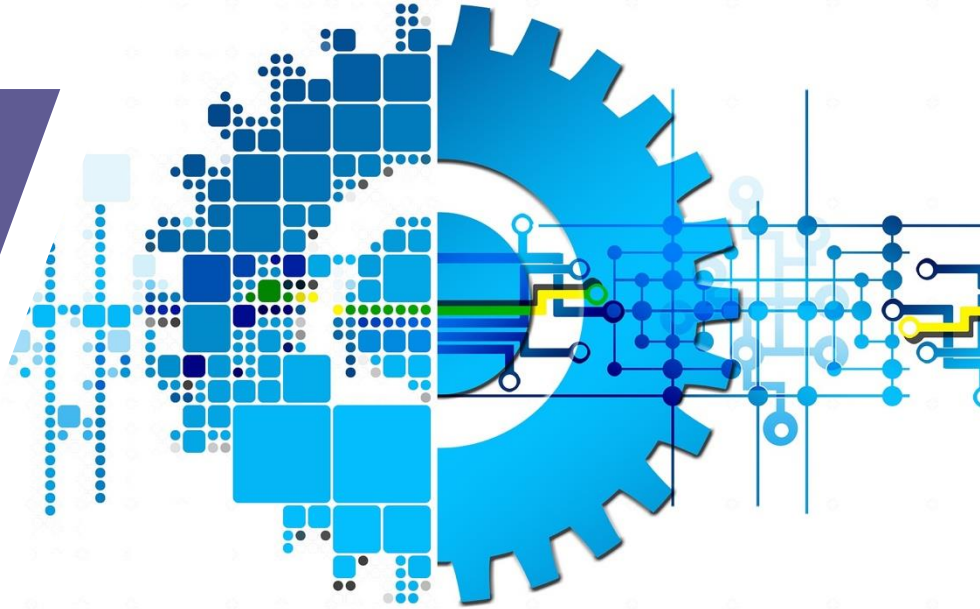
Universität Siegen

Naturwissenschaftlich-Technische Fakultät

Digital Engineering



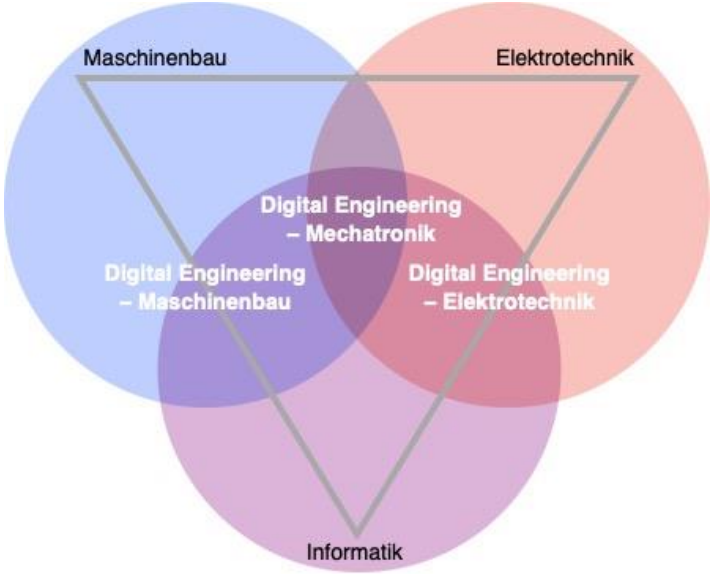
www.uni-siegen.de

Digital Engineering

Idee: Integrierte Studiengänge Ingenieurwesen & Informatik

- Informatik-Anteile werden für Ingenieure immer wichtiger. Ein bisschen mehr Informatik im MB/ET-Studium reicht nicht.
- Herausforderungen der Digitalisierung (insb. Software) und Industrie 4.0 mitgestalten.
- Ingenieure und Informatiker lernen von Anfang an *anders* zu denken. Beispiel: Stark unterschiedliche Mathematik.

  → Von Anfang an beide Denkweisen schulen: Differentialgleichungen (ET/MB) & Diskrete Mathematik (INF), daher schon im Bachelor!
- Pluspunkte: Alleinstellungsmerkmal, großer Bedarf in Wirtschaft,
  Software & KI goes Engineering, Treiber für Industrie 4.0.



Motivation

- „Digital-vernetztes Innovieren erfolgt im Grenzbereich der Disziplinen, insbesondere von Ingenieuren und IT-Spezialisten.
  Dafür sind Offenheit und **interdisziplinäres** Verständnis notwendig“ [VDMA].
- Forderung des Verbands Deutscher Maschinen- und Anlagenbau (VDMA) und dessen Impuls-Studie „Ingenieurinnen und Ingenieure für **Industrie 4.0**“, Januar 2019.
- Beispiel: „Wert an Fahrzeug-**Software** pro Fahrzeug wird sich in den nächsten 10 Jahren **verdreifachen**“ [Berylls].

Digital Engineering

3 Studiengänge

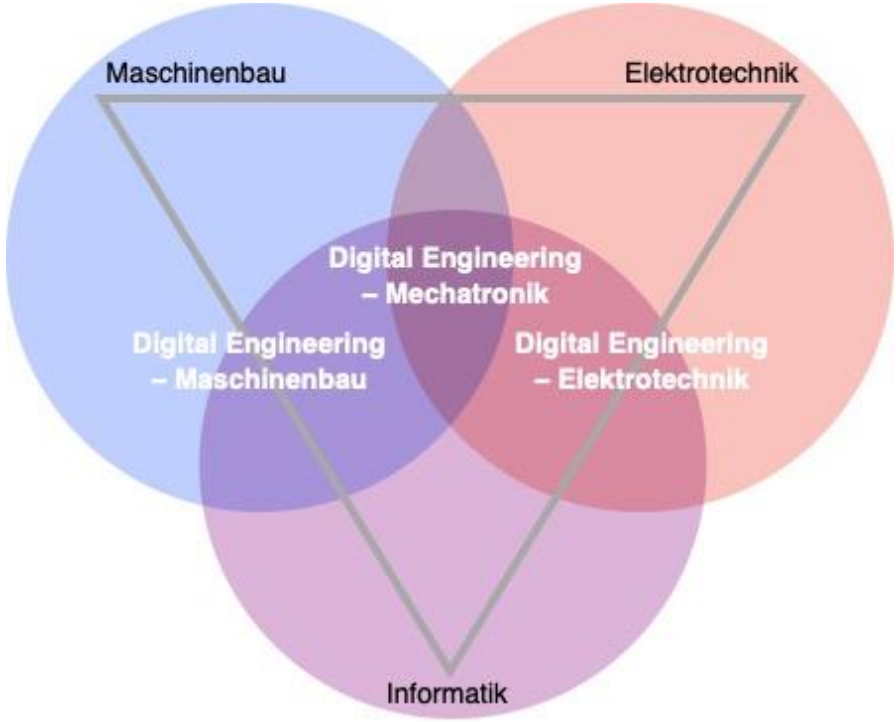
- Digital Engineering – Mechatronik
- Digital Engineering – Maschinenbau
- Digital Engineering – Elektrotechnik

Gemeinsame Mathematik

- Höhere Mathematik I und II
- Diskrete Mathematik
- Numerische Methoden bzw. Höhere Mathematik III

Gemeinsame Informatik

- Digitaltechnik
- Algorithmen und Datenstrukturen
- Objektorientierte und funktionale Programmierung
- Programmierpraktikum
- Machine Learning



Digital Engineering

Warum?

- Bedeutung von **Informatik und KI im Ingenieurwesen** nimmt stetig zu
- Interdisziplinarität: Brückenschlag zwischen **analoger** und **digitaler Welt**
- Maschinenbau → Mechatronik → Digital Engineering

Für Wen?

- Technologiebegeistert
- Digital Natives – Zukunftsorientiert
- Logisches und strukturiertes Denken
- Mathematisches und technisches Verständnis
- Kreative Lösungsfindung für technische Herausforderungen

Nutzen

- Exzellente **Zukunftsperspektiven** und Jobchancen in einem flexiblen Arbeitsumfeld
- Entwicklung der **Innovationen** von Morgen
- Ressourcenschonung durch **Industrie 4.0**
- Nachhaltige Produktion: Mit Industrie 4.0 die **ökologische Transformation** aktiv gestalten
- Fusion von **Software** und **maschinellem Lernen** mit Mechanik und Elektronik
- Effizienzsteigerung und **Ressourcenschonung** durch
  - Digitalisierung
  - Produktivitätserhöhung durch Automatisierung
  - Prozessoptimierung
  - Produktoptimierung
  - Erhöhung der Zuverlässigkeit

Motivation: „Why Software is Eating the World“

In seinem berühmten Essay “Why Software Is Eating The World” von 2011 skizziert Marc Andreessen, wieso sich dauerhaft keine Industrie vor der kreativen Zerstörung durch software- und technologiegetriebene Geschäftsmodelle schützen kann.

