INFORMATIK ▪ CONSULTING ▪ SYSTEMS AG

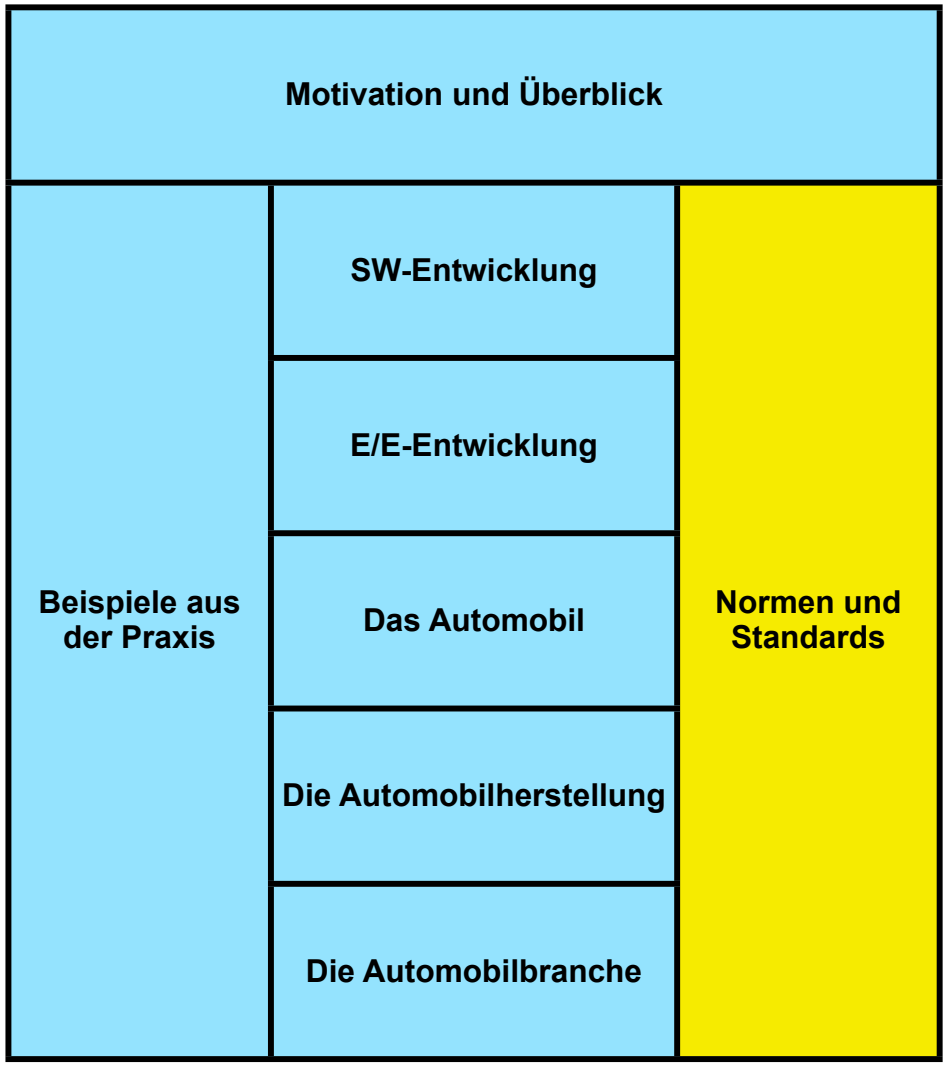
Vorlesung
Automotive Software Engineering
Teil 7 Normen und Standards (4)

Sommersemester 2013

Prof. Dr. rer. nat. Bernhard Hohlfeld

Bernhard.Hohlfeld@mailbox.tu-dresden.de

Technische Universität Dresden, Fakultät Informatik
Honorarprofessur Automotive Software Engineering



