ist klar, dass viele Vorgänge beeinflussbar und damit steuerbar sind. Aber allein aus rückwärtiger Betrachtung gelingt diese Steuerung immer weniger. Es ist vielmehr deutlich notwendig, die vorwärts gerichtete Betrachtung in Form von Analysen der Chancen und Risiken zumindest parallel Platz greifen zu lassen, vor allem vor dem Hintergrund von grossen Entwicklungssprüngen in Technik und Wirtschaft. Sie haben sich als vorwärts gerichtete Verfahren etabliert.

Risikoanalysen und -betrachtungen müssen systematisch in das verantwortbare Handeln eingebunden werden. Sie sind ein Teil des Standes der Technik und demzufolge anzuwenden. Das Risikomanagement als autonomes Verfahren oder besser als Bestandteil eines integrierten Managementsystems hilft, den notwendigen „kontinuierlichen Verbesserungsprozess“ in Sachen Sicherheit und Zuverlässigkeit und damit in der Verlässlichkeit zu erreichen. Die geeigneten Risiko- und Managementverfahren müssen von der Wirtschaft gewollt, weiterentwickelt und vorurteilsfrei sowie transparent angewendet werden.

Risikoanalysen und -betrachtungen müssen systematisch in das verantwortbare Handeln eingebunden werden.

Die technologische Zivilisation begibt sich von der Risiko- in die Wagnisgesellschaft und muss vor allem lernen, Entscheidungen auch auf der Basis von Nicht-Wissen zu treffen. Dies scheint manchem überspitzt formuliert, zeigt aber das eigentliche Problem sehr deutlich auf: wir verfügen über Unmengen von Daten, leiten daraus enorme Mengen von Informationen ab und bemühen uns, manchmal auch vergeblich, daraus verlässliches Wissen zu generieren.

Vorsorge um jeden Preis und vor allem ohne planmässige Korrektur über der Zeit führt zu Stillstand oder gar Rückschritt. Chancen

wahrnehmen und nicht auf Risiken achten führt sicher auch nicht zu nachhaltigem Erfolg. Auf die Balance zwischen Vorsorge und schlichtem *trial and error* kommt es an, auch wenn Wissen und Kontextwissen nicht in ausreichendem Masse vorhanden sind.
Vorwärts gerichtete Verfahren und damit das Denken in die Zukunft nehmen in allen Branchen und zugeordneten Fachgebieten zunehmend grösseren Platz ein. In der Technik gibt es eine Reihe von kodifizierten Verfahren, die sich etabliert haben und ständig zur Anwendung kommen. In der Finanzwirtschaft wird umfassend und detailliert geprüft, inwieweit Chancen und Risiken kalkuliert werden können. In der Medizin ist es Alltag, Statistiken heranzuziehen und die Chancen und Risiken von Therapien dezidiert zu kalkulieren. Schlägt man den Bogen größer und zieht alle Branchen von gesellschaftlicher Relevanz in die Betrachtungen mit ein, so erhält man einen Überblick über alle Vektoren, die zusammengenommen die Aussichten der Gesellschaft für die Zukunft abbilden.

Auf die Balance zwischen Vorsorge und schlichtem *trial and error* kommt es an.

Im Vereinigten Königreich, in den Niederlanden und in der Schweiz werden insbesondere bei Industrietätigkeiten schon seit längerer Zeit Risikobetrachtungen angestellt und die Analysen für Entscheidungen herangezogen. So werden einerseits Ansiedlungen von Industriebetrieben vorab kritisch untersucht oder andererseits auch einschneidende und damit auch kostenträchtige Maßnahmen ergriffen, um das Gesamt-Risiko einer solchen Ansiedlung im akzeptablen Bereich zu halten.

Konkret ist das Verfahren in der Schweiz für den Transport und die Lagerung von Flüssiggas in der Rheinebene angewandt worden. In der Bundesrepublik Deutschland ist dieses Verfahren bei dem Neubau der Startbahn am Flughafen Frankfurt/ Main angewandt worden – im Verlauf der Start – und Landebahn befand sich eine Chemiefabrik, die aus Risikobetrachtungen heraus verlagert wurde.

In der Raumfahrt wurde von Anfang an ein Risikomanagement eingeführt, um vor und während der Projektarbeit frühzeitig alle Gefahren identifizieren und analysieren zu können. Beispielsweise ist der Verlust eines ganzen Satelliten sehr viel gravierender als das Versagen eines Experimentes an Bord, was die Steuerung des Erfolgsrisikos für das ganze System effizient macht.

Im Finanzsektor gehört Chancen- und Risikoabwägung zum Kerngeschäft, die jetzt viel beschworenen Stresstests sind Ausdruck dieser Arbeitsweise. Eine Besonderheit in diesem Bereich ist vielleicht der Fakt, dass die Kombination von Risiken das Gesamt-Risiko senken kann. Vor 2008 war man optimistisch, mit den vorgesehenen Massnahmen – Stichwort Basel II – ausreichend Vorsorge getroffen zu haben. Zwischenzeitlich sind entsprechende Korrekturen vorgenommen worden, was auch Resultat umfangreicher Forschungsarbeiten ist.

Bei Fragen der Energiewende und des Klimaschutzes betritt man einen hochkomplexen Bereich, der in besonderem Maße das in die Zukunft gerichtete Denken herausfordert. Viele sind sich sicher, dass mit der Energiewende der Klimawandel bekämpft werden kann. Handelt es sich hierbei um Risikobewältigung oder um Chancenvernichtung? Schliesslich ist das Klima ein globales Phänomen und muss dann auch durch globales Handeln korrigiert werden. Der Diskurs zu diesem Thema muss unter Einschluss aller Aspekte ohne Vorurteile fortgesetzt werden. Es bleibt ein Reizthema besonderer Art.

Das Klima ist ein globales Phänomen und muss auch durch globales Handeln korrigiert werden.