Quelle: https://digitaleneuordnung.de/blog/why-software-is-eating-the-world/

Softwarekrise

Der Ausdruck „Softwarekrise“ stammt von Edsger Dijkstra, NATO Software Engineering Conference (1968) und Turing Award Lecture (1972)

Die Ursachen wurden der überbordenden Komplexität der Hardware- und Software-Entwicklungsprozesse gesehen.
Die Krise manifestiert sich in verschiedensten Formen:

- Projekte laufen aus dem Budget
- Projekte laufen aus dem Zeitrahmen
- Software ist sehr ineffizient
- Software hat sehr niedrige Qualität
- Software kann die Anforderung nicht erfüllen
- Projekte werden nicht managebar und der Code schwierig zu warten
- Software wird nie ausgeliefert

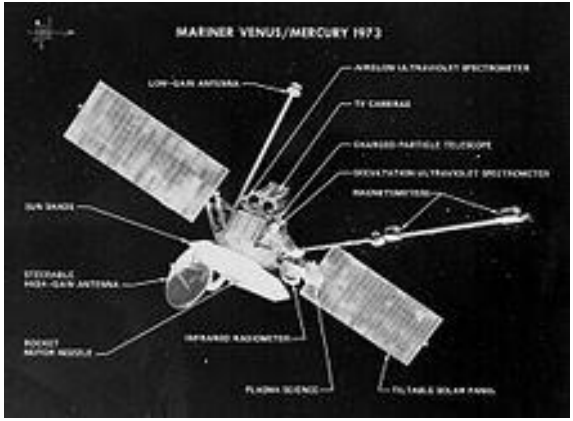
> The major cause of the software crisis is that the machines have become **several orders of magnitude more powerful**! To put it quite bluntly: as long as there were no machines, programming was no problem at all; when we had a few weak computers, programming became a mild problem, and now we have gigantic computers, programming has become an equally gigantic problem.
>
> *Edsger Dijkstra*, The Humble Programmer

Quelle: https://en.wikipedia.org/wiki/Software_crisis

Software-Desaster

Quelle: https://www.devtopics.com/20-famous-software-disasters/

1962 Ausfall der Mariner-Raumsonde $18,5 Mio.



Desaster: Die Mariner-1-Rakete mit einer Raumsonde auf dem Weg zur Venus wurde kurz nach dem Start von ihrer vorgesehenen Flugbahn abgelenkt. Mission Control zerstörte die Rakete 293 Sekunden nach dem Start.

Ursache: Ein Programmierer hat eine handgeschriebene Formel falsch in Computercode übertragen und dabei ein einzelnes „ ¯ “ übersehen. Ohne die durch den Überstrich angezeigte Glättungsfunktion behandelte die Software normale Geschwindigkeits-schwankungen als schwerwiegend, was zu fehlerhaften Korrekturen führte, die die Rakete vom Kurs abbrachen.

1978 Hartford Kolosseum-Kollapse $70 Mio.



Desaster: Nur wenige Stunden nachdem Tausende Fans das Hartford Coliseum verlassen hatten, stürzte das Stahlgitterdach unter der Last des nassen Schnees ein.

Ursache: Der Programmierer der CAD-Software, mit der das Kolosseum entworfen wurde, ging fälschlicherweise davon aus, dass die Stahldachstützen nur einer reinen Kompression ausgesetzt wären. Doch als eine der Stützen unerwartet durch den Schnee einknickte, löste dies eine Kettenreaktion aus, die die anderen Dachteile wie Dominosteine zum Einsturz brachte.

Software-Desaster

1985 **Medizintechnik tötet** **Drei Tote, drei Verletzte**

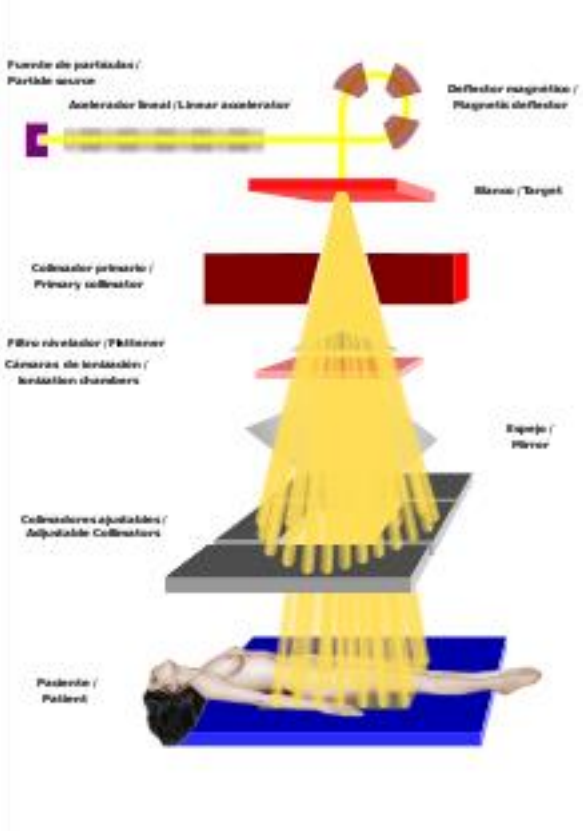
Desaster: Kanadas Strahlentherapiegerät Therac-25 funktionierte nicht richtig und gab den Patienten tödliche Strahlendosen ab.

Ursache: Aufgrund eines subtilen Fehlers namens Race Condition konnte ein Techniker Therac-25 versehentlich so konfigurieren, dass der Elektronenstrahl im Hochleistungsmodus ohne ordnungsgemäße Patientenabschirmung abgefeuert wurde.

1987 **Wall Street Crash** **$500000 Mio.**

Desaster: Am „Schwarzen Montag“ (19. Oktober 1987) stürzte der Dow Jones Industrial Average um 508 Punkte ab und verlor 22,6 % seines Gesamtwertes. Der S&P 500 fiel um 20,4 %. Das war der größte Verlust, den die Wall Street jemals an einem einzigen Tag erlitten hat.

Ursache: Ein langer Bullenmarkt wurde durch eine Flut von Untersuchungen der SEC zu Insidergeschäften und durch andere Marktkräfte gestoppt. Als Anleger in einem Massenexodus aus Aktien flohen, erzeugten Computerhandelsprogramme eine Flut von Verkaufsaufträgen, die den Markt überwältigten, Systeme zum Absturz brachten und die Anleger faktisch blind machten.



Software-Desaster

1990 AT&T-Telefonleitungen Ausfall $75 Mio.

Desaster: Bei einer einzelnen Vermittlungsstelle in einer der 114 Vermittlungszentralen von AT&T kam es zu einem geringfügigen mechanischen Problem, wodurch die Zentrale abgeschaltet wurde. Als die Zentrale wieder hochkam, sendete sie eine Nachricht an andere Vermittlungsstellen, was wiederum dazu führte, dass diese heruntergefahren wurden und das gesamte AT&T-Netzwerk neun Stunden lang lahmlegte.

Ursache: Eine einzelne Zeile fehlerhaften Codes in einem komplexen Software-Upgrade, das zur Beschleunigung von Anrufen implementiert wurde, verursachte einen Welleneffekt, der das Netzwerk lahmlegte.

1991 Patriot-Raketen-Abwehr versagt 28 Tote, 100 Verletzte

Desaster: Während des ersten Golfkrieges gelang es einem amerikanischen Patriot-Raketensystem in Saudi-Arabien nicht, eine ankommende irakische Scud-Rakete abzufangen. Die Rakete zerstörte eine Kaserne der amerikanischen Armee.

Ursache: Durch einen Software-Rundungsfehler wurde die Zeit falsch berechnet, was dazu führte, dass das Patriot-System die ankommende Scud-Rakete ignorierte.