AUTOSAR

MOST

CAN

lin

HIS

OSEK/ VDX

ASAM

FlexRay

ISO 26262
Road vehicles - Functional safety

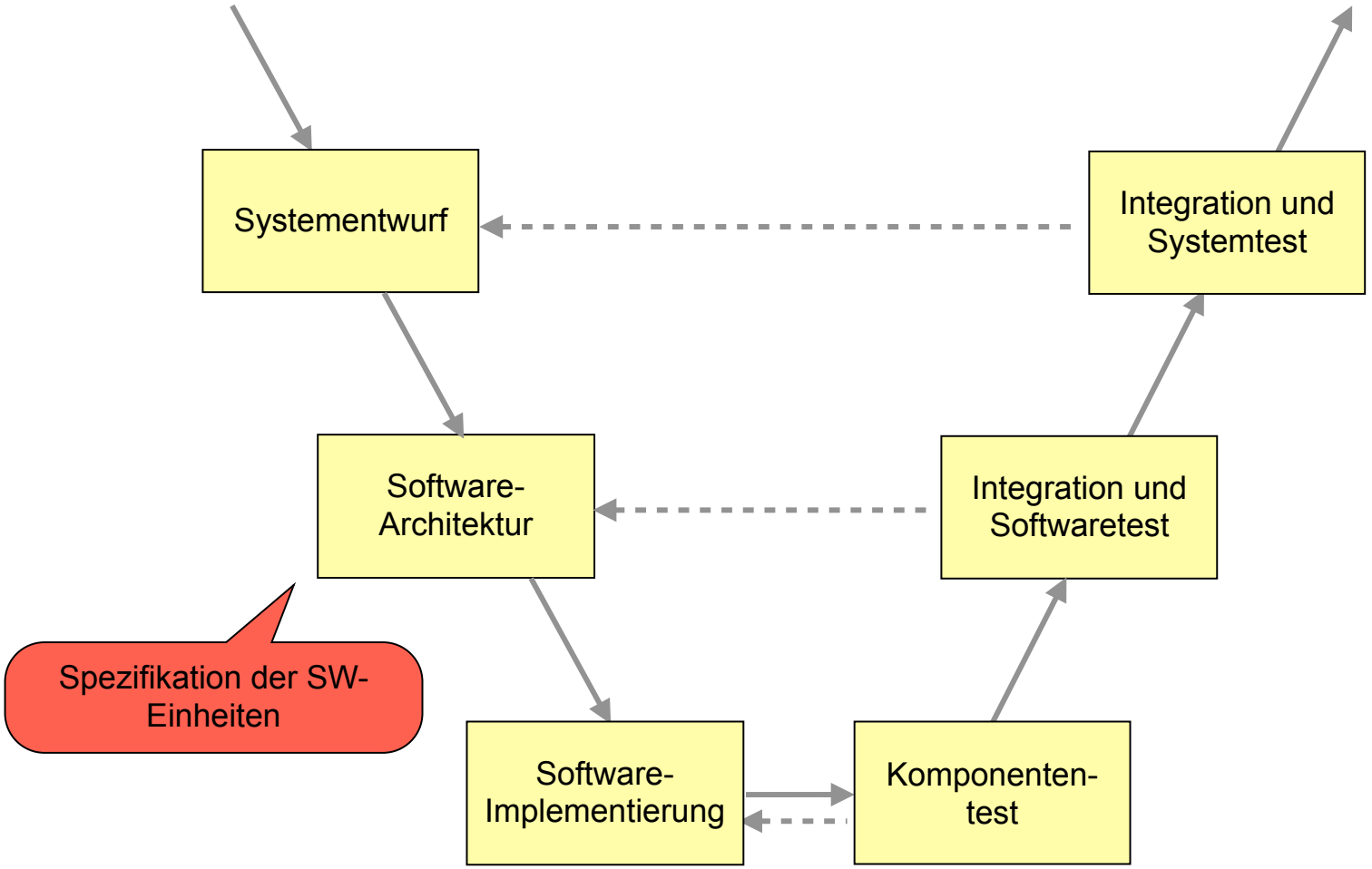
Lernziele
Normen und Standards

- Die Bedeutung von Normen und Standards für industrielle Entwicklung verstehen.
- AUTOSAR Automotive Open System Architecture kennenlernen
  - Motivation
  - Technik
  - Beispiele
- ISO 26262 Road Vehicles Functinal Safety kennenlernen
- Den Begriff COTS einordnen
- Entwurfs- und Codierstandards kennenlernen

