Software-Qualitätsmanagement

Vorlesung im Modul 10-202-2319 Software-Management

Sommersemester 2013

Prof. Dr. Hans-Gert Gräbe

http://bis.informatik.uni-leipzig.de/HansGertGraebe

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Grundsätze für die Qualitätssicherung in der Software-Entwicklung

- produkt- und prozessabhängige Qualitätszielbestimmung
- quantitative Qualitätssicherung
- maximale konstruktive Qualitätssicherung
- frühzeitige Fehlerentdeckung und -behebung
- entwicklungsbegleitende, integrierte Qualitätssicherung
- unabhängige Qualitätssicherung

Prinzip der produkt- und prozessabhängigen Qualitätszielbestimmung

Nur 50% der Betriebe legen überhaupt Qualitätsmerkmale fest

Explizite und transparente Qualitätszielbestimmung ist vor Beginn des Entwicklungsprozesses aber äußerst hilfreich.

- Die in der Qualitätszielbestimmung festgelegten Qualitätsanforderungen werden vom Auftraggeber für den Abnahmetest verwendet.
- Für den Software-Lieferanten ergeben sich aus den Qualitäts-anforderungen die Maßnahmen für den Entwicklungsprozess und die Qualitätsprüfung.

Vorteil: Prinzip der produkt- und prozessabhängigen Qualitätsziel-bestimmung vor Beginn des Entwicklungsprozesses bringt Planungs- und Kalkulationssicherheit.

Nachteil: Qualitätssicherung beansprucht zusätzliche Ressourcen

Prinzip der quantitativen Qualitätssicherung

Nachteile:

- Schwierigkeiten bei der Quantifizierung von Soll- und Istwerten:
  - Metriken sind ziel- und kontextabhängig
  - Anzahl der Variationsparameter ist um ein Vielfaches höher als bei traditionellen Produktionsprozessen
- kreativer Charakter vieler Aspekte der Software-Entwicklung
- unkontrollierte Variabilität von Entwicklungsprozessen

Vorteile:

- Messen ist geeignet
  - zum besseren Verständnis unterschiedlicher Qualitätsmerkmale
  - zur besseren Planung und Sicherung von Qualitätsmerkmalen
  - zur Verbesserung von Entwicklungsansätzen
- Methoden und Werkzeuge zur Planung und Durchführung der Datenerfassung sowie zur Auswertung und Präsentation von Messdaten sind oft schon vorhanden.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Prinzip der maximalen konstruktiven Qualitätssicherung

„Fehler die nicht gemacht werden, brauchen auch nicht behoben werden“ ist das Ziel, das von der maximalen konstruktiven Qualitätssicherung verfolgt wird.

Reduzierung der analytischen Maßnahmen über Einschränkung der Variationsbreite durch vorausblickende konstruktive Maßnahmen.

Vorteile:

- direkte Verbesserung der Produktivität
- Reduktion analytischer Maßnahmen
- Voraussetzung für analytische Maßnahmen
- Vermeidung von Fehlern

Nachteile:

- Konstruktive Maßnahmen schränken die Variabilität und damit die Kreativität ein.

Prinzip der frühzeitigen Fehlerentdeckung und –behebung

Fehler ist:

- Abweichung von den Anforderungen des Auftraggebers
- Inkonsistenz in den Anforderungen

Ziel ist es, Fehler gar nicht erst zu machen (konstruktive Maßnahmen) oder zum frühestmöglichen Zeitpunkt zu erkennen und zu beheben