Software-Desaster

1993 Intel Pentium Divisions-Fehler $475 Mio.



Desaster: Intels hochgelobter Pentium-Chip machte gelegentlich Fehler bei der Division von Gleitkommazahlen innerhalb eines bestimmten Bereichs. Da schätzungsweise 5 Millionen defekte Chips im Umlauf waren, bot Intel an, Pentium-Chips nur Verbrauchern zu ersetzen, die nachweisen konnten, dass sie eine hohe Genauigkeit benötigten. Schließlich ersetzte Intel die Chips für jeden, der sich beschwerte.

Ursache: Der Dividierer in der Pentium-Gleitkommaeinheit hatte eine fehlerhafte Divisionstabelle, in der etwa fünf von tausend Einträgen fehlten, was zu diesen Rundungsfehlern führte.

1996 Ariane-Rakete explodiert $500 Mio.



Desaster: Ariane 5, Europas neueste unbemannte Rakete, wurde Sekunden nach dem Start bei ihrem Jungfernflug absichtlich zerstört. Ebenfalls zerstört wurde die Ladung von vier wissenschaftlichen Satelliten, die untersuchen sollten, wie das Erdmagnetfeld mit den Sonnenwinden interagiert.

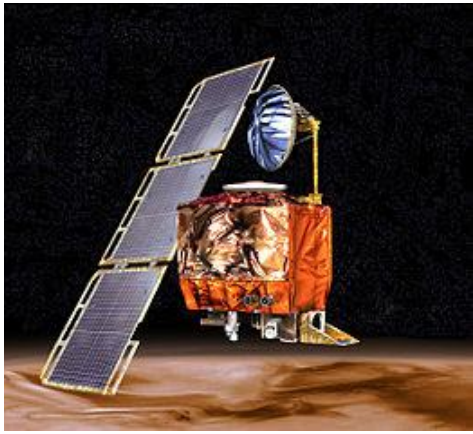
Ursache: Die Abschaltung erfolgte, als der Leitcomputer versuchte, die seitliche Raketengeschwindigkeit vom 64-Bit-Format in ein 16-Bit-Format umzuwandeln. Die Zahl war zu groß und es kam zu einem Überlauffehler. Als das Leitsystem abgeschaltet wurde, ging die Kontrolle an eine identische redundante Einheit über, die ebenfalls ausfiel, weil sie denselben Algorithmus ausführte.

Software-Desaster

1998 Mars-Orbiter crasht $125 Mio.

Desaster: Nach einer 286-tägigen Reise von der Erde zündete der Mars Climate Orbiter seine Triebwerke, um in die Umlaufbahn um den Mars zu gelangen. Die Triebwerke zündeten, aber das Raumschiff fiel zu weit in die Atmosphäre des Planeten, was wahrscheinlich zum Absturz auf dem Mars führte.

Ursache: Die Software, die die Orbiter-Triebwerke steuerte, verwendete imperiale Einheiten (Pfund Kraft) anstelle der von der NASA angegebenen metrischen Einheiten (Newton).



1999 Britische Pässe ins Nirgendwo $13 Mio.

Desaster: Die britische Passbehörde führte ein neues Siemens-Computersystem ein, das einer halben Million britischer Bürger nicht rechtzeitig Pässe ausstellte. Die Agentur musste Entschädigungen in Millionenhöhe, Überstunden des Personals und Regenschirme für Menschen zahlen, die im Regen für Pässe anstehen.

Ursache: Die Passbehörde hat ihr neues Computersystem eingeführt, ohne es ausreichend zu testen oder ihr Personal zu schulen. Gleichzeitig verlangte eine Gesetzesänderung, dass alle Kinder unter 16 Jahren, die ins Ausland reisen, einen Reisepass benötigen, was zu einem enormen Anstieg der Reisepassnachfrage führte und das fehlerhafte neue Computersystem überforderte.



Software-Desaster

1999 **Jahr 2000** **$500000 Mio.**



Desaster: Die Katastrophe des einen ist das Schicksal des anderen, wie der berüchtigte Jahr-2000-Bug zeigt. Unternehmen gaben Milliarden für Programmierer aus, um einen Fehler in veralteter Software zu beheben. Obwohl es zu keinen nennenswerten Computerausfällen kam, hatte die Vorbereitung auf den Jahr-2000-Fehler erhebliche Kosten- und Zeitauswirkungen auf alle Branchen, die Computertechnologie nutzen.

Ursache: Um Speicherplatz auf dem Computer zu sparen, speicherte ältere Software das Jahr für Datumsangaben oft als zweistellige Zahlen, beispielsweise „99“ für 1999. Die Software interpretierte „00“ auch als Bedeutung für 1900 und nicht für 2000, also als das Jahr 2000 kam, würden Fehler entstehen.

2000 **Krebsbehandlung zum Sterben** **8 Tote, 20 Verletzte**



Desaster: Die Strahlentherapie-Software von Multidata Systems International hat die richtige Dosierung falsch berechnet und Patienten schädlichen und in einigen Fällen tödlichen Strahlungsmengen ausgesetzt. Die Ärzte, die gesetzlich verpflichtet waren, die Berechnungen der Software noch einmal zu überprüfen, wurden wegen Mordes angeklagt.

Ursache: Die Software berechnete die Strahlungsdosis basierend auf der Reihenfolge, in der die Daten eingegeben wurden, und lieferte manchmal die doppelte Strahlungsdosis.

Software-Desaster

Moral

- Software ist für moderne Gesellschaften essential
- Software wirkt extremst nichtlinear: Winzige Änderungen könnten gigantische Auswirkungen haben
- Kleine Software-Fehlerchen können extrem große Schäden verursachen

Konsequenzen

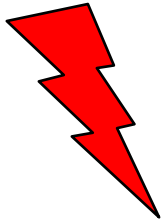
- Größerer Fokus auf Software Engineering
- Mehr Aufwand in hohe Software-Qualität

Software-Qualität

- Das A und O ist ein **guter Entwurf**
- Im industriellen Kontext muss Software- und Domian-Expertise (gleichzeitig) einfließen

Software testen

- Automatisierung ist wichtig, aber
- Testen ist niemals vollständig
- Testen kann nur Fehlersymptome, nicht Fehlerursachen finden



Klassische **Qualitätsregelkreise** funktionieren nur sehr bedingt

Fehlerkorrektur durch Regler (Umkodierung) ist extrem aufwändig

Fehlerfeststellung durch Testen ist extrem unzuverlässig

Presse

Kritische Schwachstelle in hunderten Industrieanlagen Quelle: heise.de

- heise Security hat etliche deutsche Industrieanlagen entdeckt, die leichtsinnig mit dem Internet verbunden sind. Doch damit nicht genug: Durch eine Schwachstelle kann quasi jeder die Kontrolle über Heizkraftwerke, Rechenzentren oder Brauereien übernehmen.
- Hunderte, öffentlich erreichbare IP-Adressen von virtuellen Schaltzentralen, die ohne effektive Authentifizierung sperrangelweit offen standen.
- Fernwärmekraftwerk: Per Mausklick hätten wir die Wärmeversorgung lahmlegen können, was zu einem Ausfall von mehreren Stunden geführt hätte.

Airbus A400M (bei Sevilla): Softwareprobleme sollen Airbus-Absturz verursacht haben Quelle: zeit.de

- Softwareproblem in der Triebwerkssteuerung
- Piloten hätten nichts unternehmen können, um die Triebwerke wieder zu starten. Sie hätten zwar versucht, das Flugzeug zurück zum Flugplatz zu steuern, konnten es aber nicht mehr kontrollieren.