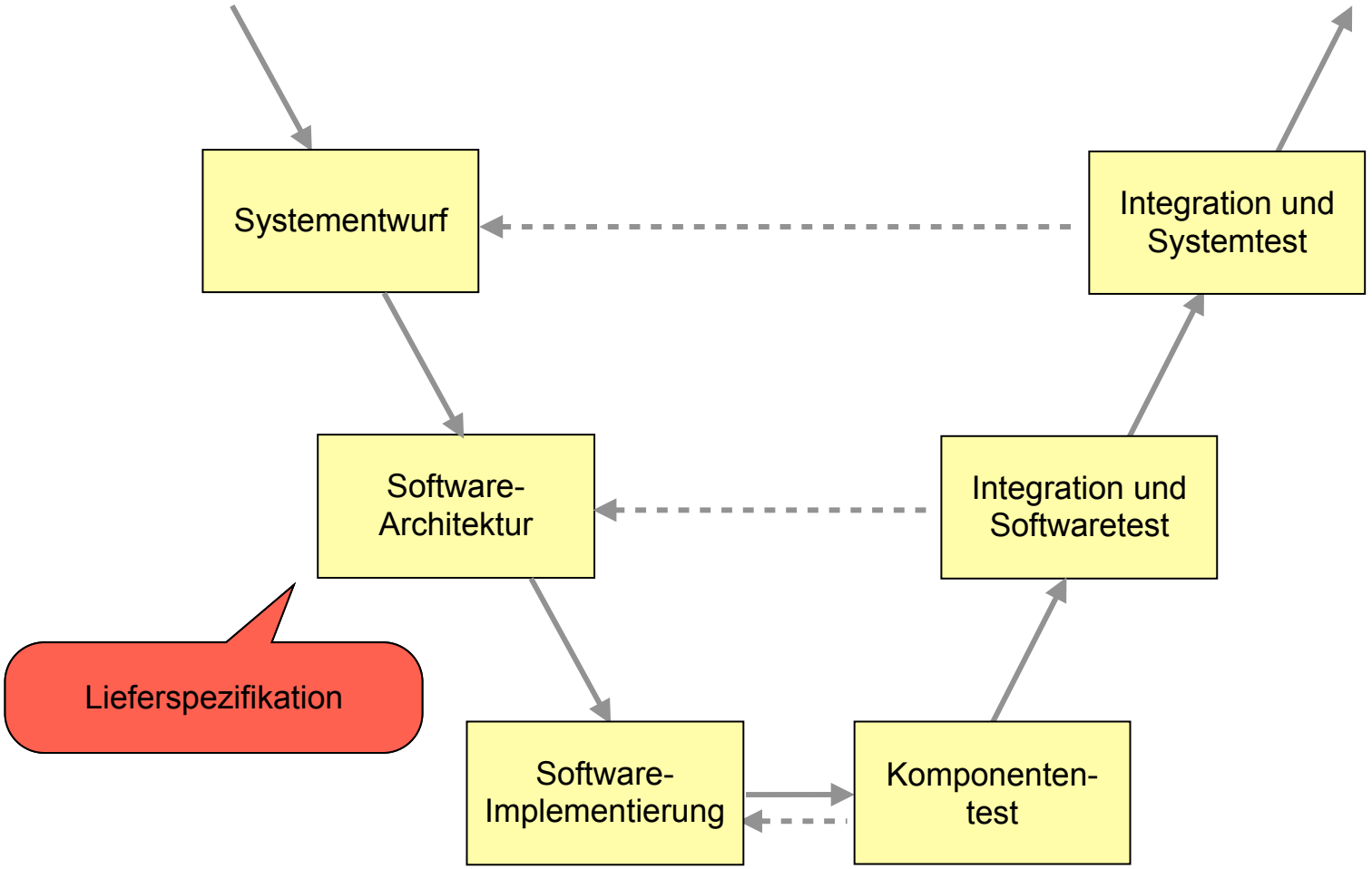
7. Normen und Standards

1. AUTOSAR
2. ARTOP
3. ISO 26262 - Road Vehicles - Functional Safety
4. **COTS**
5. Entwurfs- und Codierstandards

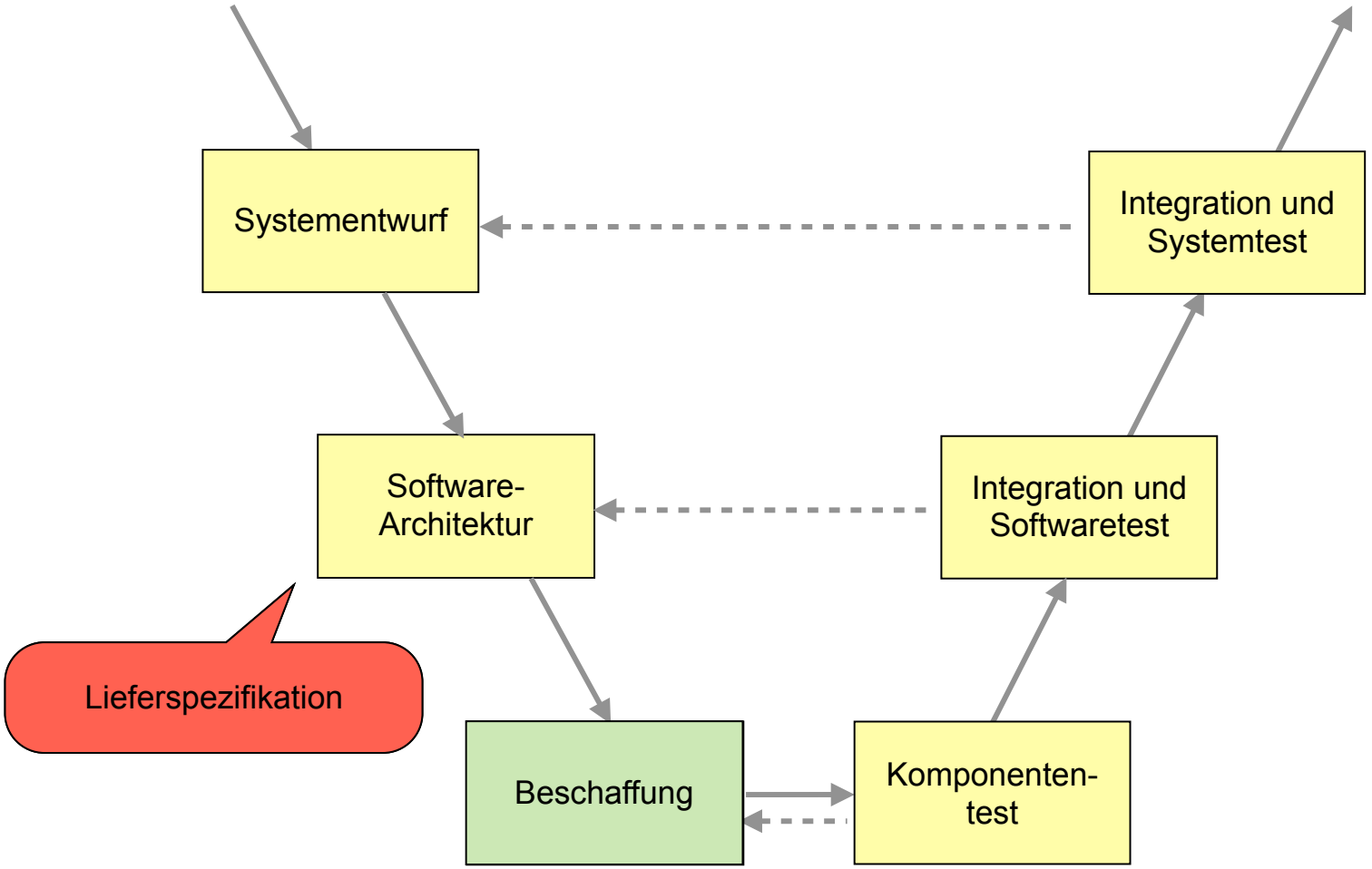
(Vereinfachtes) V-Modell der Softwareentwicklung
Eigenentwicklung



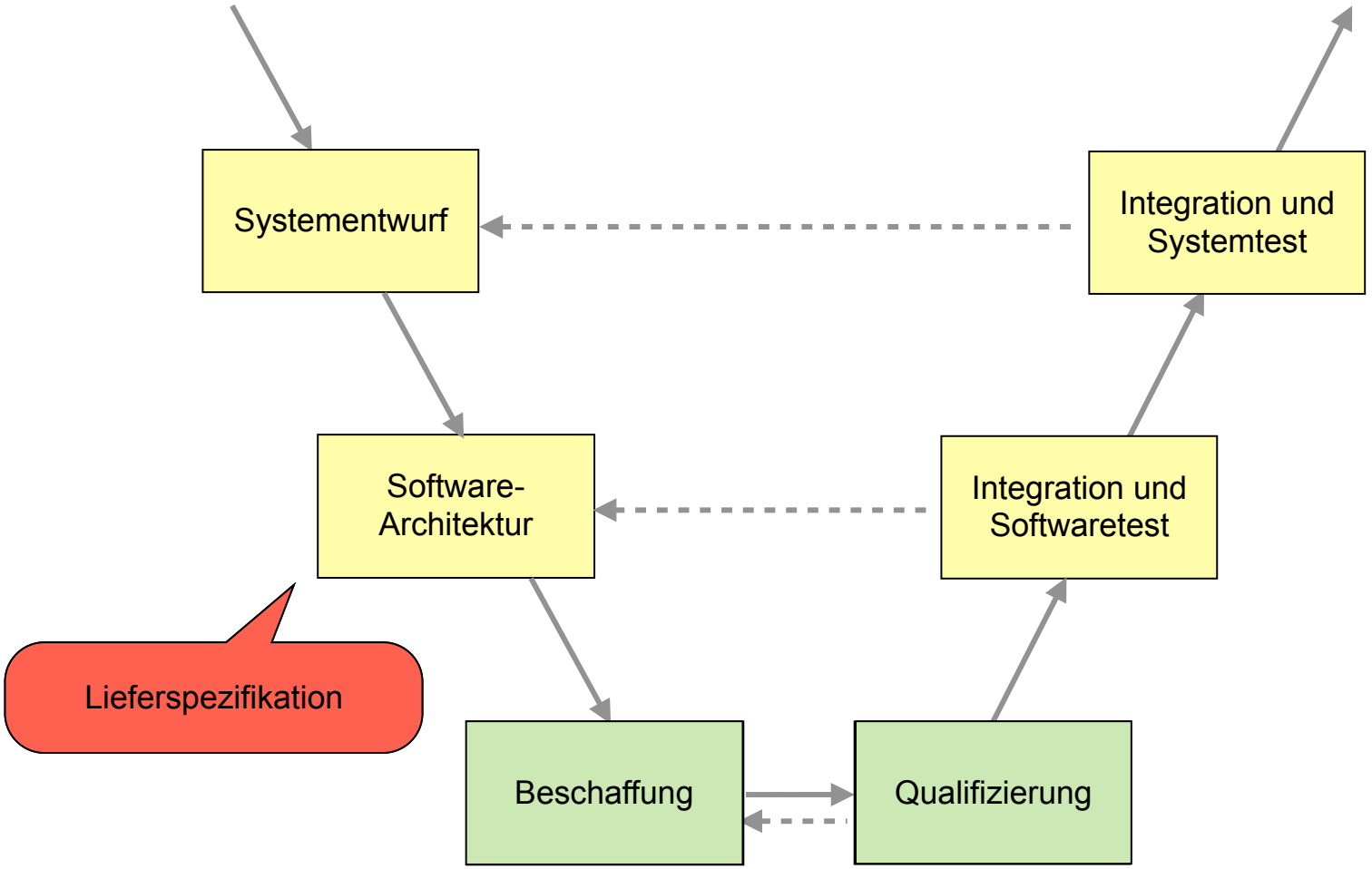
(Vereinfachtes) V-Modell der Softwareentwicklung
Eigenentwicklung