Eng verbunden mit den Risikobetrachtungen und -analysen ist die Frage nach der Beherrschung des Risikos. In den Niederlanden hat die KIVI (Koninklijk Instituut Van Ingenieurs, also so etwas wie der „niederländische" VDI) den Aufbau einer Risiko-Sektion mit diesem Ziel begonnen und startete mit den Arbeitsgruppen *Analyse & Ent-*

wicklung, Praxis & Implementation und *Kommunikation & Information*. Anhand von konkreten Aufgaben wie Chancen und *Risiken der Nano-Technologie*, *Entwurf der Tore des neuen Panama-Kanals* und Begleitung des Grossprojekts *Bau der Metro in Amsterdam* bereitet die Sektion die Abstraktionen der Vorgehensweise für die Risiko-Beherrschung vor.

Denkweisen und Verfahren, Massnahmen und Methoden erlauben es heute, prognostische Aussagen für die Gesellschaft als Ganzes zu treffen. Alle Branchen und Fachgebiete können eingebunden werden, um das gesamte Kraftfeld der Gesellschaft darzustellen und voraussichtliche Änderungen in naher Zukunft zu beschreiben. Die EU hat *ETPIS* (European Technology Platform on Industriel Safety) aufgelegt und ein virtuelles Institut *EU-VRi – The European Virtual Institute for Integrated Risk Management* zeigt auf einer Meta-Ebene die Risiken aus fast allen Bereichen und vor allem die Entwicklung in der nahen Zukunft *(E2R2:European Emerging Risk Radar)* auf .

Alle Branchen und Fachgebiete können und müssen eingebunden werden.

Die Funktionalität heutiger technischer Systeme basiert in allen Lebenswelten in den wesentlichen übergeordneten Funktionen auf der Informatik. Gleichzeitig wird der Träger dieser Funktionen, die Hardware, nahezu verschwindend klein. Die Produktion von Software hat noch lange nicht den Stand wie in den klassischen Ingenieursdisziplinen erreicht. Weiterhin tritt durch die intensive Vernetzung aller Systeme, vom Auto über Haushaltsgeräte bis zu kritischen Infrastrukturen ein immenses und zurzeit nicht beherrschtes Risiko im Bereich der Cybersicherheit auf. Informatik entwickelt sich zu einer herausragenden Kulturtechnik, die von jedem Menschen wie Schreiben, Rechnen und Lesen grundlegend beherrscht werden muss. Zu diesem Bereich findet sich in dieser Dokumentation ein ausführlicher Bericht, so dass an dieser Stelle schon wegen der Alltagsbedeutung ausdrücklich auf diesen hingewiesen wird.

Wenn wir uns von der Risiko- in die Wagnisgesellschaft bewegen, dann heisst das auch, dass wir zunächst einen gesellschaftlichen Konsens über die zulässige Höhe der möglichen Konsequenz benötigen, um dann mit den Risikoverfahren Vergleiche durchführen zu können. Wir sollten dabei Lösungen zurückweisen, die grundsätzlich unbegrenzt in ihrer Auswirkung sein können. Es gilt, Lösungen zu bevorzugen, die eine gewisse Überschaubarkeit garantieren. In jedem Fall sollten die Vorgänge einfach prüfbar und die Reversibilität muss gegeben sein.

Zwei große Richtungen sind zu beachten und zu verfolgen. Die Methoden in den verschiedenen Bereichen werden weiterentwickelt, wobei die Erfolge aus Nachbarstaaten und verschiedensten Fachbereichen im Hinblick auf gegenseitige Befruchtung kommuniziert und einbezogen werden.

Es gilt, Lösungen zu bevorzugen, die eine gewisse Überschaubarkeit garantieren.

Parallel dazu wird der laufende Prozess einer europaeinheitlichen Vorgehensweise von den Ausformungen in einigen Staaten derart getriggert, dass die Beiträge der Kommission deutlicher und rascher zum Tragen kommen. In beiden Fällen ist es nötig, dass Staat und Wirtschaft noch enger und offener kooperieren, und zwar unter Beteiligung sämtlicher Stakeholder.

In Deutschland bietet der Verein Deutscher Ingenieure (VDI) die geeignete Pattform zum Diskurs über die Fragen zur Anwendung der Technik, ihrer Chancen und Risiken. Von hieraus kann auch ein Beitrag auf der Meta-Ebene aller Disziplinen geleistet werden, die sich in verschiedenen Foren finden und artikulieren werden. Das Forum Technologie & Gesellschaft im Forum46 (Interdisziplinäres Forum für Europa) wird weiterhin interdisziplinäre und europäische Diskurse anregen.

GESELLSCHAFTLICHE RELEVANZ DER INFORMATIK ALS STRUKTURTECHNOLOGIE

Hubert B. Keller, Karlsruher Institut für Technologie (KIT)
Institut für Angewandte Informatik (IAI)

1. Unsichtbare Automation als Rückgrat der Gesellschaft

90 % aller Computer sind nicht im PC, sondern als „eingebettete Systeme" hochintegriert in die zu automatisierenden oder überwachenden Systeme wie ABS, ESP, Herzschrittmacher usw. untrennbar eingebaut oder automatisieren kritische Infrastrukturen wie Verkehr, Wasser, Strom etc. (/ART2006/). Die Informatik spielt die Schlüsselrolle bei solchen Systemen mit höchsten Zuverlässigkeits- und auch sicherheitskritischen Anforderungen. Diese Systeme sind allerdings immer mehr vernetzt, offen und damit gefährdet (siehe Stuxnet Angriff). Die gleichzeitig ablaufenden, teils sicherheitskritischen Automationsaktivitäten werden quasi unsichtbar abgearbeitet. Alle Welt spricht dennoch nur vom Internet und den Anwendungen dort und sieht nicht die grundlegende Rolle einer allumfassenden, vernetzten Automatisierung.