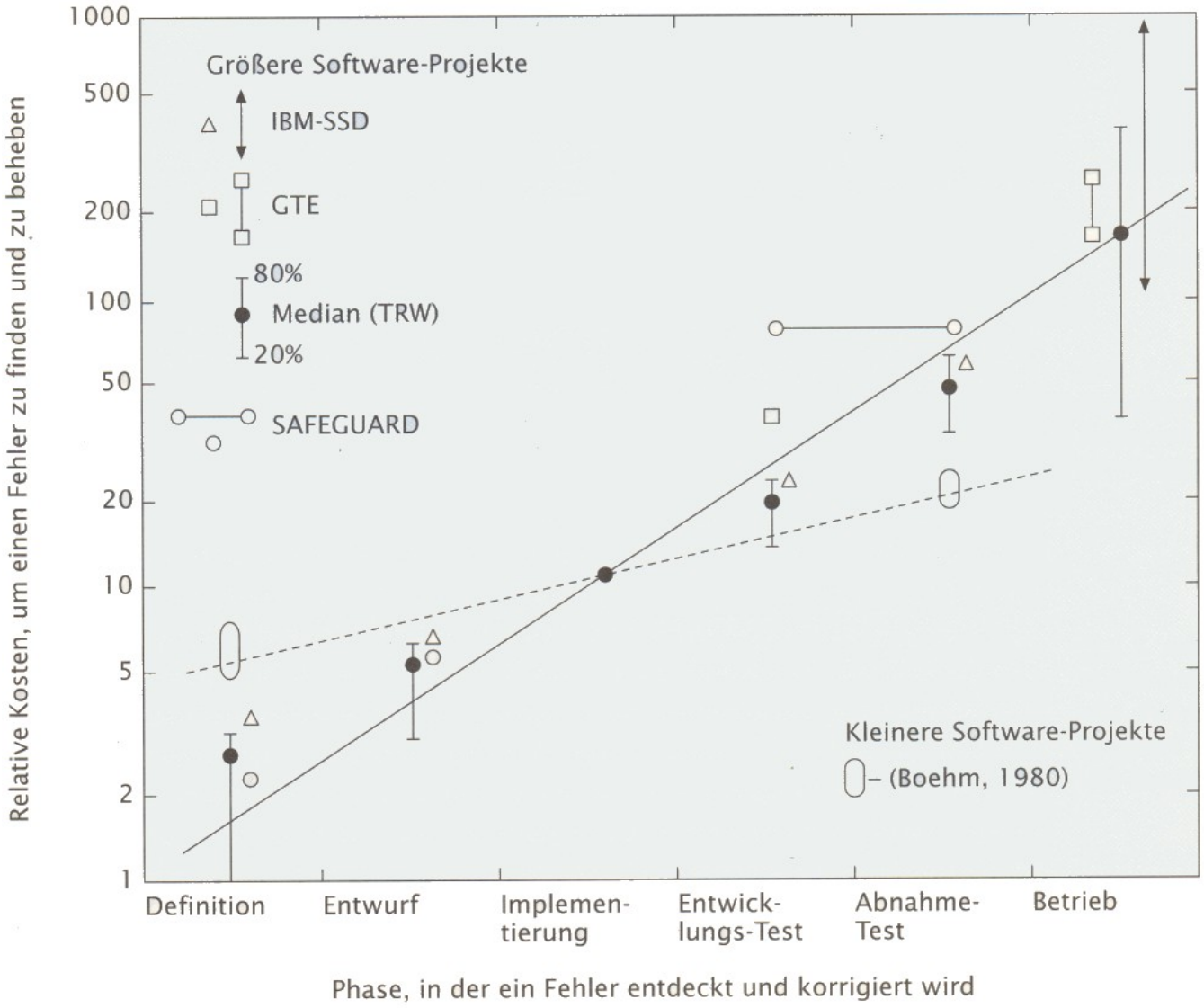
Vorteile:

- Fehler in späteren Phasen werden vermieden
- Kosten werden reduziert
- mit höherer Wahrscheinlichkeit werden Fehler richtig korrigiert
- die Fehlerfortpflanzung wird reduziert