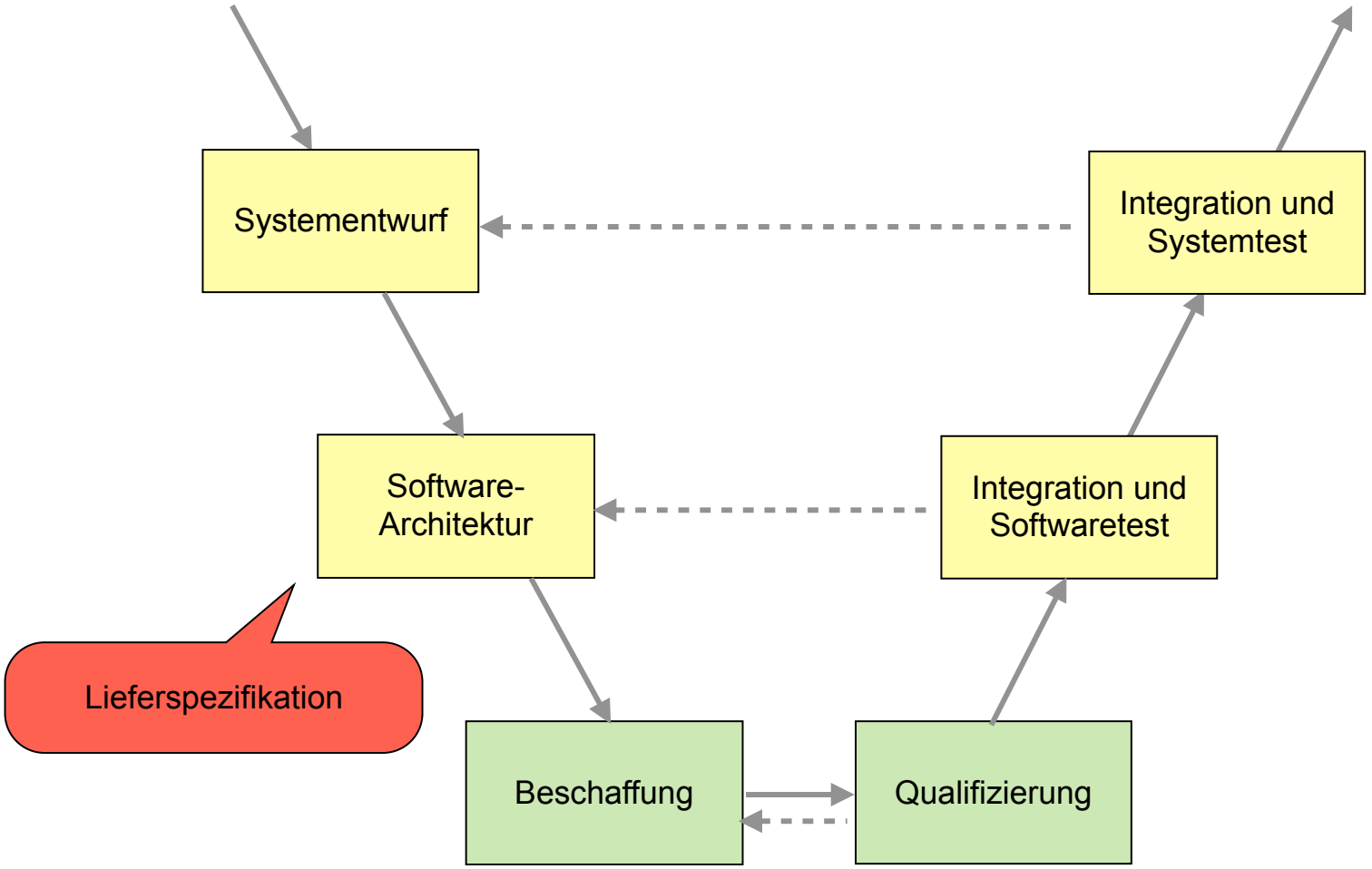
(Vereinfachtes) V-Modell der Softwareentwicklung
Eigenentwicklung



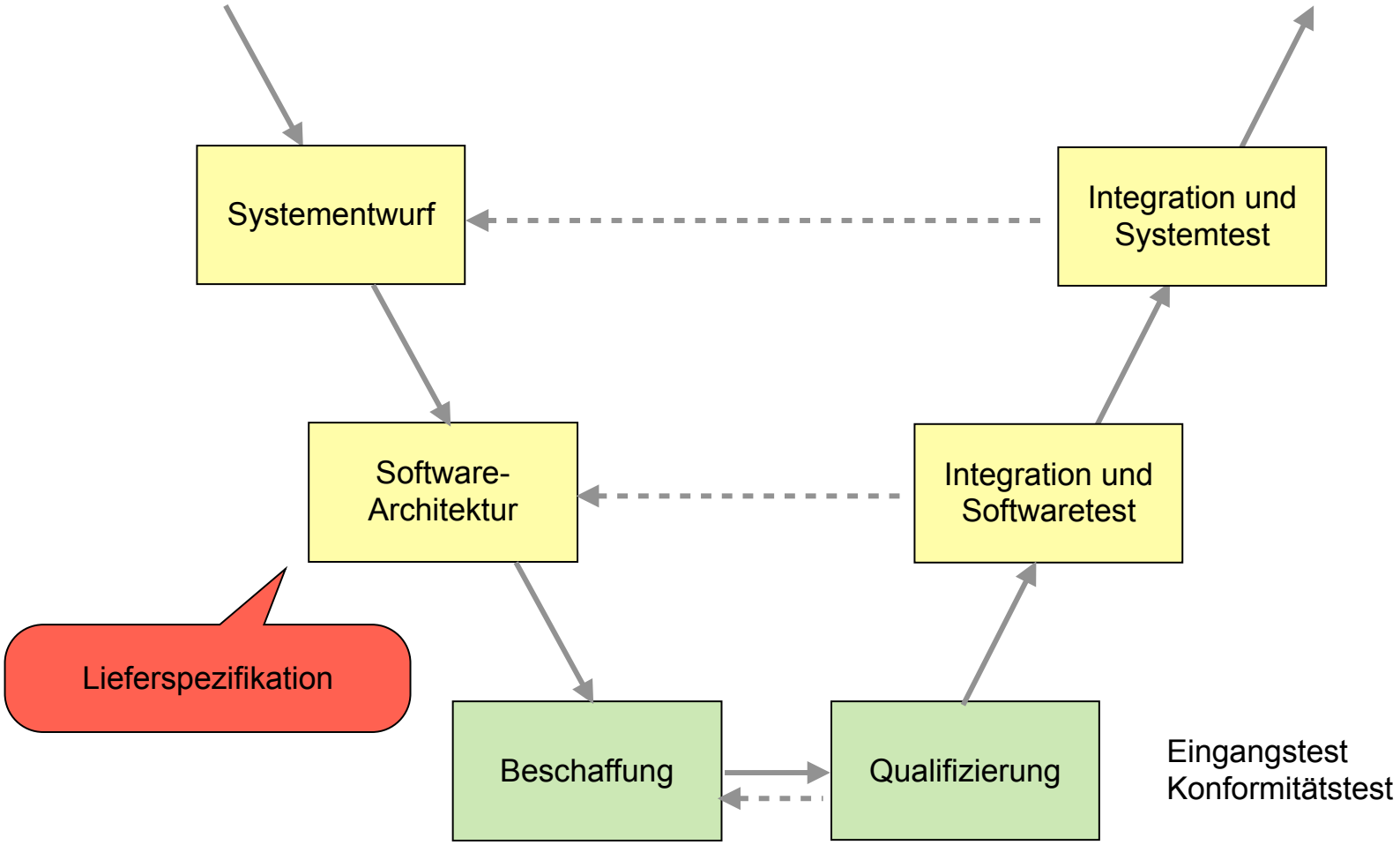
(Vereinfachtes) V-Modell der Softwareentwicklung
Eigenentwicklung