Die Informatik spielt die Schlüsselrolle bei Systemen mit höchsten Zuverlässigkeits- und sicherheitskritischen Anforderungen.

Computer verschwinden auch größenmäßig, d. h. wir verlassen uns auf unsichtbare Computer ohne direkten Zugriff durch einen Menschen. Unser Leben und unsere Gesundheit hängen damit ab vom absolut korrekten Funktionieren dieser Systeme unter allen Umständen. Der Zentralverband der deutschen elektrotechnischen Industrie hat 2006 die Informationsverarbeitung unter Echtzeitbedingungen sowie Securityanforderungen auf solchen „Automatisierungs-Computer" als eine der zukünftigen Herausforderungen definiert (/ZVEI2006/). Die eingesetzten Softwaresysteme, z. B. im

Auto, regeln nicht nur, sondern treffen auch höhere Entscheidungen wie automatisches Bremsen aufgrund automatisierter Interpretation von Messgrößen. Die Vernetzung dieser Systeme führt zu einem Security- und nachfolgend auch zu einem Safety-Problem. Dies betrifft alle automatisierten Bereiche wie Auto, Zug, Gebäude, verfahrenstechnische Anlagen, kritische Infrastrukturen etc. Die Sicherheit in der Vernetzung (Security) berührt also direkt die Sicherheit des technischen Systems und seiner Umgebung (Safety).

Die Zuverlässigkeit komplexer vernetzter Software muss dabei aber sehr deutlich in Frage gestellt werden. Dies sei anhand der folgenden Sachverhalte verdeutlicht:

- Fehler in Software vernichtet in 45min 440 Millionen Dollar durch automatisierten Kauf von Aktien für 4,5 Mrd.
- Ein Pumpsystem als Lockvogel führte innerhalb von 28 Tagen zu 39 Angriffen, zwölf Angriffe gezielt, 13 wurden an mehreren Tagen wiederholt, der erste Angriff fand nach nur 18 Stunden statt. (VDI-Nachrichten vom 25.6.2013)
- Remote Motor Control per SMS kann über C2C (Car to Car Kommunikation) zum Bremsen und Beschleunigen von fremden Fahrzeug eingesetzt werden, ohne dass der dortige Fahrer eingreifen kann (gilt auch für andere Anlagen).
- Ein PKW Hersteller hatte Probleme wegen falscher Interpretation von Lenksignalen, die elektronische Dämpfung unterband Lenkbefehle des Fahrers.
- Sicherheitsrouter haben Lücken und erlauben die Zugangsdaten einschließlich Passwörter zu extrahieren.
- VPN (Virtual Private Network) erlauben Privilegien durch nicht spezifizierte Kommandobefehle zu erreichen.

Die Vernetzung dieser Systeme führt zu einem Security- und nachfolgend auch zu einem Safety-Problem.

Wie mit diesen Sachverhalten die Zielsetzung einer „Industrie 4.0“ mit dem Szenario einer offenen und vernetzten Automatisierungslandschaft zuverlässig erreicht werden kann ist nicht nachvollziehbar. Unter Industrie 4.0 wird z. B. beim ZVEI (/ZVEI2006/) die Fabrik der Zukunft mit einer hohen Anzahl vernetzter Embedded Systeme, mit autonomen Produkten, die mit dem Herstellungsprozess interagieren, mit intelligenten und autonomen Maschinen und Produkten verstanden. Alle Elemente dieser Fabrik besitzen einen virtuellen Geist (Cyber Physical System – CPS) mit Intelligenz und Entscheidungskompetenz, alles ist vernetzt und interagiert miteinander. Im Journal Computer-Automation (/CA2012/) wird dabei zumindest die IT-Sicherheit als Hemmnis auf dem Weg zu Smart Factories gesehen. Die VDI-Nachrichten (/VDI2013/) sehen zukünftig nur noch CPS statt Maschinen und als Ergebnis die Optimierung ganzer Wertschöpfungsketten. Mit einem kollaborativen Engineering und hoch verfügbarer IT sollen Apps zur einfachen Analyse und Services in der Cloud zur Beherrschung ausreichen. Manche träumen auch davon, Kontrollaufgaben in die Cloud zu verlagern. Die Zeitschrift Chemie & More (/Che2013a/, /Che2013b/) erwartet autonome Automatismen als Grundlage, um intelligente Fabriken mit hoher Vernetzung und daraus resultierender globaler Intelligenz zu realisieren. Dies führt dann zu Anlagen, die sich selbst anpassen und eine hohe Flexibilität besitzen.

Die Informatik ordnet zukünftig jeder realen Anlagenkomponente einen virtuellen Geist mit Intelligenz und Autonomie zu.

Konkret bedeutet dies, dass die Informatik zukünftig jeder realen Anlagenkomponente einen virtuellen Geist mit Intelligenz und Autonomie zuordnet, der mit allen anderen vernetzt ist. Jedes Produkt erhält ebenfalls einen virtuellen Geist und interagiert mit den CPS der Anlagenkomponenten. Die Vernetzung ist vollständig in horizontaler und vertikaler Sicht und Broker stellen alle Daten in Echtzeit allen vernetzten Komponenten zur Verfügung. Die intelligenten, vernetzten und autonom interagierenden Komponenten entscheiden situationsabhängig.

Was sind daraus folgend die technischen Herausforderungen? Die erste Ebene stellt die Zuverlässigkeit und Beherrschbarkeit lokaler, algorithmischer und intelligenter Verfahren dar (z. B. Bereich Computational Intelligence Methoden, siehe /Kel2000/).
Die nächste Ebene betrifft die Beherrschbarkeit vernetzter, synergistischer Wechselwirkungsfunktionalität. Hochvernetzte Systeme mit nichtlinearem Interaktionsverhalten lassen sich nach dem klassischen „Teile und Herrsche" Prinzip der Ingenieurswelt nicht analysieren und verstehen.

Hochvernetzte Systeme lassen sich nach dem klassischen „Teile und Herrsche" Prinzip der Ingenieurswelt nicht verstehen.