Boeing 737 MAX 8: Abstürze Quelle: entwickler.de

- Indonesien (Lion Air Flug 610)
- Äthiopien (Ethiopian Airlines Flug 302)
- Kombination aus ausgefallenem Sensor, Fehler in Steuerungs-Software und Fehler in Cockpit-Darstellung/Dokumentation

Presse

Wie Alltags-Software Menschenleben gefährden kann Quelle: manager-magazin.de

- Zahlreiche Rückrufaktionen aufgrund von Software-Problemen
- Komplexität und Time-to-Market-Druck als Ursachen
- Mit präventiver Qualitätssicherung Millionenschäden durch Software vermeiden

Rückrufe wegen Software-Fehlern
Ford: Airbags, 1,4 Mio. Fzg., 2014
Audi: Airbags, 850.000 Fzg., 2014
Volvo: Feststellbremse, 6 Modelle, 2015
Ford: Motorsteuerung, 433.000 Fzg., 2015
Toyota: Überhitzung Hybrid, 630.000 Fzg., 2015

McKinsey-Analyse: Das große Software-Desaster bei Volkswagen Quelle: cio.de

- Modellverschiebungen und Mehrkosten gravierender seien als bekannt
- Über die Laufzeit bis 2039 seien sogar Mehrausgaben von etwa neun Milliarden Euro zu erwarten

Software: Unser Grab als PKW-Besitzer der Zukunft? Quelle: borncity.com

- Ein einfaches Auto hat bereits heute Software an Bord, für die 100 Millionen Zeilen Quellcode geschrieben wurden
- High-End-Fahrzeuge mit Fahrassistenten kommen auf 150 Steuereinheiten mit 150 Millionen Codezeilen

Von gescheiterten IT-Projekten lernen Quelle: entwickler.de

- Klare Vorgaben und Ziele führen zum Erfolg
- Hauptaufgabe eines Entwicklers ist nicht das Schreiben von Code, sondern das Verstehen eines Problems

IT-Projekte
14% scheitern komplett
31% erreichen nicht (alle) ihre Ziele
43% überschreiten ihr Budget
49% liefern zu spät

Mission Statement

Robert C. Martin (über das 737 MAX 8 Desaster) in „The Clean Code Blog“:

**“Programmers must not be treated as requirement robots.
Rather, programmers must have intimate knowledge of the domain
they are programming in.”**

Je technisch anspruchsvoller die Projekte werden, desto wichtiger wird die Domain-Expertise. Wir brauchen für eine moderne Industrie Mitarbeiter/innen mit

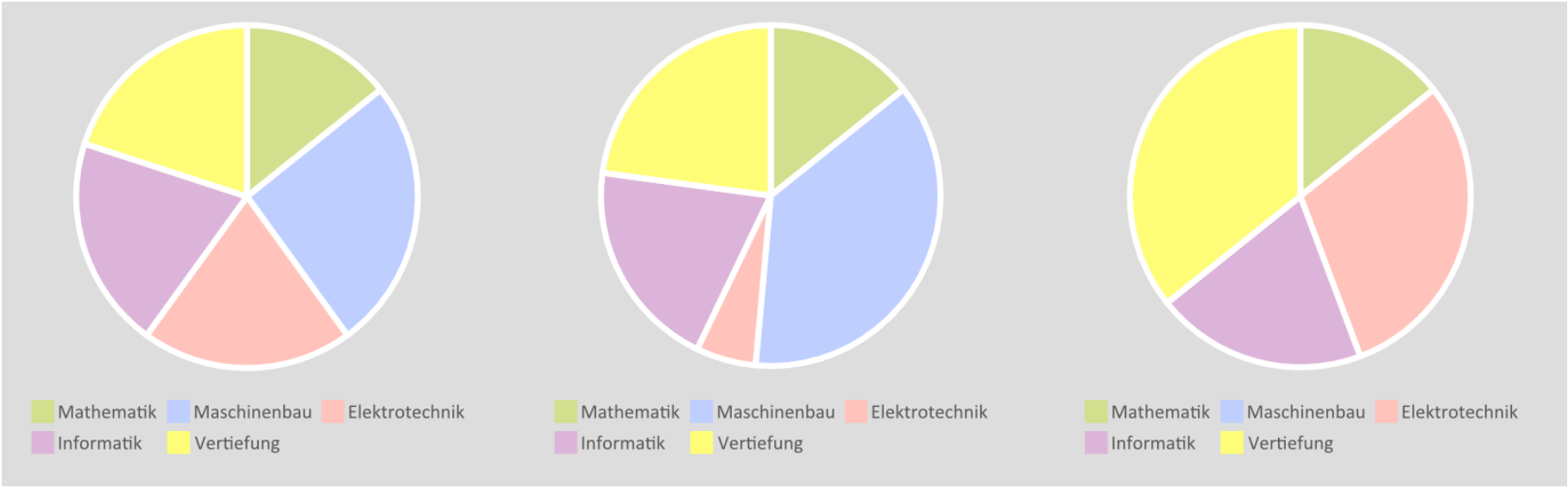
Ingenieurs- und Software-Knowhow

→ **Digital Engineering**

Bachelor Digital Engineering: 3 Studiengänge

Digital Engineering – Mechatronik

1/3 MB 1/3 ET 1/3 INF



Digital Engineering – Maschinenbau

2/3 MB 1/3 INF

Digital Engineering – Elektrotechnik

2/3 ET 1/3 INF

Vertiefungen in Digital Engineering – Mechatronik und – Elektrotechnik:

- Leistungselektronik & Antriebstechnik
- Halbleiter- & Schaltungstechnik
- Kommunikations- & Informationstechnik
- Electronic Systems
- Automatisierungstechnik
- Image Sensing & Processing
- Embedded Systems

Bachelor Digital Engineering: Module

Alle 3 Studiengänge

- Fast identische Mathematik (grün)
- Identische Informatik (lila)

Informatik

- Schwerpunkt Software-Entwicklung
- Machine Learning

Digital Engineering – Mechatronik

- Kombination aus MB und ET + viel Informatik

Digital Engineering – Maschinenbau

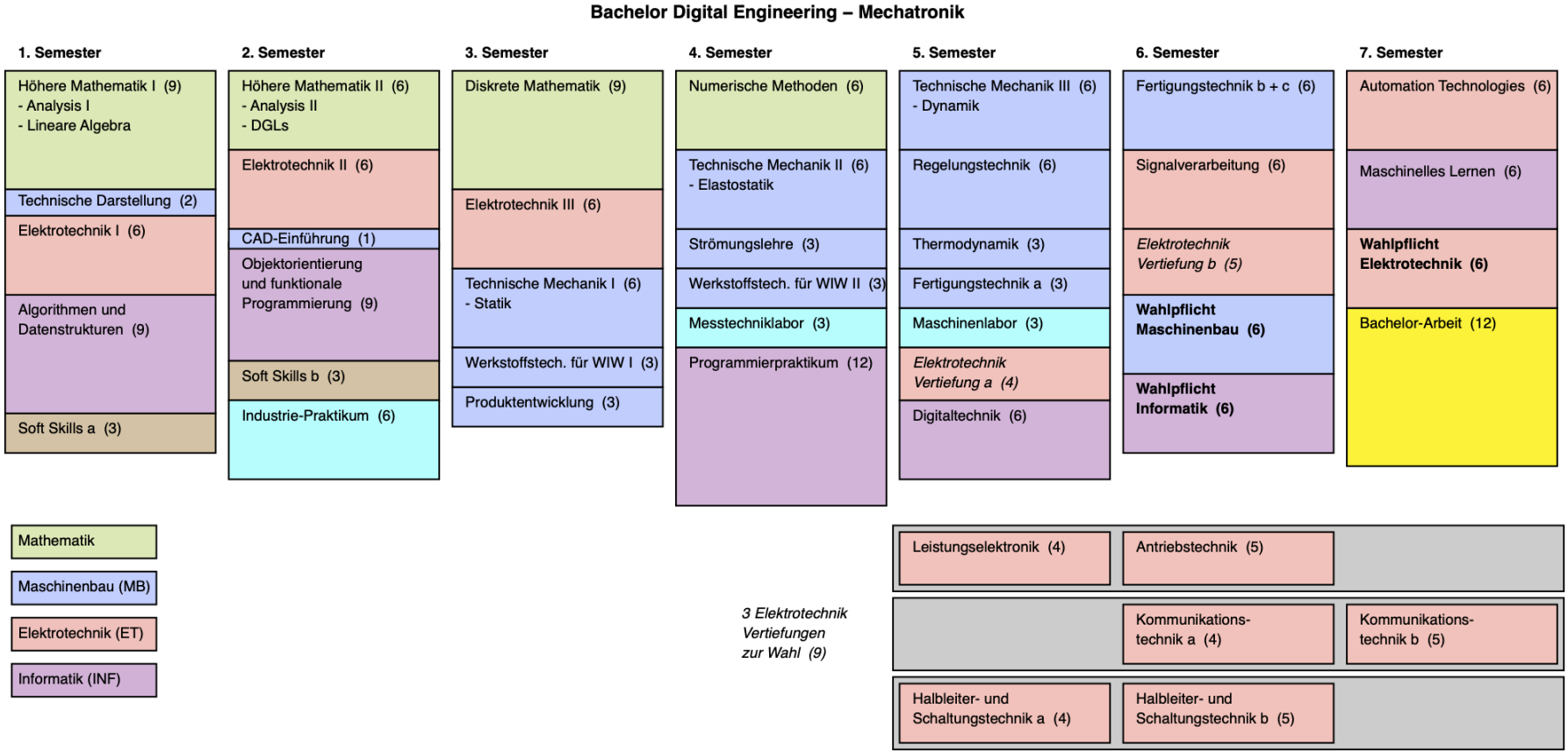
- Gekürztes Maschinenbau-Studium + viel Informatik