(Vereinfachtes) V-Modell der Softwareentwicklung
Integration von Software aus externen Quellen



(Vereinfachtes) V-Modell der Softwareentwicklung
Integration von Software aus externen Quellen



Software aus externen Quellen - Externe Elemente

- Ein Externes Element ist eine Systemelement, das von aussen beschafft wird. Dabei kann es sich um ein Fertigprodukt oder um ein Element handeln, das im Rahmen eines Unterauftrags entwickelt wird. Ein Externes Element realisiert die Anforderungen und Vorgaben der Lieferspezifikation.
- Wird das Externe Element im Rahmen eines Unterauftrages erstellt, so wird es gemäß der Abnahmekriterien der zugehörigen Lieferspezifikation geprüft.
- Handelt es sich um ein Fertigprodukt, so wird es einer Eingangsprüfung gemäß der Lieferspezifikation unterzogen.
- Nach erfolgreicher Prüfung wird das Externe Element gemäß Konfigurationsmanagementplan in die Produktbibliothek übernommen.
- Die Integration des Externen Elementes erfolgt analog zu der eines internen Systemelements gemäß Integrationsplan.

Software aus externen Quellen - Externe Elemente

- Ein Externes Element ist eine Systemelement, das von aussen beschafft wird. Dabei kann es sich um ein Fertigprodukt oder um ein Element handeln, das im Rahmen eines Unterauftrags entwickelt wird. Ein Externes Element realisiert die Anforderungen und Vorgaben der Lieferspezifikation.
- Beauftragung
- Fertigprodukt: COTS
- Fertigprodukt: Open Source

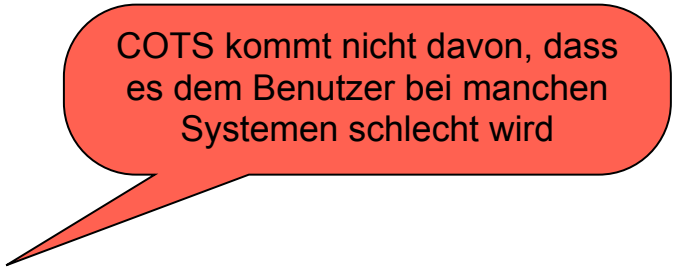
Software aus externen Quellen - Externe Elemente

Externes Element	**Kosten:** -Beschaffung -Installation -Betrieb -Schulung	**Rechtliche Aspekte**
Beauftragung	-hoch -gering bis mittel -mittel	Geregelt im Liefervertrag, insbesondere Exklusivität auf Kunden- und Lieferantenseite
Fertigprodukt: COTS	-gering bis hoch -gering bis mittel -gering bis hoch	Geregelt im Kaufvertrag, Lieferant bleibt i.a. Eigentümer, Nutzungsrecht, keine Weitergabe, Vorsicht bei Nutzung in Kundenprojekten
Fertigprodukt: Open Source	-sehr gering -gering bis mittel -gering bis hoch	Vorsicht bei Nutzungsbedingungen: Modifizierte Open Source muss Open Source bleiben, Vorsicht bei Nutzung in Kundenprojekten

Komponentenentwicklung und Systementwicklung

- Komponentenentwicklung
  - Analyse und Entwurf von Komponenten
  - Computerspiele
  - SAP
  - Keine oder wenig Bezug zu realer Umwelt
  - Benutzer und betriebliche Abläufe müssen sich der EDV anpassen, nicht umgekehrt
- Systementwicklung
  - Analyse und Entwurf des Systems als Ganzes
  - Liefert Vorgaben für Komponentenentwicklung
  - Embedded Systems
    - Automotive
    - Luftfahrt
    - Bahnen
    - Medizin
  - Hoher Bezug zu realer Umwelt
  - Systeme haben sich z.B. der Physik anzupassen

Komponentenentwicklung und Systementwicklung



- Komponentenentwicklung
  - Analyse und Entwurf von Komponenten
  - Computerspiele
  - SAP
  - Keine oder wenig Bezug zu realer Umwelt
  - Benutzer und betriebliche Abläufe müssen sich der EDV anpassen, nicht umgekehrt
- Systementwicklung
  - Analyse und Entwurf des Systems als Ganzes
  - Liefert Vorgaben für Komponentenentwicklung
  - Embedded Systems
    - Automotive
    - Luftfahrt
    - Bahnen
    - Medizin
  - Hoher Bezug zu realer Umwelt
  - Systeme haben sich z.B. der Physik anzupassen

COTS

- Als commercial off-the-shelf, oder auch components-of-the-shelf (englisch für Kommerzielle Produkte aus dem Regal) oder kurz COTS werden seriengefertigte Produkte aus dem Elektronik- oder Softwaresektor (vgl. Standardsoftware) bezeichnet, die in großer Stückzahl völlig gleichartig (ugs. „von der Stange“) aufgebaut verkauft werden. Dies kann beispielsweise bei Office-Produkten oder Warenwirtschaftssystemen praktiziert werden.
- Dadurch, dass ab Werk keine Anpassungen an die Bedürfnisse des Individualkunden vorgenommen werden, erhofft sich der Nutzer weitgehende Kosteneinsparungen, da hier die Entwicklungskosten nicht vom Auftraggeber alleine, sondern vom Markt getragen werden. Vor allem im Behördenbereich, sowie bei militärischen Anwendungen findet gerade ein weitgehender Umstieg zu COTS statt; beim Militär ganz besonders von eigens entwickelten, robusten Geräten hin zu Lösungen mit Standard-PC-Hardware.
- Quelle: Wikipedia (deutsch)

COTS

- In the United States, Commercially available Off-The-Shelf (COTS) is a Federal Acquisition Regulation (FAR) term defining a nondevelopmental item (NDI) of supply that is both commercial and sold in substantial quantities in the commercial marketplace, and that can be procured or utilized under government contract in the same precise form as available to the general public. For example, technology related items, such as computer software, hardware systems or free software with commercial support, and construction materials qualify, but bulk cargo, such as agricultural or petroleum products, do not...
- COTS purchases are alternatives to in-house developments or one-off government-funded developments. COTS typically requires configuration that is tailored for specific uses. The use of COTS has been mandated across many government and business programs, as such products may offer significant savings in procurement, development, and maintenance.
- Quelle: Wikipedia (englisch)

Beispiel:
ISO 26262 Road vehicles - Functional safety
siehe auch 5. Entwurfs- und Codier-Standards

DRAFT INTERNATIONAL STANDARD ISO/DIS 26262-6

ISO/TC **22**/SC **3** Secretariat: **DIN**

Voting begins on:
2009-07-08

Voting terminates on:
2009-12-08

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION

Road vehicles — Functional safety —

Part 6:
Product development: software level

Véhicules routiers — Sécurité fonctionnelle —

Partie 6: Développement du produit: niveau logiciel

ICS 43.040.10

7 Software architectural design
7.4 Requirements and recommendations

- 7.4.6 Every safety-related software component shall be categorised as one of the following:
  a) newly developed;
  b) reused with modifications;
  c) reused without modifications;
  d) or a COTS product.
- 7.4.7 Safety-related software components that are newly developed or reused with modifications shall be developed in accordance with ISO 26262:—.
- **7.4.8 Safety-related software components that are reused without modifications or that are COTS products shall be qualified in accordance with ISO 26262-8:—, Clause 12.**

  NOTE The use of qualified software components does not affect the applicability of Clauses 10 and 11. However, some activities described in Clauses 8 and 9 may be omitted.