Weitere Aspekte sind das Verständnis, die Vorhersagbarkeit, das Zeitverhalten und der Determinismus vernetzter CPS-Systeme, denn in der Automatisierung ist ein zeitgerechtes und deterministisches Verhalten grundlegend. Durch die Vernetzung und Öffnung entstehen sowohl Fragen nach der Zuverlässigkeit als auch der Security verteilter und hochgradig wechselwirkender Komponenten im Cyber Space. Daraus folgend sind dann auch die Auswirkungen von Security Defiziten auf die Zuverlässigkeit und Safety zu betrachten (/Kel2013/).

Daher stellen sich Fragen nach der konstruktiven Basis für zuverlässige, intelligente Algorithmen, welche Konzepte die Informatik als konstruktive Basis für sichere, verteilte Architekturen bietet und wie die Beherrschung der resultierenden Verhaltenskomplexität zu sichern ist.

Die Automatisierung technischer Systeme, von kritischen Infrastrukturen bzw. sicherheitskritischen Anwendungen stellt das Rückgrat unserer Gesellschaft in allen Lebensbereichen dar (Energie, Wasser, Verkehr, Medizin, ...). Die Zuverlässigkeit und die Beherrschbarkeit sind zentral für das Funktionieren dieser Infrastrukturen, sehr oft ist die Kontrolle sicherheitskritischerProzesse (über-)lebenswichtig.

Die Automatisierung von kritischen Infrastrukturen stellt das Rückgrat unserer Gesellschaft dar.

Die Lebensdauer der Systeme ist hoch, das Alter und die Strukturen existierender Systeme und deren Softwarearchitekturen ebenfalls. Der Entwurf dieser Systeme ist historisch begründet und entspricht daher keineswegs heutigen Anforderungen. Aber auch heute entworfene Systeme berücksichtigen oft nur die klassischen IT-Sicherheitsaspekte im Sinne äußerer Kontrolle oder passiver Maßnahmen (/BSI2012/). Bei komplexen, vernetzten Systemen kommt zusätzlich die Frage nach der mentalen Beherrschbarkeit der Funktionalität durch den Bediener hinzu.

2. Software Engineering

Software-basierte Funktionen unter Echtzeitbedingungen sind das zentrale Element jeglicher Automatisierung – ob im Auto, in der Medizin oder in der Verfahrenstechnik. Die Fusion von Informationen mit einer nachfolgenden interpretativen Auswertung, eine hohe Wechselwirkung mit überlappend verschachtelten und dynamisierten Informationsflüsse, einer massiven Öffnung nach außen und einer systemischen Vernetzung muss auf der Basis zuverlässiger, robuster und beherrschbarer Software erfolgen.

Dabei ist Software ein Produkt mit hohem Einfluss – als Innovationsfaktor wie auch als Kostenfaktor. In allen technischen Bereichen wirkt Software als massiver Innovationstreiber (/Ros2003/). Gleichzeitig birgt der Einsatz dieser Funktionalität auch Kostenrisiken. Innovationssteigerungen von 35 % durch Software stehen gleichzeitig Kosten in gleicher Höhe – oder mehr – dagegen (/FAST2005/).

Software ist ein Produkt mit hohem Einfluss – als Innovationsfaktor wie auch als Kostenfaktor.

Hinzu kommt nun die massive Anfälligkeit im Securitybereich. Die Absicherungen gegen Cybercrime Attacken führen zu weiteren Kostensteigerungen. Worin liegen diese Effekte ursächlich begründet und wie sieht eine solide ingenieursmäßige Lösung der Problematik aus?

Zwei Aspekte treten hier in Vordergrund:

- Der erste Aspekt betrifft offene Türen durch eine analytisch-funktionale Nichtbeherrschung der Komplexität der Software zur Automatisierung. Fragen sind dabei, wie viele zu testenden Abläufe in einer Software gibt es? Welche Parameter eines Systems wirken sich unsicher aus? Wie stelle ich fest, ob die Kommunikationsstrukturen sicher sind?
- Der zweite Aspekt betrifft die Güte der Implementierung. Wie fehlersicher ist die Programmiersprache (Syntax)? Wie sicher sind die Programmiersprache und die Implementierung bzgl. der Prüfung von algorithmischen Eigenschaften (Semantik, Indexbereiche von Vektoren, z. B. in C und C++ gegenüber Ada)?

Zentrale Aspekte sind beim Einsatz der Informatik zur Realisierung Software-basierter Funktionen die Zuverlässigkeit und die funktionale Beherrschbarkeit komplexer Softwaresysteme. Die Zuverlässigkeit betrifft die intendierte Funktion (Zielfunktion) beim bestimmungsgemäßen Gebrauch und bei Störungen über einen definierten Zeitraum und ist nicht mit der Verfügbarkeit zu verwechseln. Dabei ist die klassische Zuverlässigkeit nicht auf Software anwendbar, da das Kontinuumsgesetz nicht gilt. Bei mechanischen Systemen bedeutet eine kleine Lasterhöhung eine entsprechend kleine Veränderung im Zustand. Ein typisches Beispiel ist hier die Durchbiegung eines Balkens bei einer kleinen Lastveränderung. Bei Software-basierten Funktionen kann allerdings eine minimale Änderung der Eingangsdaten eine unabsehbare Veränderung bei den Ausgangsdaten bewirken.

Die klassische Zuverlässigkeit ist nicht auf Software anwendbar, da das Kontinuumsgesetz nicht gilt.

Die Beherrschbarkeit betrifft das Verständnis über das Verhalten des Systems und dessen Kontrollierbarkeit durch den Benutzer. Ein fehlerhaftes Verhalten kann auch durch eine falsche Bedienung bei korrekter Funktion auftreten.