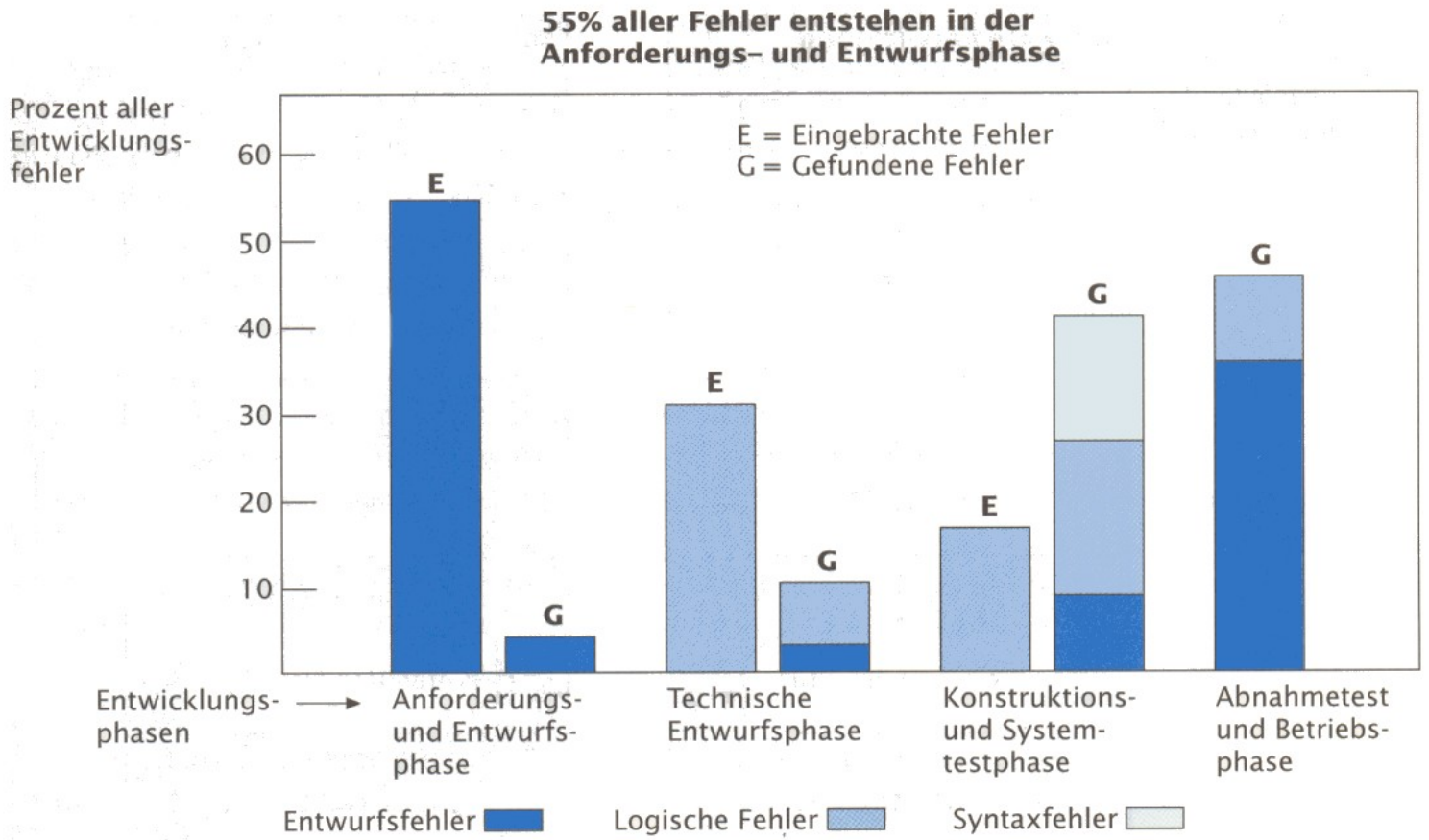
Nachteil:

- Langsamerer Projektfortschritt durch Fehlersuche bereits in frühen Projektphasen
- Wird aber durch die Vorteile mehr als aufgewogen

Folgerung: Viel Aufmerksamkeit den frühen Projektphasen



Kosten einer verzögerten Fehlerentdeckung [Boehm 76]



Fehlerbeseitigungskosten (abgeleitet von Alberts 1985)