Digital Engineering – Elektrotechnik

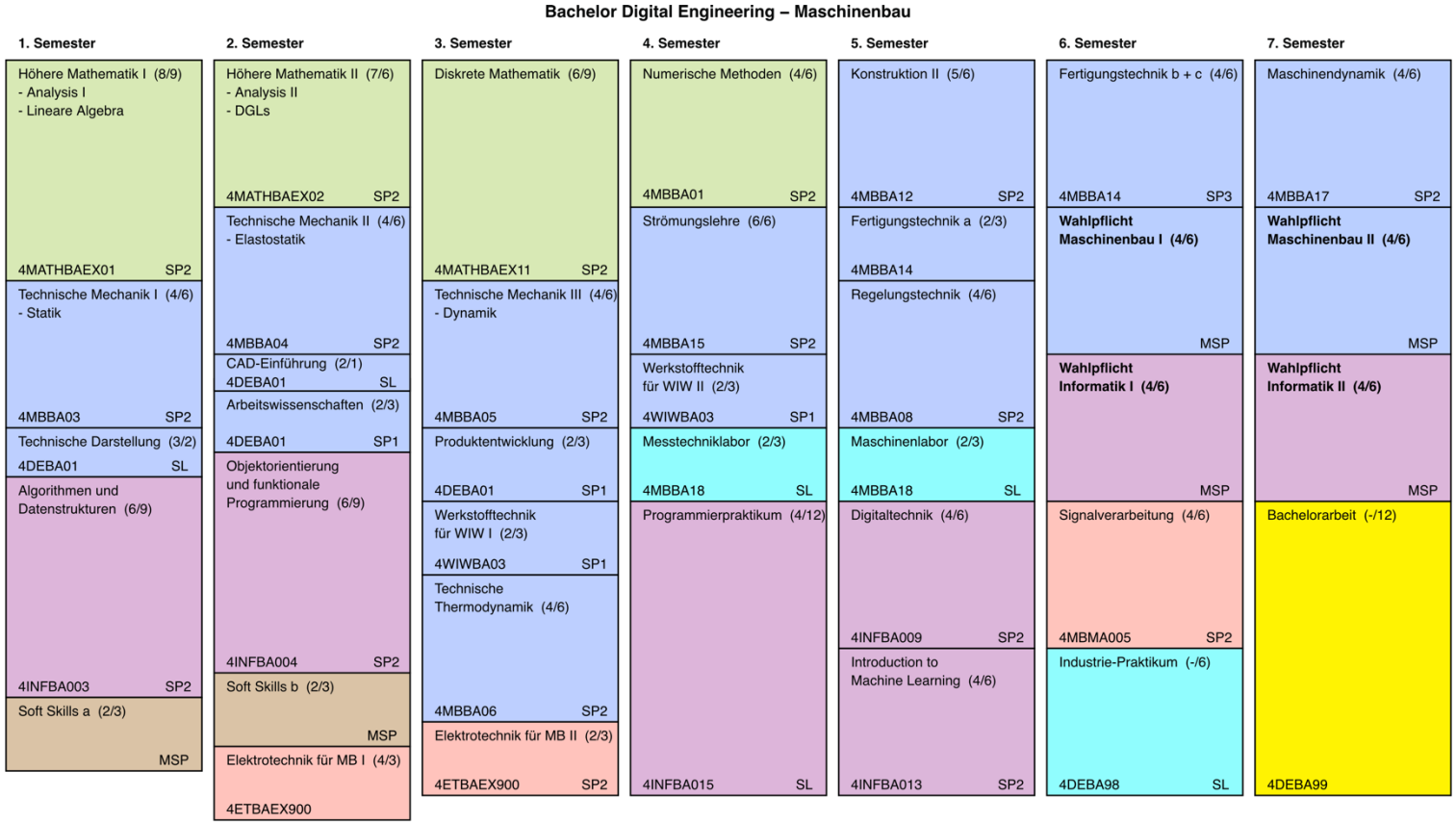
- Gekürztes Elektrotechnik-Studium + viel Informatik



Bachelor Digital Engineering – Mechatronik: Studienverlauf



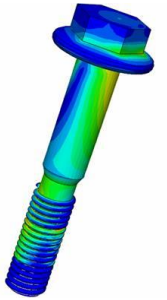
Bachelor Digital Engineering – Maschinenbau: Studienverlauf

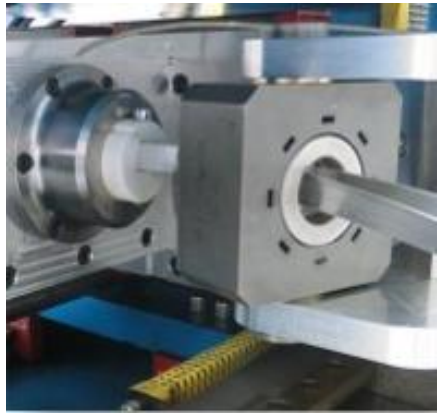


Department Maschinenbau (MB)

Konstruktion und Produktion

- Produktentwicklung
- Verbindungstechnik und Produktinnovation
- Umformtechnik
- Fertigungsautomatisierung und Montage
- Arbeitswissenschaft und Ergonomie
- International Production Engineering and Management

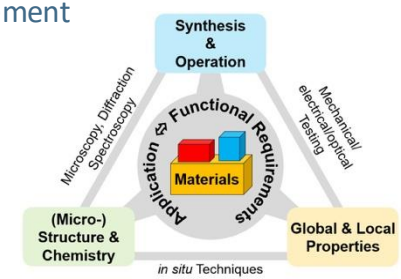


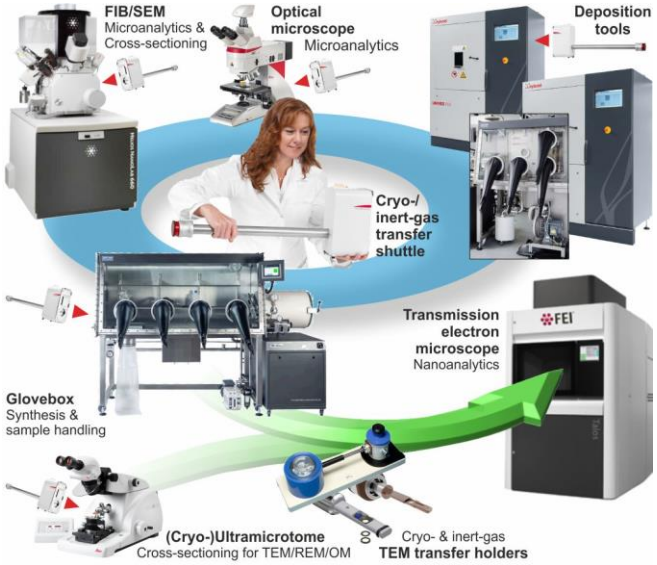


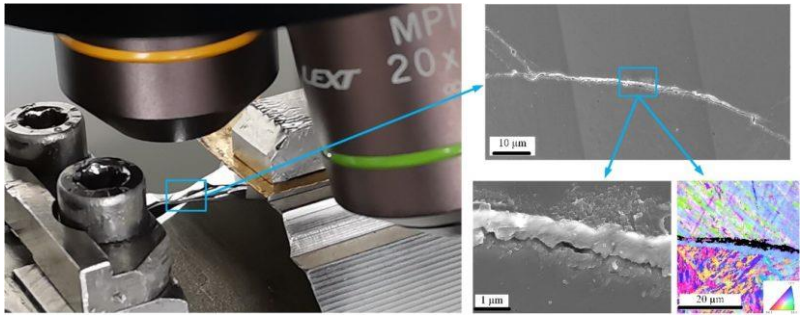


Werkstofftechnik und Fahrzeugtechnik

- Materialkunde und Werkstoffprüfung
- Oberflächen- und Werkstofftechnologie
- Mikro- und Nanoanalytik
- Werkstoffsysteme für den Fahrzeugleichtbau
- Fahrdynamik



