
EBS43 Embedded Software Development

Dokumentation im Embedded Software Process

Vladimir
Zhelezarov

Eingebettete Systeme

Definition:

Eingebettete Systeme sind Rechenmaschinen, die für den Anwender weitgehend unsichtbar in einem elektrischen Gerät „eingebettet“ sind. Ein eingebettetes System ist typisch konstruiert um eine bestimmte Aufgabe auszuführen.

Beudeutung:

- Rund 98 Prozent aller hergestellten Prozessoren sind für eingebettete Systeme
- Anwendung bei der industriellen Kontrolle und Automatisierung, Unterhaltungselektronik, in Internet der Dinge, Automobilen, Medizin uvm.

Besonderheiten:

- Zeitliche Reaktion, manchmal sogar kritisch - Echtzeitsysteme
- Laufen “unendlich”
- Interagieren mit der Umgebung und nicht mit dem Benutzer
- Enge Verzahnung von Software und Hardware
- Begrenzte Hardware-Ressourcen
- Schwer änderbar/aktualisierbar, manchmal mit ROM
- Spezifische Aufgaben; einzigartig

Techinsche Dokumentation

Definition:

Alle Dokumente, die dazu dienen ein Produkt in seinen Eigenschaften zu beschreiben.

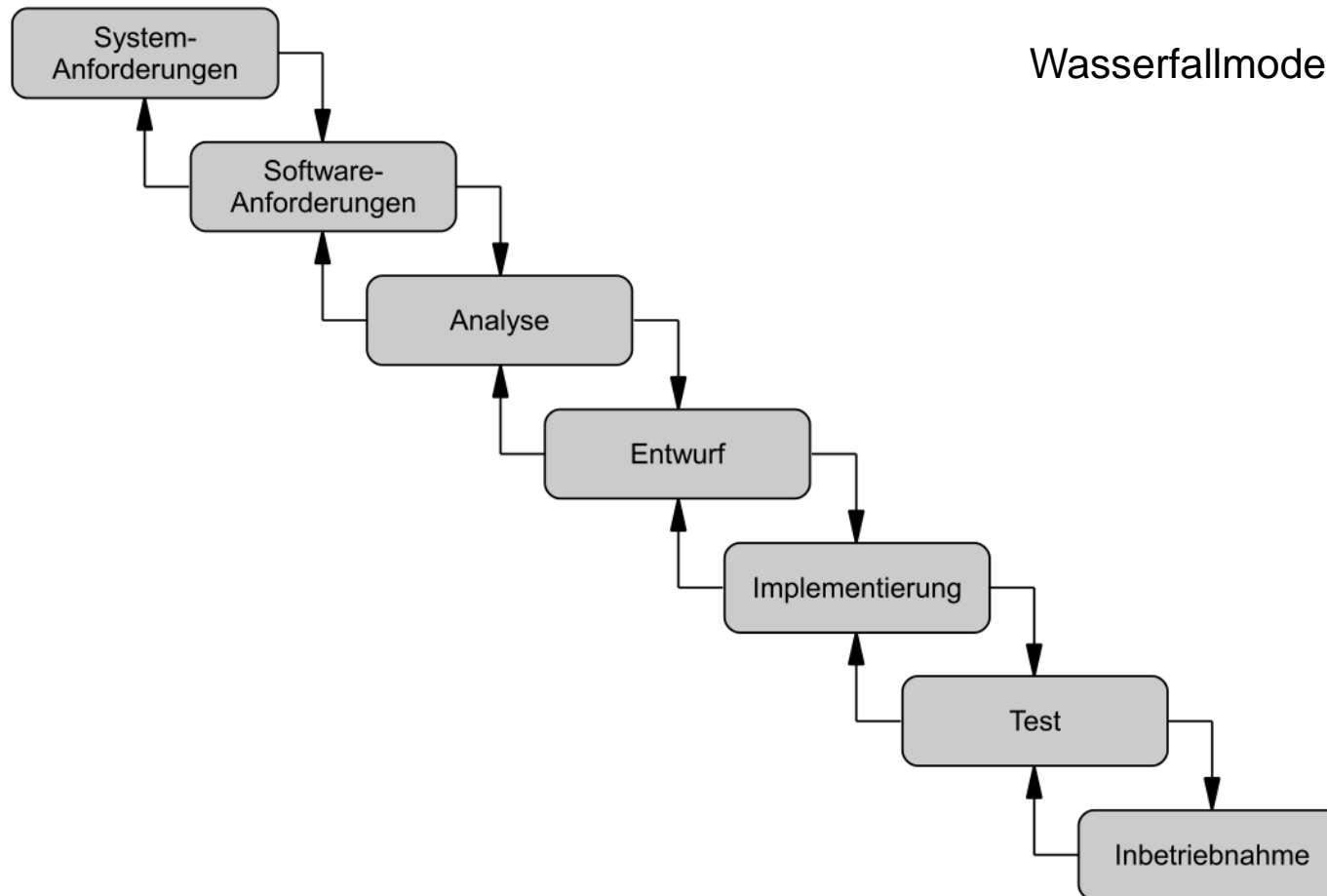
Anforderungen:

- Vollständigkeit; Einheitlichkeit, Strukturiertheit und Übersichtlichkeit; Benutzbarkeit und Anschaulichkeit; Änderbarkeit und Anpassungsfähigkeit; Widerspruchsfreiheit; Aktualität und Wirtschaftlichkeit der Erstellung
- Zusätzlich gilt die eingebettete Software als Bestandteil des Produkts oder Geräts, in dem sie enthalten ist. Daraus folgt, dass für sie vorrangig alle für das Produkt anwendbaren Richtlinien, Gesetze und Verordnungen gelten.
- Normen für Software-Dokumentation (empfehlenswert, keine Pflicht):
 - IEEE 1063, ISO/IEC 6592, ISO/IEC 9294, ISO/IEC 18019, ISO/IEC 9127, ISO/IEC 26514

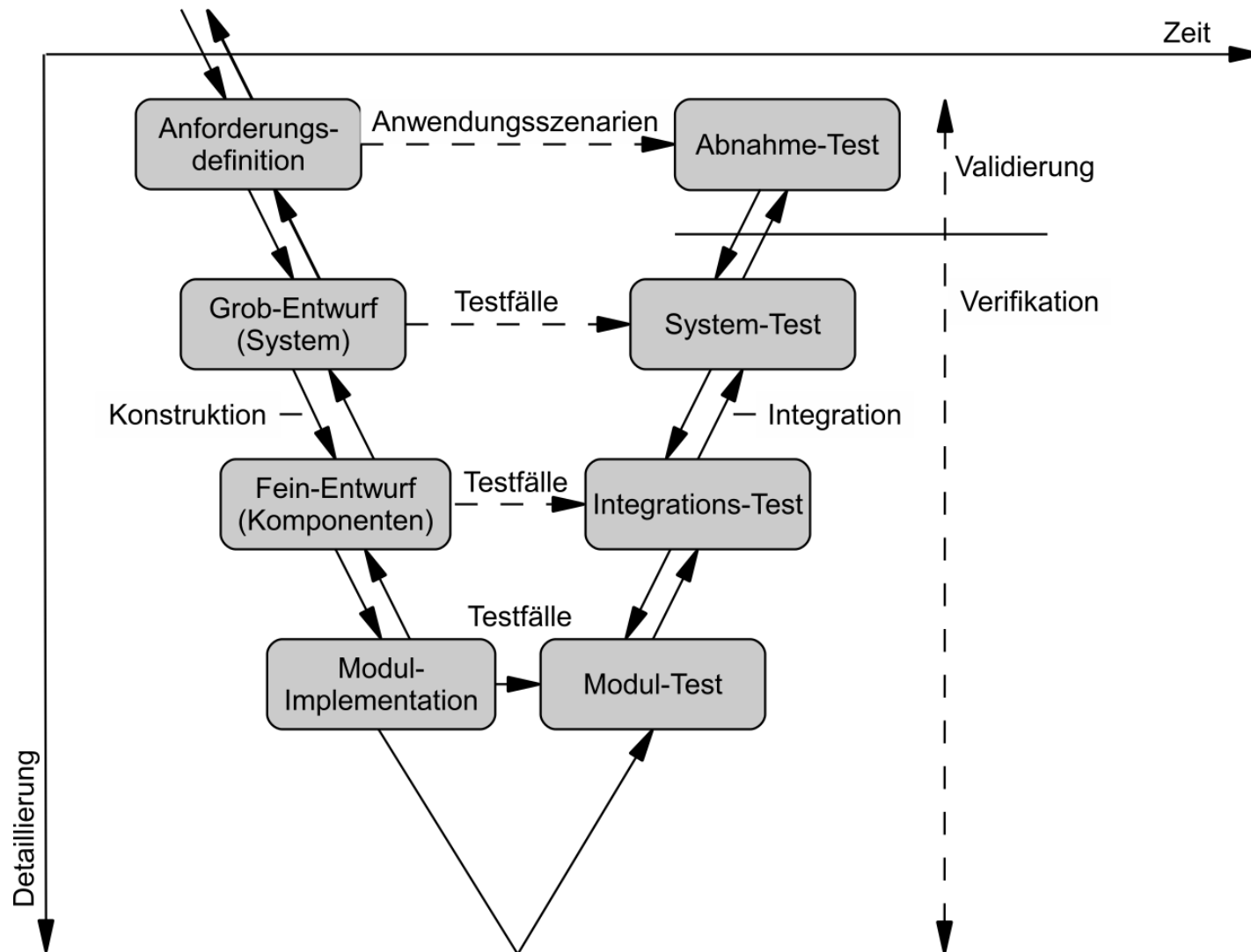
Entwicklungsprozess-Modelle

- Entwicklung vom eingebetteten System ist auch ein IT-Projekt;
- Gruppierung nach Formalität:
 - Spezifikationsorientiert: V-Modell, Wasserfall; Sehr formell; Dokumentation im Mittelpunkt;
 - Prototypenorientiert: Prototypenmodell; Weniger formell; Das Produkt steht im Mittelpunkt;
 - Agil: nicht formell; Menschen und Beziehungen haben Priorität.

Wasserfallmodell



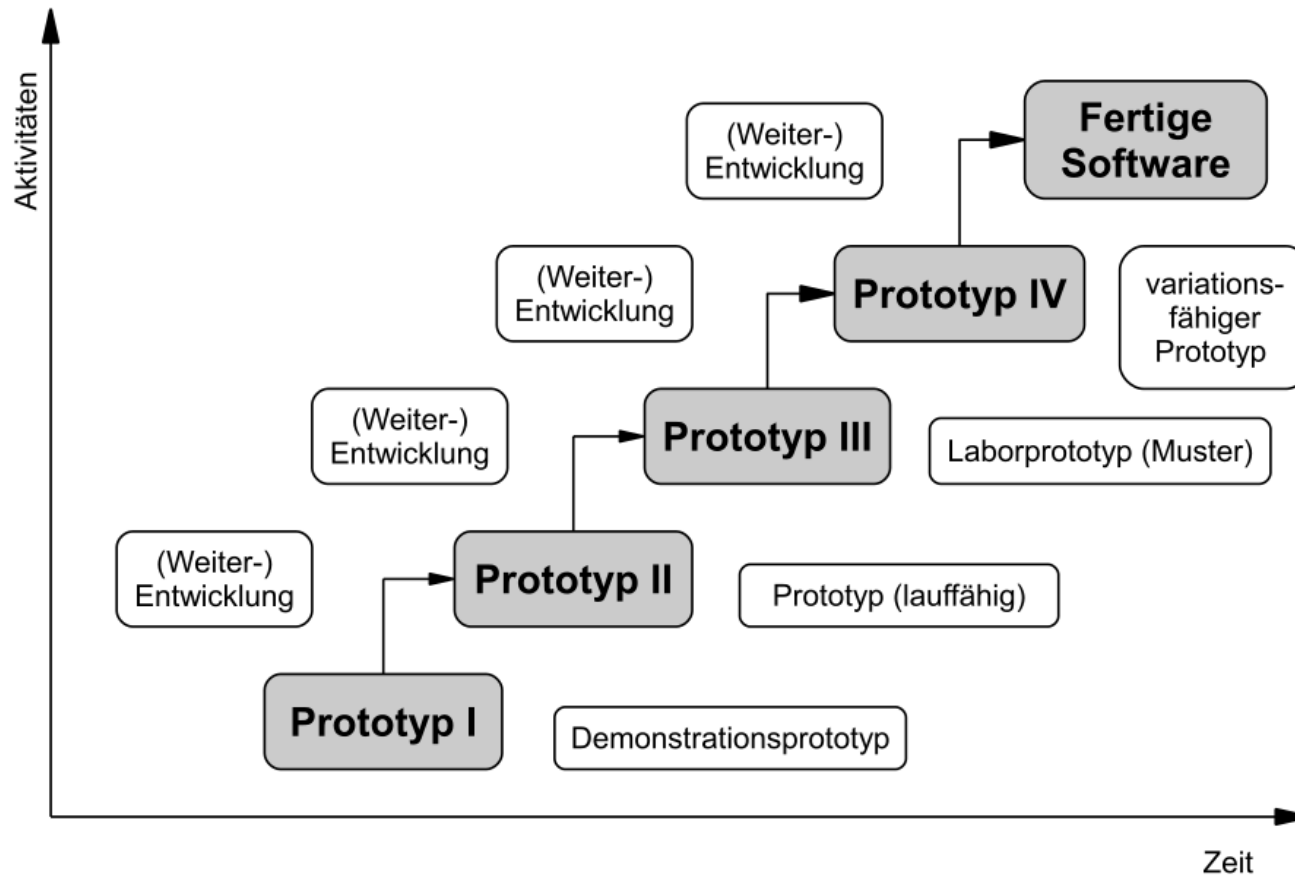
Quelle: Gessler (2014)