Software muss als Produkt mit entsprechend notwendigen Produktionsprozessen, Organisationsabläufen und eingesetzten Methoden auf Basis einer adäquaten Konzeption verstanden werden. Die Entwicklung von Software ist ein komplexer Herstellungsprozess, der weit vor einer Programmierung mit der Analysephase beginnt. Vorgehensmodelle definieren, welche Schritte wann zu erfolgen haben. "If you don't know where you're going, you're unlikely to end up there." (Forrest Gump) führt zur Erkenntnis, dass wir erst einmal wissen müssen, was wir wollen. Ansonsten scheitern Software-Projekte massiv(/Stan2009/). Die Analysephase erfolgt oft zu kurz und mit zu geringem Aufwand. Das Ergebnis zeigt die HSE Analyse von Softwarefehler: "44 % had inadequate specification as their primary cause" (/HSE2003/). Die Fehler werden in frühen Phasen injiziert und produzieren massiv späte Kosten (/Cross2005/), d. h. früher Aufwand reduziert Kosten und erhöht entsprechend den Gewinn.

Die Fehler werden in frühen Phasen injiziert und produzieren massiv späte Kosten.

Die Produktion von Software erfordert aber auch adäquate Sprachen und Laufzeitsysteme zur Implementierung sowie Betriebssysteme zum Einsatz der Software. High Level Features in Programmiersprachen sind unnütz, wenn nicht die grundlegende mathematische Eigenschaft elementarer Objekte, z. B. die Indexgrenzen eines Vektors, durch die eingesetzte Programmiersprache gesichert werden kann (/ISO2012/). Bereichsverletzungen, auch bei Rückgabewerten von Prozeduren (call frame), sind das grundlegende Übel, warum Viren etc. überhaupt funktionieren. Im Gegensatz zur Trennung von Daten und Anweisungen zur Erhöhung der Sicherheit in den frühen Jahren der Informatik, führen diese Probleme zu den „offenen Scheunentoren" von Software. Bei automatisierungstechnischen Systemen kommen zeitliche Randbedingungen sicherheitskritischer Funktionen hinzu, wie z. B. bei der Führung exothermer Prozesse oder der Regelung beim ESP.

3. Modellierung und Implementierung

Die Kenntnis der Kernanforderungen ist die Grundvoraussetzung für weitere effiziente Entwicklungen in der Zukunft.

Schwierigkeiten bestehen in der Erkennung der Kernanforderungen einer softwarebasierten Problemlösung, um eine solide Architektur, d. h. die Bausteine und deren gegenseitigen Bezüge, zu definieren. Die Kenntnis der Kernanforderungen ist die Grundvoraussetzung für weitere effiziente Entwicklungen in der Zukunft. Nach dem Erarbeiten der Kernanforderungen wird das Modell der Lösung mit seiner Logik in der gewählten Struktur, den modellierten Abläufen etc. beschrieben. In dieser Phase müssen Anforderungen aus Security und Safety berücksichtigt und konzeptionell vorgesehen werden. Identitäten von Softwarekomponenten, private und öffentliche Schlüssel von Softwarebausteinen (Tasks etc.), Sicherheitslevels von Kommunikationsbeziehungen, Transportschichten mit virtuellen Kanälen und eine zentrale Sicherheitsinstanz zur Konfigurationsüberwachung sind in dieser Phase zu definieren. Werden Methoden wie UML zur Beschreibung eingesetzt, kann das Modell über Transformationen zur automatisierten Codeerzeugung (Programmierung) mit Anreicherung von Safety- und Securityeigenschaften über „stereotypes“ und Profile verwendet werden vgl. /Kel2002/). Zufällige Fehler eines Programmierers oder „geniale Lösungen“ eines Einzelnen sind hier ausgeschlossen. Damit wird auch im Softwarebereich aktives Knowledge Management betrieben, da alle Kenntnis über alles haben.

Jede Art programmtechnischer Realisierung erfordert den Einsatz einer Programmiersprache und deren Laufzeitsystem. Hier zeigt sich

die Fähigkeit einer Sprache, die modellierten Objekte in ihren Eigenschaften sauber umzusetzen, z. B. sind für numerische Berechnungen die Darstellungsgenauigkeiten explizit und unabhängig von Betriebssystem und Rechnerarchitektur anzugeben. Vektoren besitzen Indexbereiche, deren Einhaltung es zu prüfen gilt. Auch der syntaktische Abstand zwischen grammatikalisch als korrekt erkannten Programmanweisungen analog der Hammingdistanz ist für die Fehlervermeidung lebenswichtig. Die Programmiersprache C/C++ hat hier gegenüber der Sprache Ada erhebliche Defizite (Abb. 1 S. 36 und vgl. /ISO2012/).

Die Trennung von Daten und Anweisungen, die Einführung von Basisregistern und Grenzregistern zur Absicherung der Speicherbereiche, sind von der Sprache einzusetzen. Dann werden Texte als Daten und Anweisungen nicht beliebig tausch- und veränderbar. Die Welt des Cybercrime lebt von Buffer Overflow und der Manipulation von eigentlich zu schützenden Programmbereichen. Die „Nationale Roadmap Embedded Systems“ des ZVEI (/ZVEI2009/) fordert, „Herstellung und Aufrechterhaltung des Vertrauens in Embedded Systems sind unabdingbare Voraussetzung für die Akzeptanz von komplexen, vernetzten, eingebetteten Systemen, wie sie zur Lösung der gesellschaftlichen und ökonomischen Herausforderungen benötigt werden. Bisherige IT-Sicherheitskonzepte sind hier nützlich, aber nicht ausreichend, da sie oft auf den Aspekt Security fokussieren“.

Die Welt des Cybercrime lebt von Buffer Overflow und der Manipulation von eigentlich zu schützenden Programmbereichen.