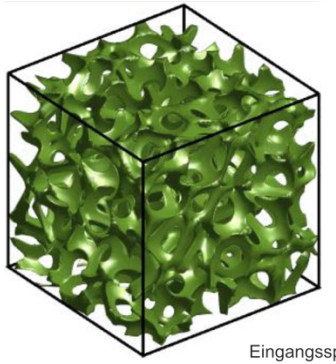


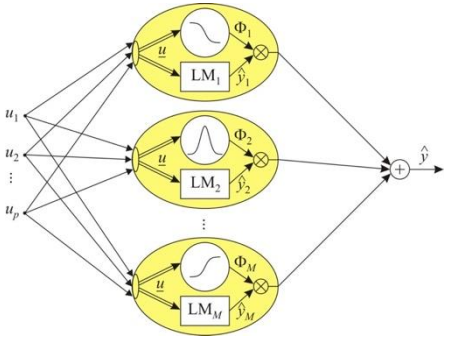


Department Maschinenbau (MB)

Mechanik und Regelungstechnik – Mechatronik

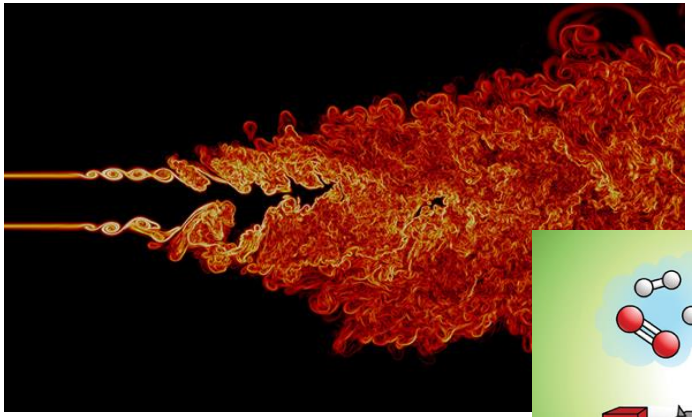
- Numerische Mechanik und FEM
- Experimentelle Mechanik und Structural Health Monitoring
- Machine Learning für dynamische Modelle

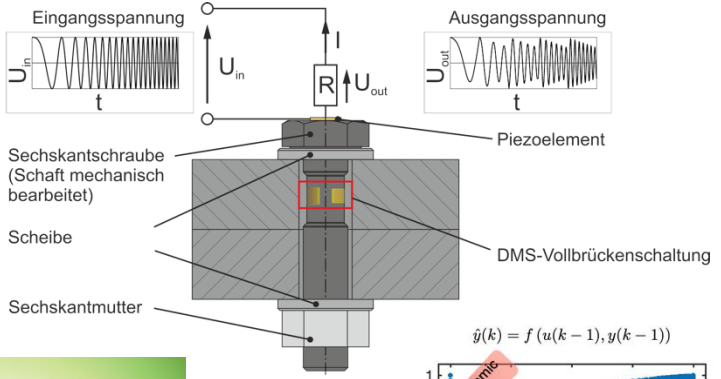


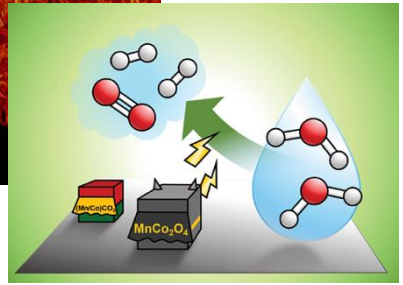


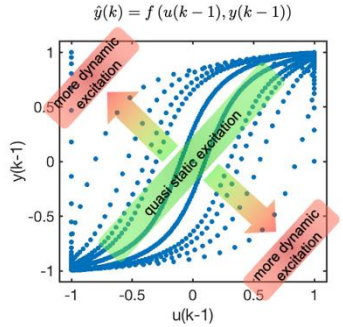
Fluid-, Thermodynamik und Energietechnik

- Numerische Strömungssimulation
- Verbrennungsanalyse
- Wasserstofferzeugung









Department MB

- Studierende:
  - 237 BA MB
  - 240 BA WIW
  - 86 MA MB
  - 91 MA WIW
- 18 Professoren

Bachelor Digital Engineering – Elektrotechnik: Studienverlauf

5 Vertiefungsrichtungen (VT)

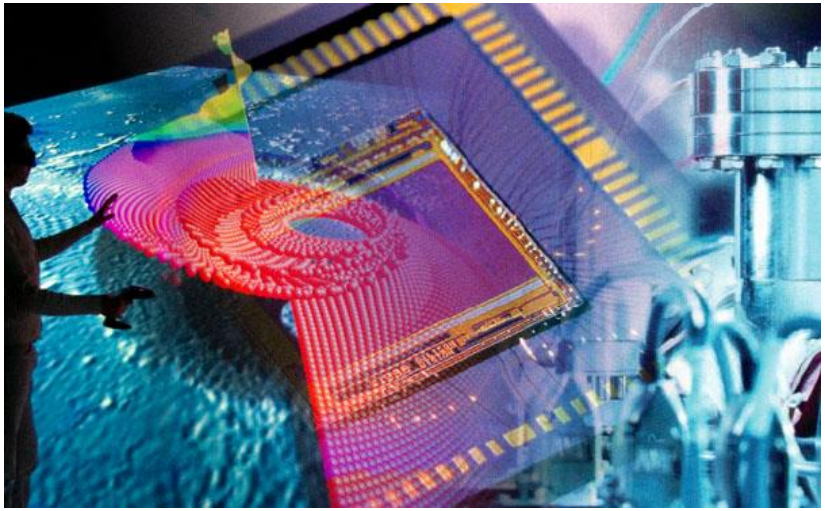
- Automatisierungstechnik
- Electronic Systems
- Embedded Systems
- Image Sensing and Processing
- Kommunikations- und Informationstechnik

Wichtige Inhalte der VTs

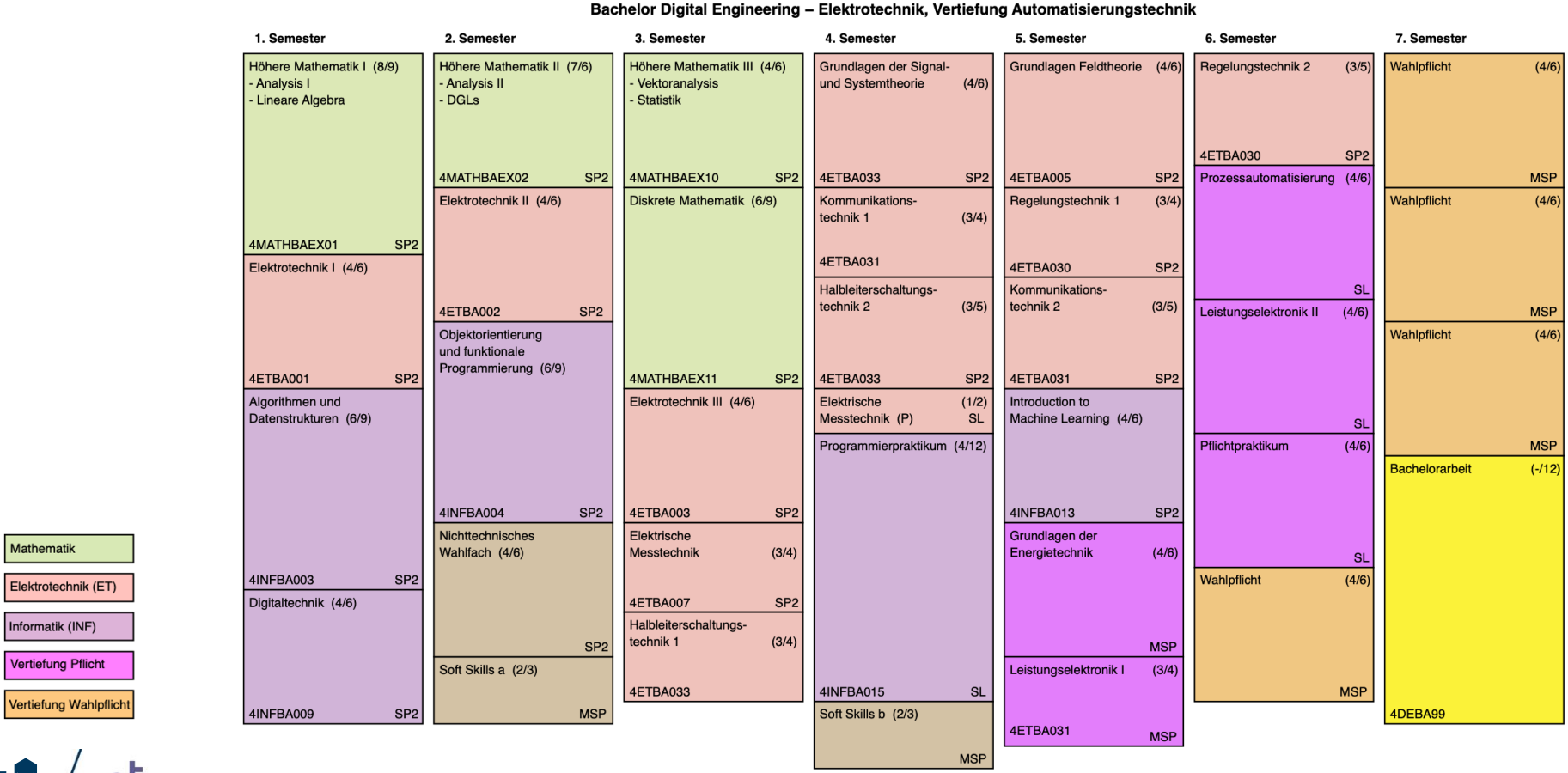
- Energietechnik & Leistungselektronik
- Semiconductors & Nanotechnology
- Rechnerarchitekturen & Betriebssysteme
- Digitale Bildverarbeitung & Visual Computing
- Signals and Systems & Hochfrequenztechnik