V-Modell

Quelle: Gessler (2014)

Prototypen-Modell



Quelle: Gessler (2014)

Agile Prinzipien

<i>Hohen Wert</i>		<i>Niedriger Wert</i>	
Individuen und Interaktionen	>	Prozesse und Werkzeuge	
Funktionierende Software	>	umfassende Dokumentation	
Zusammenarbeit mit dem Kunden	>	Vertragsverhandlung	
Reagieren auf Veränderung	>	Befolgen eines Plans	

Dokumente bei der Entwicklung von eingebetteten Systemen

- Anforderungsdefinition, in der Kundenwünsche und Vereinbarungen über dem Verhalten des System definiert sind;
- Architekturbeschreibung, die sich mit dem grundlegenden Design beschäftigt und die Richtung der Entwicklung bestimmt;
- Diagramme, die die Struktur oder das Zusammenspiel der Module erklären;
- Hardware-Beschreibungen, die nicht nur als Schaltpläne, sondern auch als Definitionen in z.B. Verilog oder VHDL angeboten sein können;
- Nicht zuletzt klare und eindeutige Kommentare im Code selbst.

Benutzung von Dokumentation in der Praxis

Objekt der Untersuchung:

- 7 populäre Open-Source Projekte für eingebettete Software
- Populär = viele Sterne in GitHub
- Open-Source = Agil

Methodik der Untersuchung:

- Vorhandensein und Art der Dokumentation. Abgrenzung zwischen Entwickler- und Benutzerdokumentation, Beschreibung der Architektur;
- Externe Dokumentation und Kommentare im Code.

Messbare Größen:

- Verhältnis Kommentare/Code
- Anzahl der Commits die Dokumentation/Kommentare zufügen oder ändern aus den letzten 20 Commits
- Begründung der Auswahl und Kritik der Methodik

Benutzung von Dokumentation in der Praxis

Vorstellung der Projekte:

- Arduino AVR Core
 - . Repository für die Firmware zur Interaktion mit Atmel/Microchip Boards für die Entwicklungsplattform Arduino
- RT-Thread
 - . Open-Source Betriebssystem für Internet der Dinge
- PX4 Autopilot Software
 - . Software für Dronen
- Asuswrt-Merlin
 - . Erweiterte Firmware für Router von Asus
- Tasmota
 - . Alternative Firmware für ESP8266 Wireless-Modulen
- Quantum
 - . Alternative Firmware für mechanische Tastaturen mit AVR oder ARM Steuerung
- Klipper
 - . Firmware für 3-D Drucker