Softwareprogramme, die einen Zahlenüberlauf ignorieren und damit korrekte Ergebnisse suggerieren, sind untragbar, in sicherheitskritischen Systemen lebensgefährlich. Auch Testen oder eine betriebsbedingte Bewährung helfen hier nicht weiter. Erstens treffen Tests nur eine Aussage hinsichtlich der getesteten Daten, zweitens ist die Betriebsbewährung nur in Routinenutzungen vorhanden und drittens treten die Fehler immer in kritischen Situationen auf, wenn eine besondere Verlässlichkeit der Software notwendig ist. Komplexe Software mit vielen Verzweigungen kann nicht erschöpfend getestet werden. Eine Modellrechnung des SEI (/Watt2008/) für Programme mit einem bestimmten Umfang an Anweisungen und entsprechend typischer Zahl von Verzweigungen zeigt, dass für ein Softwaresystem mit 400 Verzweigungen typisch etwa 1.38E+11 mögliche Pfade im Ablauf zu testen wären. Ein Programm mit 100 Millionen Programmzeilen kann damit in endlicher Zeit nicht getestet werden. Die Aussage „With the test-based software quality strategy, large-scale life-critical systems will be least reliable in emergencies – and that is when reliability is most important.“ fordert also konstruktive Maßnahmen, um Zuverlässigkeit zu erreichen. Das Betriebssystem Linux hat z. B. eine durchschnittliche cyclomatische Komplexität (Abb. 2 S. 37, vgl./Ram2004/) von 4,94, während einfache Ablaufstrukturen wie z. B. aus der „strukturierten Programmierung“ einen Wert von zwei bis drei haben. In Linux gibt es aber ca. 100 Funktionen mit einer cC von über 100 und sogar eine Funktion mit 2261 LOC und einer cC von 352 (/Gan2012/). Solche Funktionen sind nicht mehr prüfbar.

Komplexe Software mit vielen Verzweigungen kann nicht erschöpfend getestet werden.

Auch die Schlussfolgerung über Reifegradmodelle, dass Software bei langem Einsatz und abnehmenden Fehlverhalten stabil und zuverlässig sei, scheitert an der unzulässigen Extrapolation über die betrachteten Einsatzprofile im Gegensatz zur klassischen Mechanik (/Kel2013/). Die erhöhte Durchbiegung eines Balkens bei erhöhter Belastung kann extrapoliert werden, da das Kontinuumsgesetz gilt. Dies ist bei Software nicht gegeben, hier haben wir ein nicht stetiges Verhalten. Die Zuverlässigkeit von Software ist also konstruktiv zu sichern – logisch korrekt, einfach in der Struktur und nach Prinzipien erstellt, welche Fehler vermeiden. Robustes Verhalten hilft Restfehler zu beherrschen.

Durch Cyber Crime Attacken tritt eine weitere Dimension hinzu (/Kel2012/). Die Angriffe auf sicherheitskritische Strukturen zeigen die gegenseitige Abhängigkeit von Safety und Security Aspekten. Ein Remote Motor Control per SMS über Car2Car Kommunikation zum Bremsen und Beschleunigen fremder Fahrzeuge ist ein Hinweis auf die Problematik. Dies gilt analog auch für andere Anlagen und erfordert konzeptionell Architekturen, welche diese Aspekte berücksichtigen. Die Konsequenz wäre sonst, dass die Batterie eines Elektroautos ab Werk als eine durch Viren zündfähige Bombe zu betrachten ist.

Interessanterweise eignen sich Redundanzkonzepte wie diversitäre Software mit Voting auch zu Erkennung von Manipulationen. Gherbi (/Ghe2011/) setzt diversitäre Software ein, um an Inspektionspunkten die Korrektheit von Zwischenwerten (Verhaltensanalyse, siehe Bild 5) oder beim Ergebnis die Übereinstimmung von Berechnungen zu prüfen. Noch einen Schritt weiter geht Mottok (/Bra2012/), indem die Berechnungen aus dem reellen Zahlenbereich komplett in den Bereich der Primzahlen verlegt werden. Manipulation führen mit hoher Wahrscheinlichkeit zu Abbildungen in den reellen und damit außerhalb des zulässigen Primzahlenbereichs. Dabei stellt eine HW-Implementierung die zeitliche Leistungsfähigkeit sicher. Ein weiteres Konzept ist die Einführung einer rückwirkungsfreien Softwareschnittstelle zur Anbindung von COTS-Produkten oder beliebiger Fremdsoftware. Eine Stellvertreterkomponente übernimmt die Schnittstellenfunktion und die Interaktion mit der externen Software. Der Stellvertreter führt selbst keine Aktionen aus, sondern nimmt nur Eingaben an, prüft diese und wenn er korrekt seinen Zielzustand bei korrektem Prüfergebnis der Eingaben erreicht, so übergibt er das Ergebnis an den eigentlichen Ausführenden. Eine Manipulation des Stellvertreters führt maximal zu dessen Abbruch und Neustart. Hinzu kommen dann ergänzend die passiven Maßnahmen wie z. B. dem BSI Schutzhandbuch (/BSI2012/).

Die Angriffe auf sicherheitskritische Strukturen zeigen die gegenseitige Abhängigkeit von Safety und Security Aspekten.

4. FAZIT

Der zentrale Fokus muss auf konstruktive und organisatorische Maßnahmen zur Sicherstellung der Zuverlässigkeit liegen. Zusätzlich ist eine integrative Betrachtung von Safety und Security Aspekten notwendig und modellmäßig zu beschreiben. Die Implementierung kann dann über den „Model Driven Architecture"-Ansatz (/Kel2002/) oder manuell erfolgen. Wesentlich ist eine die Zuverlässigkeit und Sicherheit unterstützende Programmiersprache. Diese Programmiersprache sollte semantisch und syntaktisch vollständig definiert und fehlervermeidend sein und die Umsetzung des logischen Modells in allen Eigenschaften umfassend unterstützten. Dies erfordert eine strenge Typisierung, die Prüfung auf Index Bereiche mit Ausnahmeauslösung, eine syntaktische Klammerung für lesbare Strukturen und eine Fehlerdistanz in der Grammatik, welche Einfachfehler nicht als syntaktisch korrekt einstufen kann. Schreibfehler sollten keinen syntaktisch korrekten Code ergeben, hierzu müssen mindestens zwei oder mehr Fehler auftreten (/Kel2013/). Programmiersprachen müssen fehlervermeidend aufgebaut sein und Fehlverhalten auch zur Laufzeit verhindern (Exception).