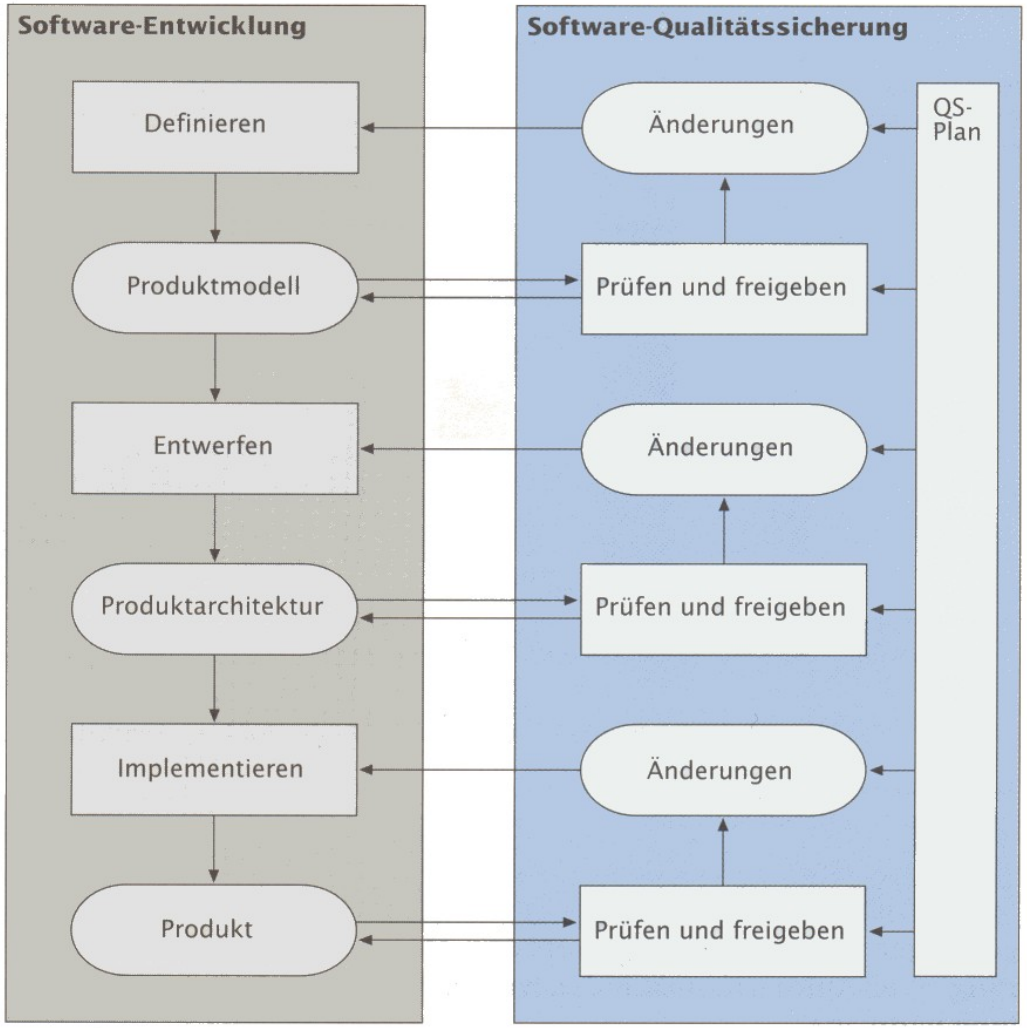
Prinzip der entwicklungsbegleitenden, integrierten Qualitätssicherung

Um das Prinzip der frühzeitigen Fehlerentdeckung zu realisieren, ist Softwareentwicklung begleitende und in den Entwicklungsprozess integrierte Qualitätssicherung nötig.

Gründe:

- Einbettung der Qualitätssicherung in das organisatorische Ablaufmodell der Software-Entwicklung
- Qualitätssicherung findet zu dem Zeitpunkt statt, zu dem sie im Entwicklungsprozess angebracht ist
- Qualitätssicherung wird nicht als Fremdkörper empfunden, sondern gehört per se zur Software-Erstellung

5. Prinzipien der Qualitätssicherung



Prinzip der unabhängigen Qualitätssicherung

*„... Testing is a **destructive** process, even a sadistic process ...“*

- Der Entwickler eines Produkts ist am schlechtesten geeignet, um durch Anwendung analytischer QS-Maßnahmen die Ergebnisse seiner Tätigkeit zu betrachten.
- Entwickler darf aber seine Aufgaben im Bereich QS nicht vernachlässigen.
- Zwei organisatorische Alternativen:
  - Qualitätssicherung als organisatorisch unabhängiger Teil von der Gestaltung
  - Qualitätssicherung als Teil der Entwicklung