Merkmal/Projekt	Arduino AVR Core	RT-Thread	PX4 Autopilot	Asuswrt-Merlin	Tasmota	Quantum	Klipper
<i>Benutzer-Dokumentation</i>	o.A.	HTML/MD, zahlreich	HTML, zahlreich	HTML/MD, zahlreich	HTML, zahlreich	HTML, zahlreich	HTML/MD, zahlreich
<i>Externe Entwickler-Dokumentation</i>	o.A.	HTML/MD, zahlreich	HTML, zahlreich	o.A.	o.A.	HTML, zahlreich	HTML/MD, zahlreich
<i>DoxyGen*</i>	o.A.	ja	ja	ja	ja	ja	o.A.
<i>Continuos Integration</i>	o.A.	Travis	mehrere Tools	o.A.	Travis u.a.	Travis mit Linter	Travis
<i>Kommentare/Code</i>	0,39	0,43	0,32	0,22	0,21	0,37	0,10
<i>Commits mit Doku</i>	5	4	5	9	7	13	12

Schlußfolgerungen

- Haupthindernis an der Beteiligung von OSS: fehlende oder unklare Dokumentation (GitHub Survey 2017)
- Wartung ist schwieriger als Entwicklung (Umfrage Aspencore 2019)
- Fehlersuche und Termindruck sind die größten Stressquellen (Umfrage Aspencore 2019)
- Flexibilität VS Dokumentation
- Wo ist die aktuellste Dokumentation
- Firmenkultur als Maßnahme – schnell heute, oder schnell morgen
- Selbst-dokumentierender Code
 - Klare und eindeutige Namen
 - Klare Struktur und Formatierung
 - Gezieltes Kommentieren – Was VS Warum
- Code und Dokumentation gehören zusammen
- "If it isn't documented, it doesn't exist." (Mike Pope)

Quellen

Aspencore (2019)

2019 Embedded Markets Study: Integrating IoT and Advanced Technology Designs, Application Development and Processing Environments, https://www.embedded.com/wp-content/uploads/2019/11/EETimes_Embedded_2019_Embedded_Markets_Study.pdf

Barr, M.; Massa, A. (2006)

Programming embedded systems: with C and GNU development tools, Second Edition, Beijing et al.

Beck, K. et al. (2001)

Manifest für Agile Softwareentwicklung , <https://agilemanifesto.org/iso/de/manifesto.html>

Dang, D.; Pack, D. J.; Barrett, S. F. (2017)

Embedded Systems Design with the Texas Instruments MSP432 32-bit Processor , in: Synthesis Lectures on Digital Circuits and Systems 11.3, S. 1-574, o.O.

Digital.ai Software, Inc. (2020)

14th Annual State of Agile Report , <https://explore.digital.ai/state-of-agile/14th-annual-state-of-agile-report>

Fowler, M. (2006)

Continuous Integration, <https://martinfowler.com/articles/continuousIntegration.html>

Gessler, R. (2014)

Entwicklung Eingebetteter Systeme: Vergleich von Entwicklungsprozessen für FPGA- und Mikroprozessor-Systeme Entwurf auf Systemebene , Wiesbaden

GitHub Inc. (2017)

Open Source Survey , <https://opensourcesurvey.org/2017/>

Glass, R. L. (2002)

Facts and Fallacies of Software Engineering, Boston

Goll, J. (2011)

Methoden und Architekturen der Softwaretechnik, Wiesbaden

Quellen

Kothes, L. (2011)

Grundlagen der Technischen Dokumentation: Anleitungen verständlich und normgerecht erstellen , Heidelberg et al.

Lee, E. A.; Seshia, S. A. (2016)

Introduction to embedded systems: A cyber-physical systems approach, Second Edition, o.O.

McConnell, S. (2004)

Code complete: A practical handbook of software construction , Second Edition, o.O.

Sommerville, I. (2011)

Software engineering , 9th Edition, Boston et al.

Sun, J. et al. (2015)

Embedded Firmware Solutions: Development Best Practices for the Internet of Things, o.O.

Wieczorrek, H. W.; Mertens, P. (2011)

Management von IT-Projekten: von der Planung zur Realisierung, 4., überarbeitete und erweiterte Auflage, Heidelberg et al.

Danke für Ihre Aufmerksamkeit