- 8 Software unit design and implementation
- 9 Software unit testing.
- 10 Software integration and testing
- 11 Verification of software safety requirements

7 Software architectural design
7.4 Requirements and recommendations

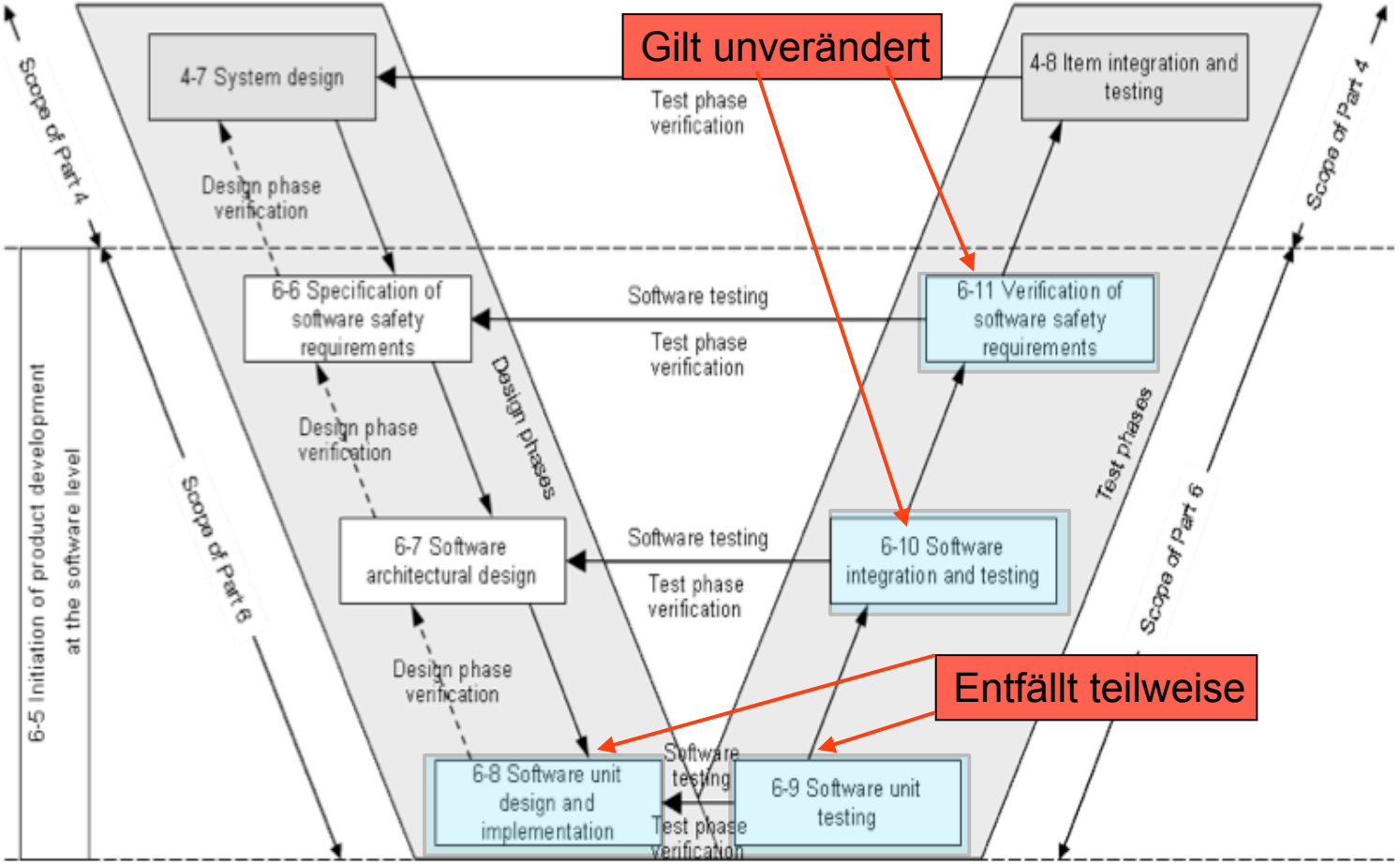
- 7.4.6 Every safety-related software component shall be categorised as one of the following:
  a) newly developed;
  b) reused with modifications;
  c) reused without modification;
  d) or a COTS product.
- 7.4.7 Safety-related software components developed or reused with modifications shall be developed in accordance with ISO 26262-6:—, Clause 10.
- **7.4.8 Safety-related software components reused without modifications or that are COTS products shall be qualified in accordance with ISO 26262-8:—, Clause 12.**

  NOTE The use of qualified software components does not affect the applicability of Clauses 10 and 11. However, some activities described in Clauses 8 and 9 may be omitted.

Nummerierungen in diesem Abschnitt beziehen sich auf ISO/DIS 26262 Road vehicles - Functional safety Part 6: Product development: software level