Prinzip der unabhängigen Qualitätssicherung
Qualitätssicherung als organisatorisch unabhängiger Teil

Vorteile:

- Entwicklung übt keinen „Druck“ auf die Qualitätssicherung aus
- Neutralität
- klare Budgetaufteilung
- Betonung der Qualitätssicherung

Nachteile:

- Gefahr der Isolierung der Qualitätssicherung von der Entwicklung
- gleichmäßige Personalauslastung ist unter Umständen sicherzustellen

Prinzip der unabhängigen Qualitätssicherung
Qualitätssicherung als Teil der Entwicklung

Vorteile:

- flexiblere Einsetzung des Personals
- Qualitätssicherung „bekommt alles mit“
- Erleichterung der Teamarbeit
- vertrauensvolle Zusammenarbeit

Nachteile:

- Entwicklungsmanagement kann „Druck“ auf die Qualitätssicherung ausüben
- Budgetmittel können zugunsten der Entwicklung umverteilt werden

Prinzip der unabhängigen Qualitätssicherung

Personalalternativen

3 Möglichkeiten für die Personalausstattung der Qualitätssicherung:

Personal arbeitet nur in der Qualitätssicherung

- Ermöglicht die Einstellung von Mitarbeitern mit einem hohen Spezialisierungsgrad, aber diese bekommen nie Erfahrung mit der Entwicklung von Software.

Jeder Mitarbeiter rotiert in festgelegten Abständen zwischen der Qualitätssicherung und der Entwicklung

- Ermöglicht einen systematischen Wissenstransfer, aber Mitarbeiter müssen Tätigkeiten durchführen, zu denen sie keine „Lust“ haben.

Jeder Mitarbeiter arbeitet sowohl an der Qualitätssicherung als auch an den Entwicklung (in der Praxis üblich)

- Ermöglicht einen flexiblen Personaleinsatz, aber die Vermischung der Entwicklung und Qualitätssicherung könnte dazu führen, dass keine dieser Arbeiten richtig durchgeführt werden.

Prinzip der unabhängigen Qualitätssicherung
Verhältnis zwischen Entwicklung und Qualitätssicherung

Strikte Trennung zwischen Entwicklung und Qualitätssicherung:

Die Entwicklung erstellt Produkte und die Qualitätssicherung ist für die Überprüfung zuständig.

- Entwickler kann sich auf die konstruktiven Aspekte konzentrieren
- Wird aber nicht zu Sorgfalt angehalten
- Realisierung im Ansatz des Pair Programming

Operative Qualitätssicherung wird in die Entwicklung integriert:

Die Entwicklung ist für einen definierten Qualitätszustand ihrer Produkte selbst zuständig. Erst danach setzt die externe Q.-Sicherung ein.

- Klar definierte, transparente Verantwortlichkeiten
- Eigenverantwortlichkeit der Entwickler
- Erfordert messbare Qualitätsstufen und Nachweis, dass sie erreicht wurden

Prinzip der unabhängigen Qualitätssicherung im Lichte der quantitativen Qualitätssicherung

Wird das Prinzip der quantitativen Qualitätssicherung befolgt, so werden viele Prüfparameter instrumentell erfasst.

Personal der Qualitätssicherung kann sich auf Fragen der Interpretation der Messwerte konzentrieren.

In der Qualitätssicherung sind folgende zusätzliche Maßnahmen erforderlich:

- Sammlung von Daten,
- Validierung dieser Daten und
- Einrichten einer quantitativ orientierten Datenbank.

Vorteile einer unabhängigen Qualitätssicherung sind:

- objektive, unabhängige Qualitätssicherung,
- heilsame Wirkung auf Entwicklung und
- Qualitätsvergleiche werden möglich.

Das V-Modell

Erweiterung des Wasserfallmodells um ein integriertes **Qualitäts-Sicherungssystems** zur Durchführung des Qualitätsmanagements.

- genaue Festlegungen zu Verifikation und Validierung von Teilprodukten
- Verifikation: Überprüfung auf Übereinstimmung zwischen Spezifikation und Produkt (Wird ein korrektes Produkt entwickelt?)
- Validierung: Überprüfung der Eignung eines Produkts hinsichtlich seines Einsatzzwecks (Wird das richtige Produkt entwickelt?)