Programmiersprachen müssen fehlervermeidend aufgebaut sein und Fehlverhalten auch zur Laufzeit verhindern.

Sicherheitskritische Systeme erfordern die konzeptionelle Umsetzung von Fehlertoleranzmechanismen zum Erreichen eines definierten sicheren Zustands. Methoden zur Erhöhung der Zuverlässigkeit wie diversitäre Software können gleichzeitig für die Analyse des Verhal-

tens als auch für Vergleiche an Inspektionspunkten unter Security Aspekten eingesetzt werden (/Ghe2011/). Hierzu sind die Aspekte von Safety und Security von Beginn an gemeinsam zu analysieren und in das Systemkonzept und das Modell zu integrieren (/Fre2014/).

Zur Beherrschung der Komplexität ist Wert auf eine durchschaubare und mental beherrschbare Funktionalität zu legen. Intelligentes Verhalten muss vorhersagbar und verlässlich sein (deterministisch in exaktem Kontext). Dies gilt insbesondere für die Kombination von komplexen Funktionen.

Die grundlegend zu erfüllenden Eigenschaften von Informatiklösungen in der Automatisierung sind rechtzeitig und zeitlich definiert, gleichzeitig, zuverlässig, deterministisch, sicher im Sinne Safety, sicher im Sinne Security, implementierungstreu – die mathematischen Eigenschaften der Algorithmen werden programmiersprachlich und vom Laufzeitsystem umgesetzt und gesichert, strukturtreu – ausgeführte Programmaufrufe sind tatsächlich vom echten Initiator angestoßen, benutzungsfreundlich, verlässlich im Sinne der Summe aller Eigenschaften und validierbar im Sinne der Komplexität und den intendierten Zielfunktionen. Hinzu kommt die Beherrschbarkeit, also mentale Nachvollziehbarkeit durch den Benutzer. Die von uns Menschen mit Hilfe der Informatik erzeugten Systeme genügen diesen Kriterien nicht und zwar ohne Forderungen nach Echtzeit und meist ohne Betrachtung der Fähigkeiten und Fertigkeiten des Menschen im Umgang damit. Denn wir vernetzen diese Systeme miteinander auf Basis unzuverlässiger Konzepte und Methoden und die resultierenden Verhaltensweisen ergeben sich nicht mehr nur aus dem System alleine, sondern aus den Wechselwirkungen dieser Systeme untereinander und in Verbindung mit dem Eingriff des Menschen (emergentes Verhalten). Hierzu muss die Informatik (noch) grundlegende Vorgaben leisten wie z. B.

Zur Beherrschung der Komplexität ist Wert auf eine durchschaubare und mental beherrschbare Funktionalität zu legen.

- Systemarchitekturen sind klar und einfach zu strukturieren
- Analyse- und Designphasen müssen ausgeweitet werden
- Logische Korrektheit muss Vorrang vor Effizienz haben
- Programmiersprachen müssen adäquate Konstrukte besitzen (Größen mit mathematischen Eigenschaften etc.)
- Programmiersprachen brauchen Laufzeitsysteme, welche diese Eigenschaften zur Ausführungszeit prüfen (Bereichsüberschreitung etc.)
- Funktionen müssen beherrschbar sein – insbesondere in Grenzbereichen

Die Frage ist, lassen wir uns vom Computer zentral steuern?

Die Automatisierung für die Gesellschaft ist bzw. wird allumfassend (Cyber-Physical-Systems). Die Frage ist dann aber, lassen wir uns vom Computer zentral steuern, damit wir uns als Masse intelligent bezogen auf die Anforderungen verhalten? Was bleibt dann als persönliche Handlungskompetenz? Bei welcher Komplexität kann die kognitive Last der einlaufenden Verhaltens- und Dateninformationen bei Vernetzung noch getragen werden (Verständnis der Systemstrukturen)? Das Ziel der Technologie muss Beherrschbarkeit der benutzten Systeme sein. Unsere Gesellschaft sollte und muss diese Entwicklungen diskutieren und zumindest Leitplanken setzen, sonst sind wir nur noch vom Computer geführte Nutzer unserer eigenen Technologie. Möglicherweise muss die intelligente Leistungsfähigkeit moderner Technologien, zumindest deren Komplexität, eingeschränkt werden.

Erweiterte schriftliche Fassung des Vortrags auf der Tagung „Chancen und Risiken in der Wagnisgesellschaft“ vom 15.10.2014,
Bundesanstalt für Materialforschung und -prüfung (BAM),
Unter den Eichen 42-44, 12205 Berlin

5. SCHAUBILDER

Abb. 1: Syntaktische und semantische Fehlerquellen

```
a = 3;
if (a = 4)
a = a + 1;
```

Result: a = 5 ?

Intended: (a == 4)

```
a = 3;
if (a == 4);
a = a + 1;
```

Result: a = 4 ?

(null-then-part)

Syntactic Identity in Ada:

```
a := 3;
if a = 4 then
        a := a + 1;
[else
        statement;]
end if;
```

Assignment direct after `if` and null-then-part not allowed!