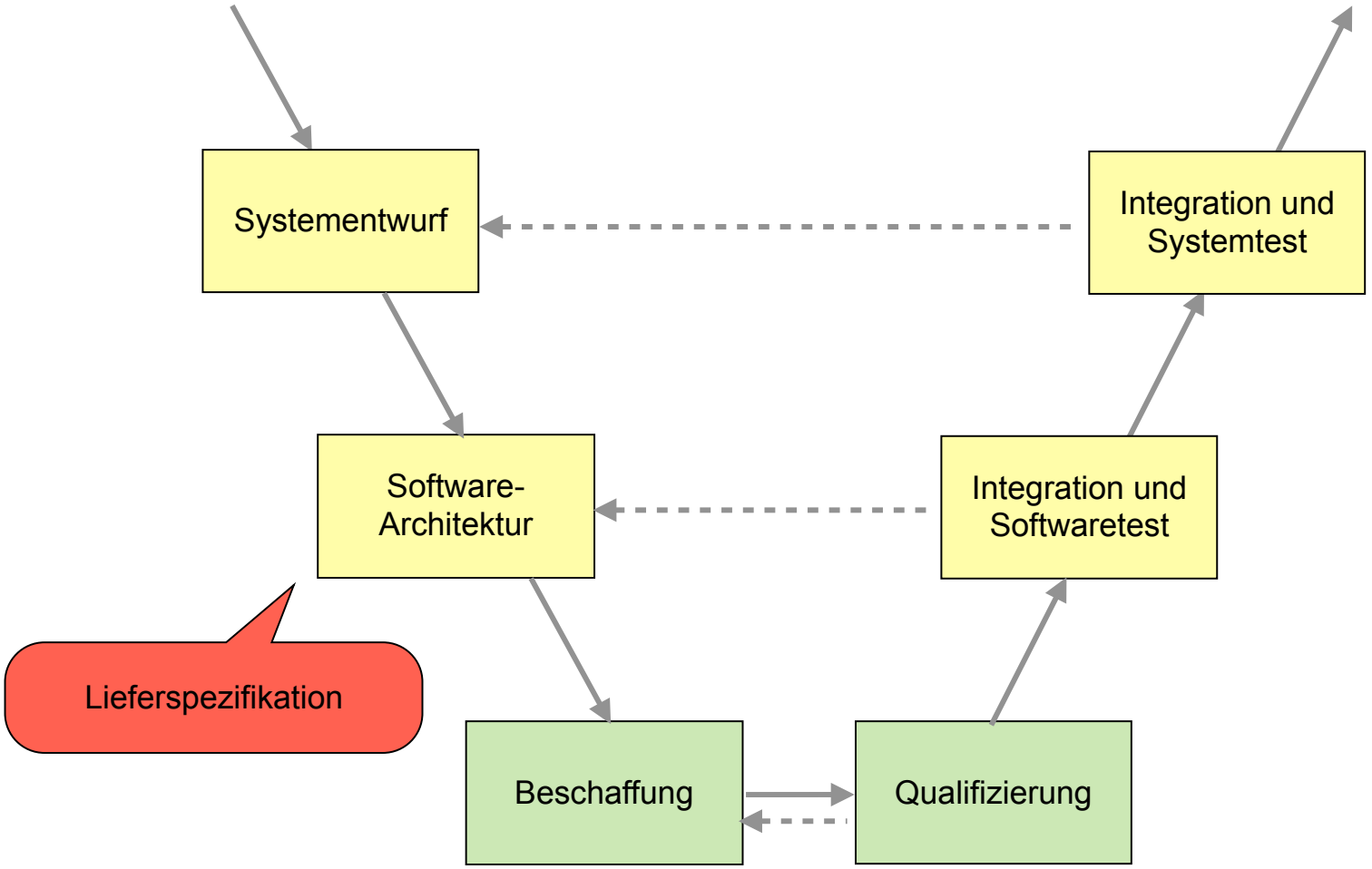
- 8 Software unit design and implementation
- 9 Software unit testing.
- 10 Software integration and testing
- 11 Verification of software safety requirements

Figure 2
Reference phase model for the software development

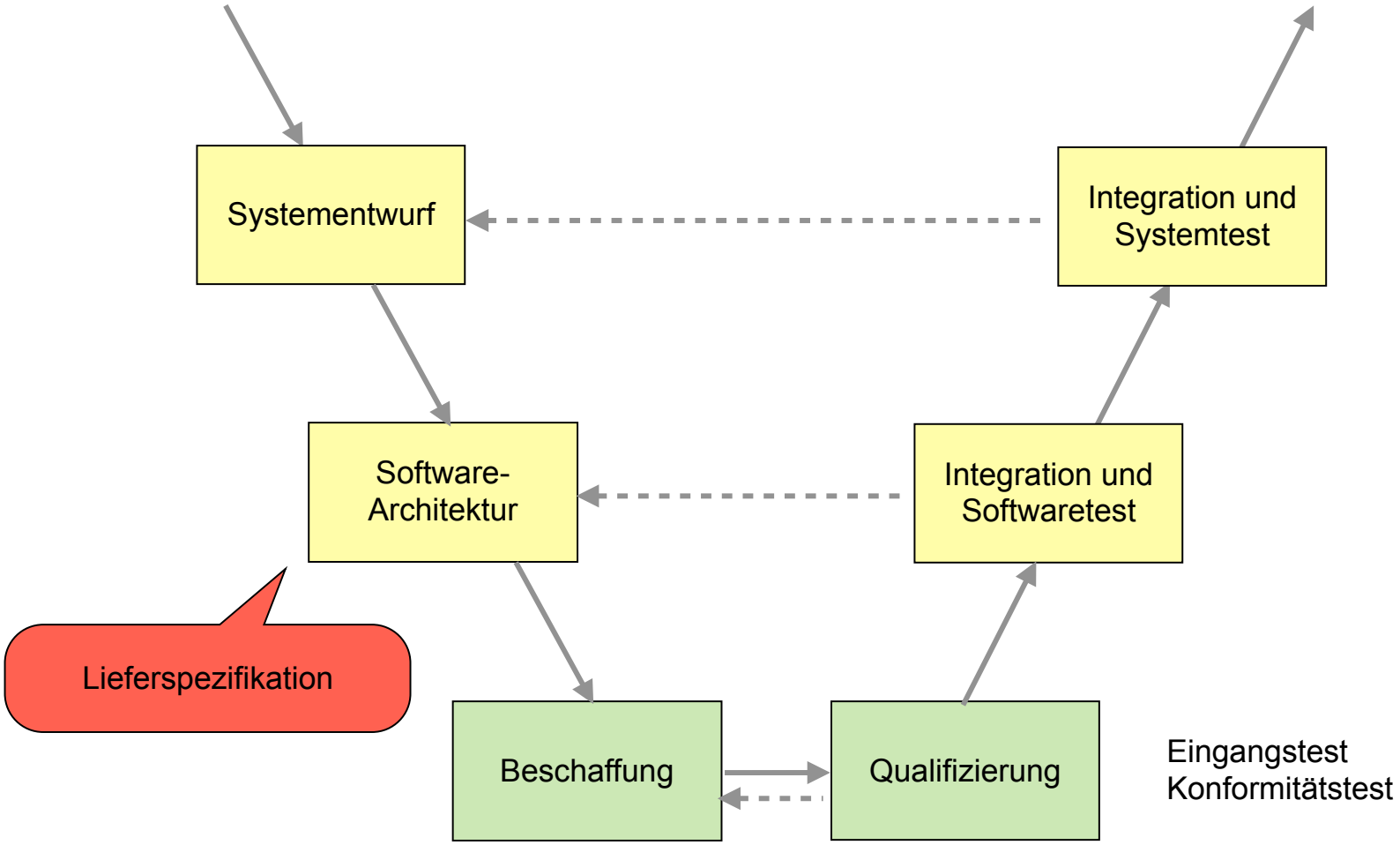


Zur Erinnerung:

(Vereinfachtes) V-Modell der Softwareentwicklung
Integration von Software aus externen Quellen



(Vereinfachtes) V-Modell der Softwareentwicklung
Integration von Software aus externen Quellen



12 Qualification of software components (1)

- 12.1 Objectives

  The first objective of the qualification of software components is to enable the re-use of existing software components as part of items, systems or elements developed in compliance with ISO 26262 without completely re-engineering the software components.