Details

- Umfang: 27 LP
- Wahl der VT ab 4-5 Semester
- Wird im Master weitergeführt



Bachelor Digital Engineering – Elektrotechnik (VT Automatisierungstechnik): Studienverlauf



Department Elektrotechnik und Informatik (ETI)

Eingebettete Systeme: Spezifische, miniaturisierte Rechner

- Signalverarbeitung, z.B. Ver- bzw. Entschlüsseln
- Kommunikation unter Echtzeitbedingungen

Sensoren & Kommunikation: Erfassung & Übertragung von Daten

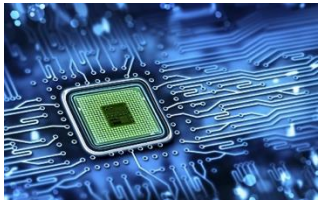
- Zustand von Systemen & Maschine (Temperatur, Füllstand, etc.)
- Erfassung komplexer Umgebungsinformation

Visual Computing: Verarbeitung von Bildinformation

- Analyse von Bildern, z.B. zur Objekterkennung
- Erzeugung von Bildern, z.B. zur virtuellen Überlagerung

Automatisierung: Systeme, die eigenständig Aufgaben lösen

- Überwachung, Steuerung und Regelung von Prozessen
- Anpassung an veränderte Bedingungen und Menschen









Department ETI

- Studierende:
  - 199 BA ET
  - 432 BA INF
  - 104 MA ET
  - 179 MA INF
  - 245 MA Mechatronics
- 22 Professoren

Department Elektrotechnik und Informatik (ETI)

Anwendungen (Beispiele)

- Moderne, autonome Mobilität



- Virtuelle Medien & Kommunikation



- Nachhaltige Energieversorgung



- Digitale Medizin



Digital Engineering: Prüfungsamt/ausschuss

Prüfungsamt des Departments *Maschinenbau*

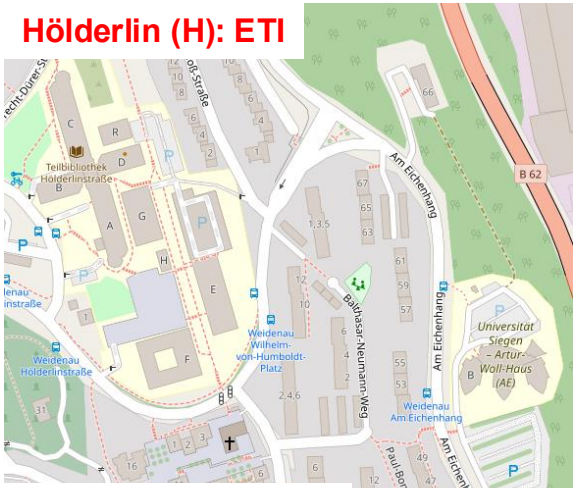
- PB-A242 / A242-1
- Montag - Donnerstag 9:30 - 11:00
- Email: pruefungsamt@maschinenbau.uni-siegen.de

Prüfungsausschuss

- Vorsitzender: Prof. Dr.-Ing. Axel von Hehl
- Ansprechpartner: Prof. Dr.-Ing. Oliver Nelles
- Sprechstunde: Mi., 10:30– 11:30
- PB-A249
- Email: oliver.nelles@uni-siegen.de

Bachelor Digital Engineering: Lageplan

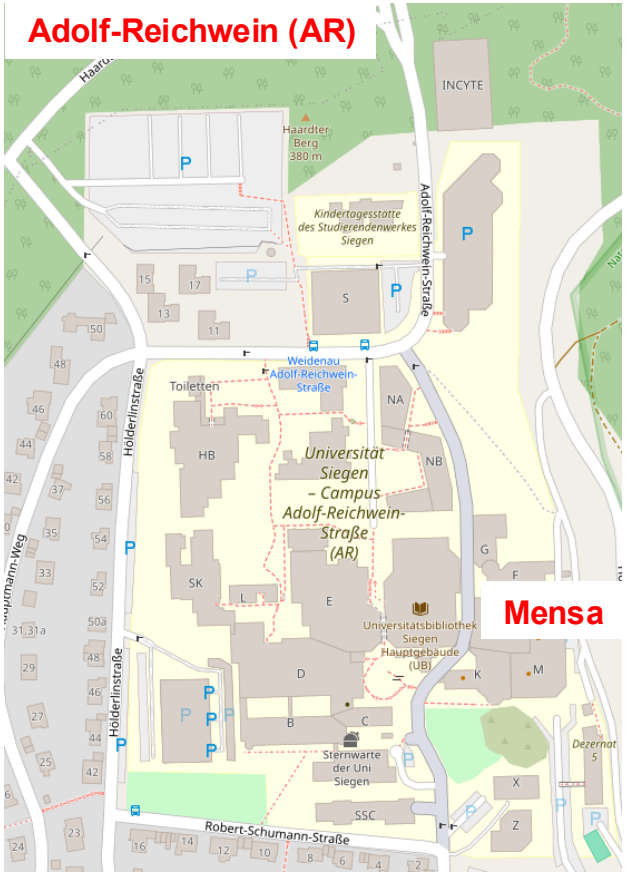
Hölderlin (H): ETI



Paul-Bonatz (PB): MB



Adolf-Reichwein (AR)



Mensa

Master Digital Engineering

7 Sem. BA + 3 Sem. MA

Nutzung des existierenden MA Mechatronics

- International (jetzt: ~ 70-100 Studierende)
- 4 Semester
- 1 Semester (30 LP) wird anerkannt,
  da im BA Digital Engineering – Mechatronik
  schon enthalten

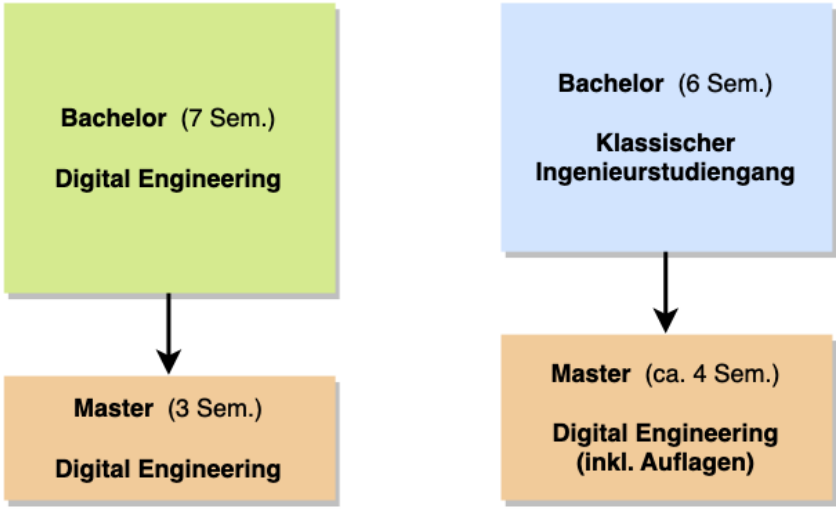


Bachelor, 7 Sem. (210) Digital Engineering – Mechatronik	Bachelor, 7 Sem. (210) Digital Engineering – Maschinenbau	Bachelor, 7 Sem. (210) Digital Engineering – Elektrotechnik
Deutsch	Deutsch	Deutsch
Master, 3 Sem. (90) (Digital Engineering) – Mechatronics	**Master, 3 Sem. (90) Digital Engineering – Maschinenbau**	**Master, 3 Sem. (90) Digital Engineering – Elektrotechnik**
Existiert bereits: MA Mechatronics, 4 Sem. (120) 1. Anerkennung von 30 LP: 12 LP Alignment 6 LP Signal Processing 6 LP Automation Technologies 6 LP Software Engineering 2. Neue FPO für 3 Sem. (90)	Wahlpflicht (54) aus einem Katalog mit Modulen aus - MA Maschinenbau - MA Informatik Fachlabore Master-Arbeit	Pflicht (24) Wahlpflicht (30) Vertiefungsrichtungen: - Automatisierungstechnik - Communication - Electronic Systems - Image Sensing & Processing - Embedded Systems Projektarbeit Master-Arbeit
Englisch	Deutsch	Englisch / Deutsch

Master Digital Engineering

Klassischer Bachelor mit 6 Semestern – Und nun?