Enumeration types in C (set of named integer constant values):

```
enum abc {A,B,C,D,E,F,G,H} var_abc;
```

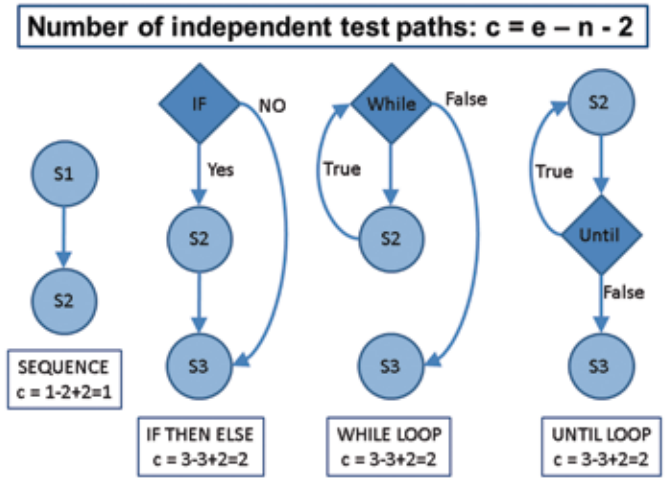
The values of the contents of abc would be A=0, B=1, C=2, etc.

C allows values to be assigned to the enumerated type as follows:

```
enum abc {A,B,C=6,D,E,F=7,G,H} var_abc;
```

This would result in: A=0, B=1, C=6, D=7, E=8, F=7, G=8, H=9

Abb.2: Cyclomatische Komplexität



c = e – n + 2

c = cyclomatic complexity
e = number of edges
n = number of nodes

6. REFERENZEN

/ART2006/ ARTEMIS Strategic Research Agenda. ARTEMIS SRA WG, 2006

/Bra2012/ Braun, J., Mottok, J., Miedl, C., Geyer, D., Minas, M.: Increasing the reliability of single and multi core systems with software rejuvenation and coded processing. In /Ploe2012/

/BSI2012/ Leitfaden Informationssicherheit, IT-Grundschutz kompakt. Bundesamt für Sicherheit in der Informationstechnik – BSI53133 Bonn, 2012, BSI-Br012/311

/CA2012/ Computer Automation "Internet und Automation: Was hinter Begriffen wie Industrie 4.0 steckt", 19.12.2012

/Che2013a/ Chemie&More "Industrie 4.0, Intelligent vernetzt", 1/2013

/Che2013b/ Chemie&More "Automatisierung, SmartFactory", 2/2013

/Cross2005/ CROSSTALK The Journal of Defense Software Engineering, December 2005, p. 16

/FAST2005/ F.A.S.T., TU München, 2005

/Fre2014/ Freiling, F.; Grimm, R.; Großpietsch, K.-E.; Keller, H. B.; Mottok, J.; Münch, I.; Rannenberg, K.; Saglietti, F.: Technische Sicherheit und Informationssicherheit – Unterschiede und Gemeinsamkeiten. Accepted for: Informatik Spektrum, GI, Springer, Ausgabe Nr. 1 (37) 2014, Seite 14-24

/Gan2012/ Ganssle, J.: The Way ahead in Software Engineering. In /Ploe2012/

/Ghe2011/ Gherbi, A. et al.: Software Diversity for Future Systems Security. CrossTalk, September/October 2011, p. 10ff

/HSE2003/ Health and Safety Executive: Out of Control, 2003

/ISO2012/ Information Technology – Programming Languages – Guidance to Avoiding Vulnerabilities in Programming Languages through Language Selection and Use. ISO/IEC TR 24772 Edition 2 (TR 24772 WG 23/N 0389), ISO/IEC JTC 1/SC 22/WG 23, 2012

/Kel2000/ Keller, H.B.; Maschinelle Intelligenz: Grundlagen, Lernverfahren, Bausteine intelligenter Systeme. Unter Mitarbeit von Fick, A.; Weinberger, T.; Gorges-Schleuter, M.; Eppler, W.; Schmauch, C.; Braunschweig [u.a.]: Vieweg, 2000 (Computational Intelligence)

/Kel2002/ Keller, H. B.; Benkle, M.: Model Driven Architecture als Basis zur modellbasierten automatischen Codegenerierung, Vortrag OMG Konferenz Model Driven Architecture, Darmstadt, 3./4.12.2002

/Kel2012/ Keller, H. B.: Im Kampf gegen Cybercrime. Chemie&More, 6/2012

/Kel2013/ Keller, H.B.: Entwicklung von zuverlässiger Software im Safety/Security Umfeld. Industrial IT Security 2013. IT-Sicherheit in Produktions- und Automations-Systemen, 03.07.13 bis 04.07.13, Frankfurt a.M.

/Ploe2012/ Plödereder, E.; Dencker, P.; Klenk, H.; Keller, H. B.; Spitzer, S. (Herausgeber): Proceedings Band 210 Automotive – Safety & Security 2012: Sicherheit und Zuverlässigkeit für automobile Informationstechnik Tagung 14.-15. 11. 2012 in Karlsruhe, Bonn, Gesellschaft für Informatik e. V.

/Ram2004/ Ramberger S.; Gruber T.: Error Distribution in Safety-Critical Software & Software Risk Analysis Based on Unit Tests. Experience Report. WSRS Ulm – 20 Sept. 2004. ARC Seibersdorf research GmbH

/Ros2003/ Rosenstiel, W.: Abschlussbericht DFG-Schwerpunktprogramm 1040: Entwurf und Entwurfsmethodik eingebetteter Systeme, Universität Tübingen, 1997 – 2003 / BMW AG

/Stan2009/ Standish Group: CHAOS Summary 2009. The Standish Group International, Inc., www.standishgroup.com

/VDI2013/ VDI-Nachrichten 1.2.2013

/Watt2008/ Watts S. Humphrey: The Software Quality Challenge. Journal of Defense Software Engineering. The Software Engineering Institute (2008).

/ZVEI2006/ Integrierte Technologie-Roadmap Automation 2015+ (ZVEI Automation 2006)

/ZVEI2009/ Nationale Roadmap Embedded Systems. ZVEI – Zentralverband Elektrotechnik und Elektronikindustrie e. V., Kompetenzzentrum Embedded Software & Systems, Frankfurt, Dezember 2009