  The second objective of the qualification of software components is to show their suitability for re-use.

- 12.2 General

  The re-use of qualified software components avoids re-development for software components with similar or identical functionality.

  NOTE Software components are understood to be software libraries from third-party suppliers (COTS), as well as in-house components already in use in electronic control units or generic components designed and developed for re-use across projects.

  EXAMPLE Graphical libraries, mathematical libraries, operating systems, operating system services, databases, device driver software

12 Qualification of software components (1)

Nummerierungen in diesem Abschnitt beziehen sich auf ISO/DIS 26262 Road vehicles - Functional safety Part 8: Supporting Processes

- 12.1 Objectives

  The first objective of the qualification of software components is to enable the re-use of existing software components as part of item development in compliance with ISO 26262 without completely re-engineering the

  The second objective of [illegible] how their suitability for re-use.

- 12.2 General

  The re-use of qualified software components avoids re-development for software components with similar or identical functionality.

  NOTE Software components are understood to be software libraries from third-party suppliers (COTS), as well as in-house components already in use in electronic control units or generic components designed and developed for re-use across projects.

  EXAMPLE Graphical libraries, mathematical libraries, operating systems, operating system services, databases, device driver software

12 Qualification of software components (2)

- 12.3 Inputs to this clause
- 12.3.1 Prerequisites

  The following information shall be available:

  - **Pre-determined maximum target ASIL**
  - **Requirements of the software component.**
- 12.3.2 Further supporting information

  The following information may be considered:

  - **Results of previous verification measures of the software component.**

12 Qualification of software components (3)

- 12.4 Requirements and recommendations
- 12.4.1 To be able to treat a software component as qualified, the following shall be available:

a) specification of the software component (see 12.4.3.1);

b) evidence that the software component complies with its requirements (see 12.4.3.2, 12.4.3.3, and 12.4.3.4);

c) evidence that the software component is suitable for its intended use (see 12.4.4).

NOTE Some re-engineering activities can be performed to comply with this subclause in case of previously developed software components.

12 Qualification of software components (4)

- 12.4.2 The planning of qualification of a software component shall determine:
  - **unique identification of the software component;**
  - **the pre-determined maximum target ASIL of any safety requirement which might be violated if the software component performs incorrectly (siehe 8. Entwurfs- und Codier-Standards); and**
  - **the activities that shall be carried out to qualify the software component.**
- 12.4.3 Qualification of a software component (1)
- 12.4.3.1 The specification of the software component shall include:

  a) **requirements of the software component;**

  EXAMPLE 1:

  - Functional requirements;
  - Accuracy of algorithm or numerical accuracy, ..;
  - behaviour in case of failure;
  - response time;
  - resource usage;
  - requirements on the runtime environment; and
  - behaviour in an overload situation (robustness).

Zurück

12 Qualification of software components (5)

Zurück

- 12.4.3 Qualification of a software component (2)
- 12.4.3.1 The specification of the software component shall include:

a) requirements of the software component;

b) description of the configuration;

NOTE 1 For software components that contain more than one software unit, the description of the configuration includes the unique identification and configuration of each software unit.

c) interfaces description;

d) application manual;

e) description of the software component integration;

NOTE 2 Description might include the development tools required to integrate and use the software component;

f) reactions of the functions under anomalous operating conditions;

EXAMPLE 2 Re-entrant calling of non-re-entrant software component functions.

g) dependencies with other software components; and

h) description of known anomalies with corresponding work-around measures.

Re-entrant / eintrittsinvariant
Quelle: Wikipedia

- Eintrittsinvarianz

  Eine Routine bzw. Methode wird als eintrittsinvariant (engl. reentrant) oder auch wiedereintrittsfähig bezeichnet, wenn sie so implementiert ist, dass sie von mehreren Prozessen gleichzeitig ausgeführt werden kann. Dabei dürfen sich die gleichzeitig ausgeführten Instanzen nicht in die Quere kommen. Die Ausführung jeder Instanz läuft also gleich ab, egal wie viele andere Instanzen es noch von dieser Methode gibt.

  Das Ziel eines Designs für eine eintrittsinvariante Methode ist es, sicherzustellen, dass kein Teil des Programmcodes selbst durch die Methode geändert wird und dass prozesseigene Informationen wie beispielsweise lokale Variablen in getrennten Speicherbereichen gehalten werden.

  Eintrittsinvariante Programmkonstrukte sind die Basis für viele Multitasking-Systeme (Threadsicherheit).

12 Qualification of software components (6)

- 12.4.3 Qualification of a software component (3)
- 12.4.3.2 To show that a software component complies with its requirements the verification of this software component shall meet the following criteria:

a) **the verification shall show a requirement coverage in accordance with ISO 26262-6:—, Clause 9 for the maximum target ASIL;**

NOTE This verification is primarily based on requirement-based testing. The results of requirement-based tests of the software component executed during its development or during previous integration tests can be used.

EXAMPLE 1 Application of a dedicated qualification test suite, analysis of all the tests already executed during the implementation and any integration of the software component.

b) The verification shall cover both normal operating conditions and behavior in case of failure;

c) The software component errors occurring during verification shall be analysed; together with information on their possible consequences and with measures to avoid or detect them.

EXAMPLE 2 Functional errors, runtime errors, incorrect timing, violation of data integrity, erroneous operating and resource usage

12 Qualification of software components (7)

- 12.4.3 Qualification of a software component (4)
- 12.4.3.3 This subclause applies to ASIL D in accordance with 4.3.

  The structural coverage shall be measured in accordance with ISO 26262-6:—, Clause 9 to evaluate the completeness of the test cases. If necessary, additional test cases shall be specified or a rationale shall be provided.

- **12.4.3.4 The verification shall only be valid for an unchanged implementation of the software component.**

Zurück

- **12.4.3.5 The qualification of a software component shall be documented including the following information:**

  a) unique identification and configuration of the software component;

  b) person or organisation who carried out the qualification;

  c) the environment used for qualification;

  d) results of the verification measures applied to qualify the software component; and

  e) the pre-determined maximum target ASIL of any safety requirement which might be violated if the software component performs incorrectly.applied.

12 Qualification of software components (8)

- 12.4.4 Verification of qualification of a software component
- 12.4.4.1 The results of qualification of a software component together with the validity of these results regarding the intended use of the software component shall be verified. If necessary, additional measures shall be applied.

  NOTE The validity of the qualification may be influenced when the qualification has been performed in the context of another industrial or automotive domain.

  EXAMPLE Engine control, body control and chassis control are different automotive domains. Railways and civil avionics are different industrial domains.
- 12.4.4.2 The specification of the software component shall comply with the requirements of the planned use of this software component.
- **12.5 Work products**
- 12.5.1 Software component documentation resulting from requirement 12.4.3.1.
- 12.5.2 Software component qualification report resulting from requirements 12.4.3.5.
- 12.5.3 Safety plan (refined), resulting from requirements 12.4.2