- Mit zusätzlichen Auflagen im Umfang von ca. 30 CPs (ca. 1 Sem.) kann aus dem 3-semestrigen Master Digital Engineering ein 4-semestriger Master (inkl. Auflagen) entstehen
- Die Auflagen umfassen typischerweise die fehlenden Informatik-Module
- Die Auflagen werden vom Prüfungsausschuss individuell zusammengestellt



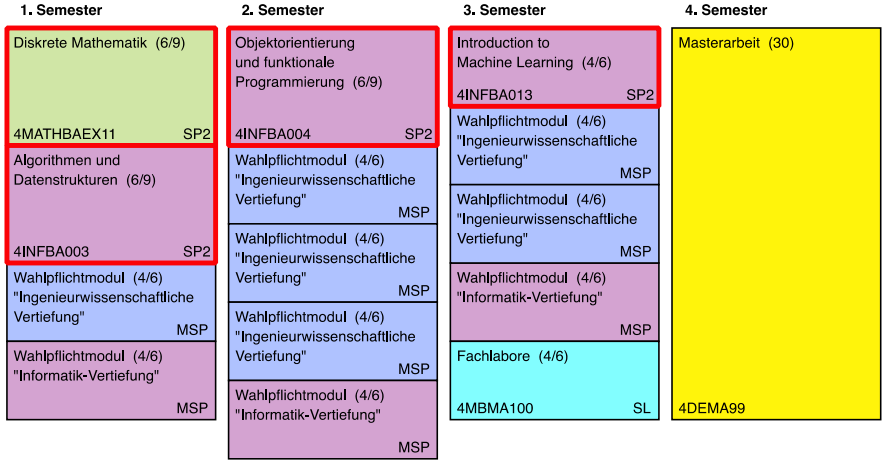
Master Digital Engineering

Speziell für Bachelor-Absolventen der Universität Siegen

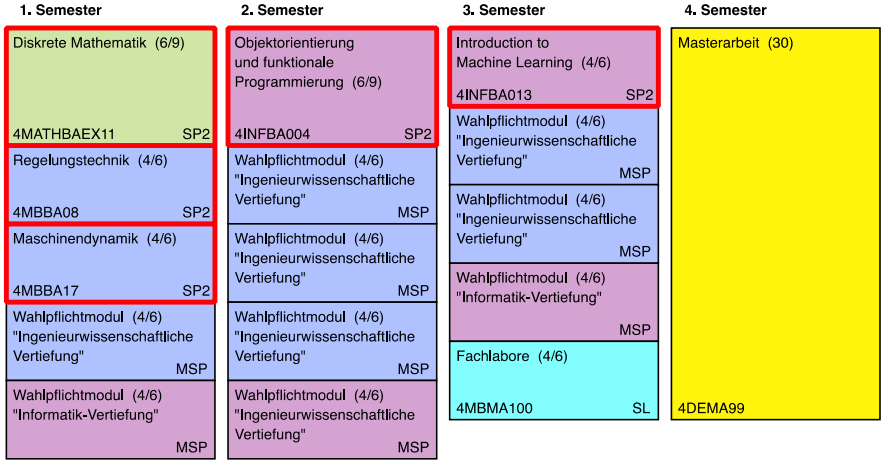
- Standardisierte Auflagen
- Bachelor Maschinenbau: 33 LP
- Bachelor Wirtschaftsingenieurwesen: 36 LP

Auflagen (rot umrandet)

bei Bachelor Maschinenbau (Uni Siegen)



bei Bachelor Wirtschaftsingenieurwesen (Uni Siegen)



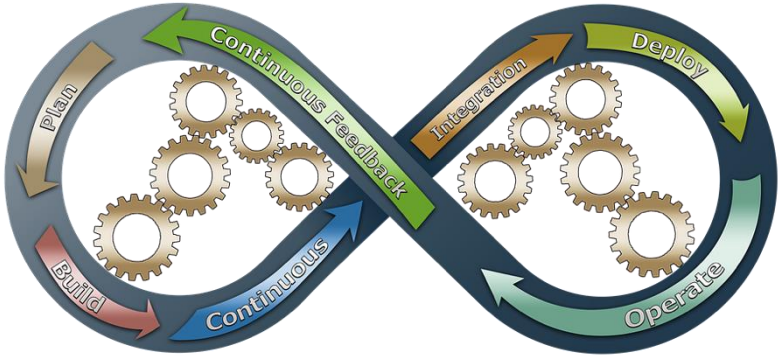
Relevanz der neuen Studiengänge

Denkweise beider Disziplinen vereinen

- Mathematik Ingenieurwesen: Orientierung an physikalischen (auch dynamischen) Prozessen mit Differential-, Integralrechnung und Differentialgleichungen
- Mathematik Informatik: Diskrete Mathematik mit Kombinatorik, Zahlen- und Graphentheorie, Kryptographie

Problem heute

- Software-Anteil an Wertschöpfung steigt stetig und schnell
- IngenieurInnen: zu wenig Software-Knowhow
- InformatikerInnen: zu wenig physikalisch/industrieller Bezug



Inhalte der Studiengänge „Digital Engineering“

- Grundlagen der Mathematik und Numerik für Ingenieure und Informatiker
- Verständnis für Prozesse, Signale und Daten
- Knowhow auf den Gebieten: Algorithmen, Datenstrukturen und Software-Entwicklung
- Mögliche Vertiefung in den Schwerpunkten:
  Produktionstechnik (MB), Numerische Methoden (MB), Zustandsüberwachung (MB), Mess- und Automatisierungstechnik (MB/ET), Kommunikations- und Informationstechnik (ET), Electronic Systems (ET), Image Sensing & Processing (INF), Embedded Systems (INF)

Mögliche Berufsfelder

Kernkompetenz Modelle

- Leistungsfähige Ansätze benötigen gute Modelle (*digitale Zwillinge*) zur Simulation, Steuerung/Regelung, Diagnose, Adaption, …
- White-Box (aus Physik)

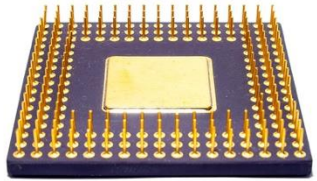


  - Mechanik, Elektrotechnik, Thermo- und Fluiddynamik, …
- Black-Box (datengetrieben)



  - Signalverarbeitung, Statistik, Machine Learning, …





Kernkompetenz Software

- Informatik-Module: Algorithmen & Datenstrukturen, Objektorientierte Programmierung, Programmierpraktikum, …



Arbeitsfelder (Beispiele)

- Entwicklung und Software-Implementierung neuer Funktionen in Steuergeräten, Regelungen, Überwachungs- und Diagnosesystemen, für virtuelle Sensoren, …
- Optimierung von Prozessautomatisierungen
- Robuste KI-Anwendungen in der Industrie
- Data Mining in cyber-physischen Systemen
- Predictive Maintenance in mechatronischen Systemen

Von MotorBlog from Ca, USA - IAA 2013: BMW i3Uploaded by AVIA BavARia, CC BY 2.0, https://commons.wikimedia.org/w/index.php?curid=29302575

Universität Siegen

Naturwissenschaftlich-Technische Fakultät

Kontakt

Prof. Dr.-Ing. Oliver Nelles

Email: oliver.nelles@uni-siegen.de

Webseite: www.mb.uni-siegen.de/mrt/lehre/digitalengineering