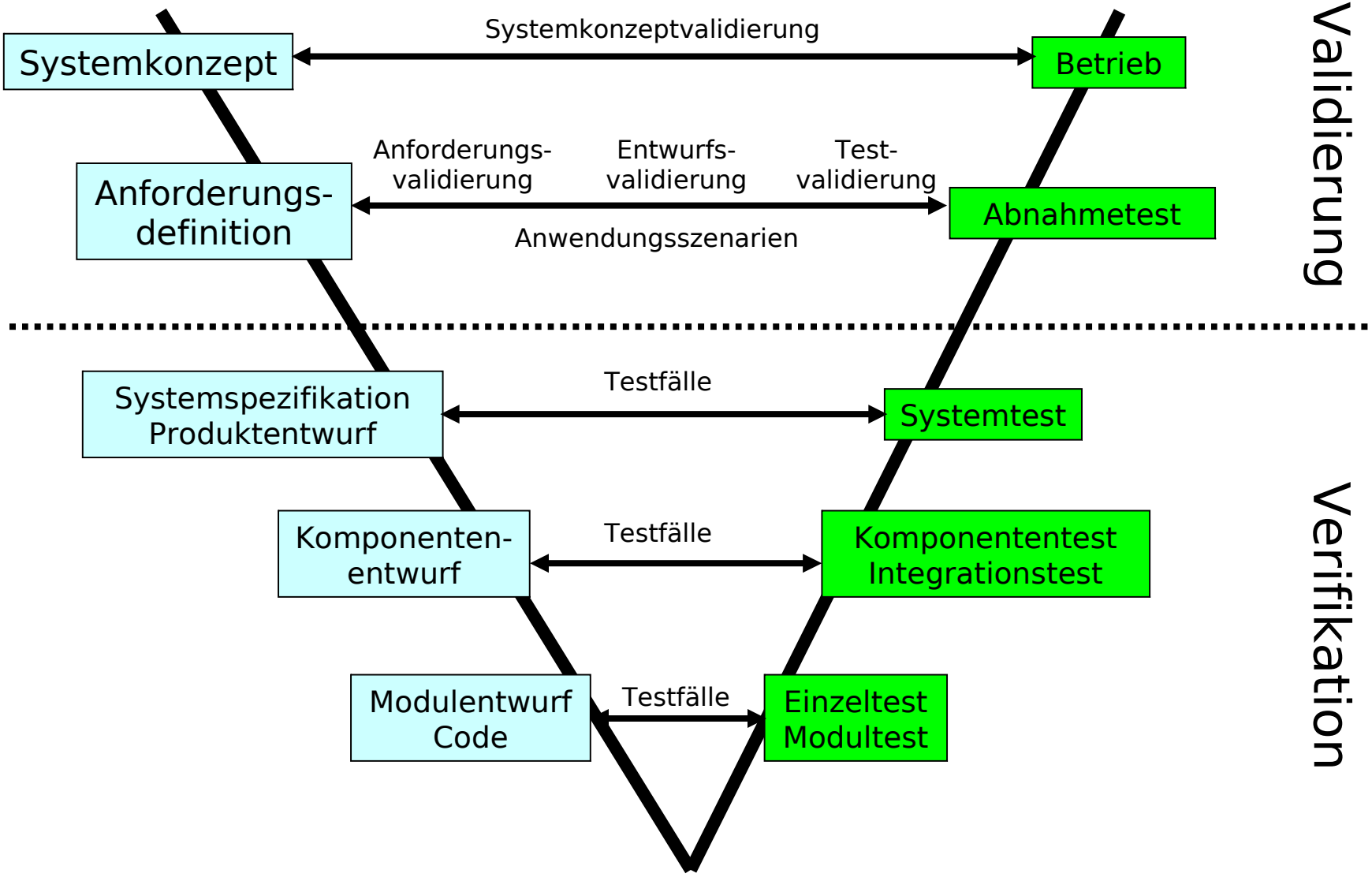
V-Modell ist ein **Vorgehensmodell**. Es gliedert sich in 4 Submodelle:

- System-Entwicklung (SE)
- Qualitätssicherung (QS)
- Konfigurationsmanagement (KM) und
- Projektmanagement (PM)

Für uns ist der Teil **Qualitätssicherung** wichtig**.**

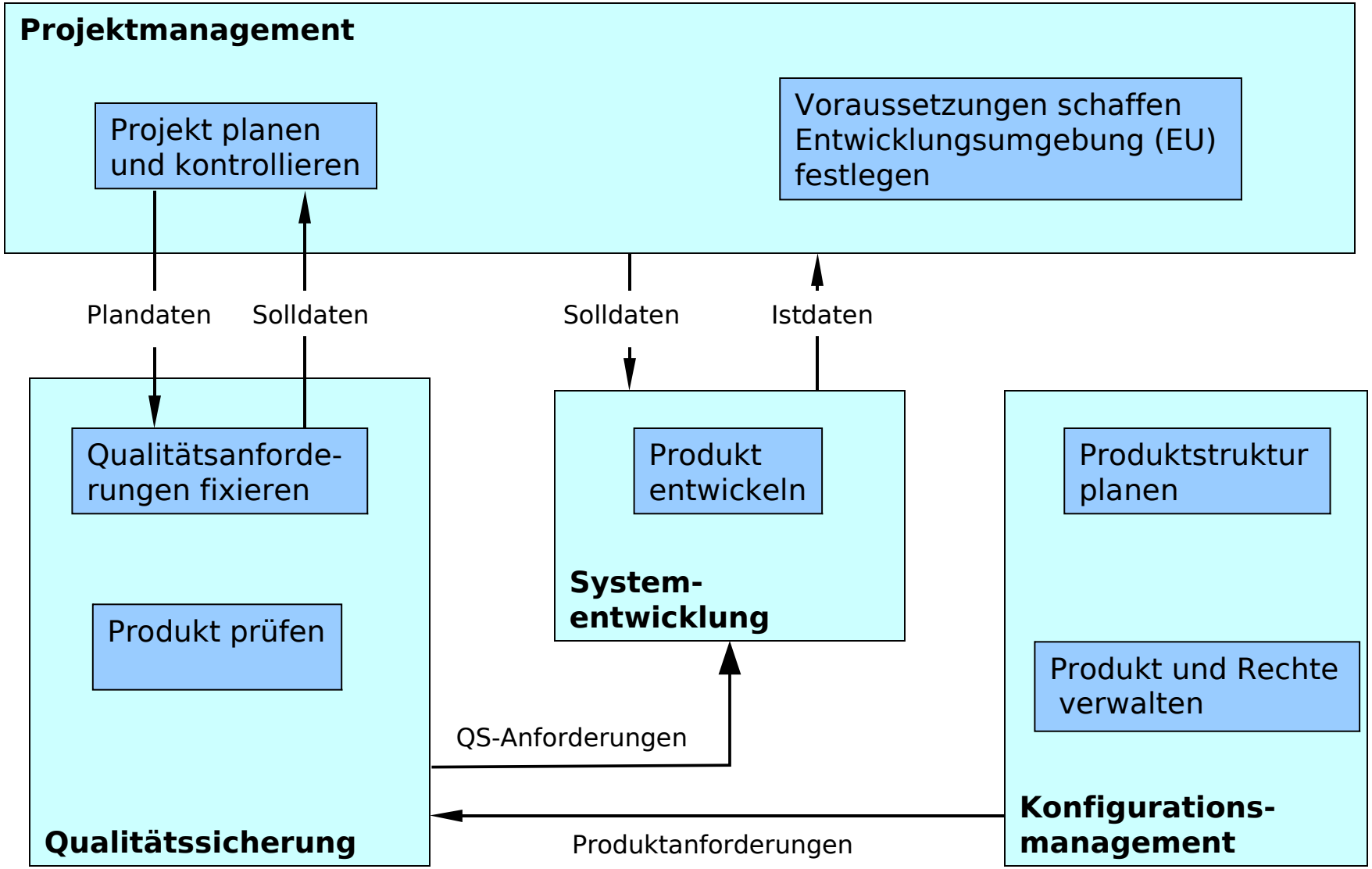
6. Beispiel: Qualitätssicherung im V-Modell

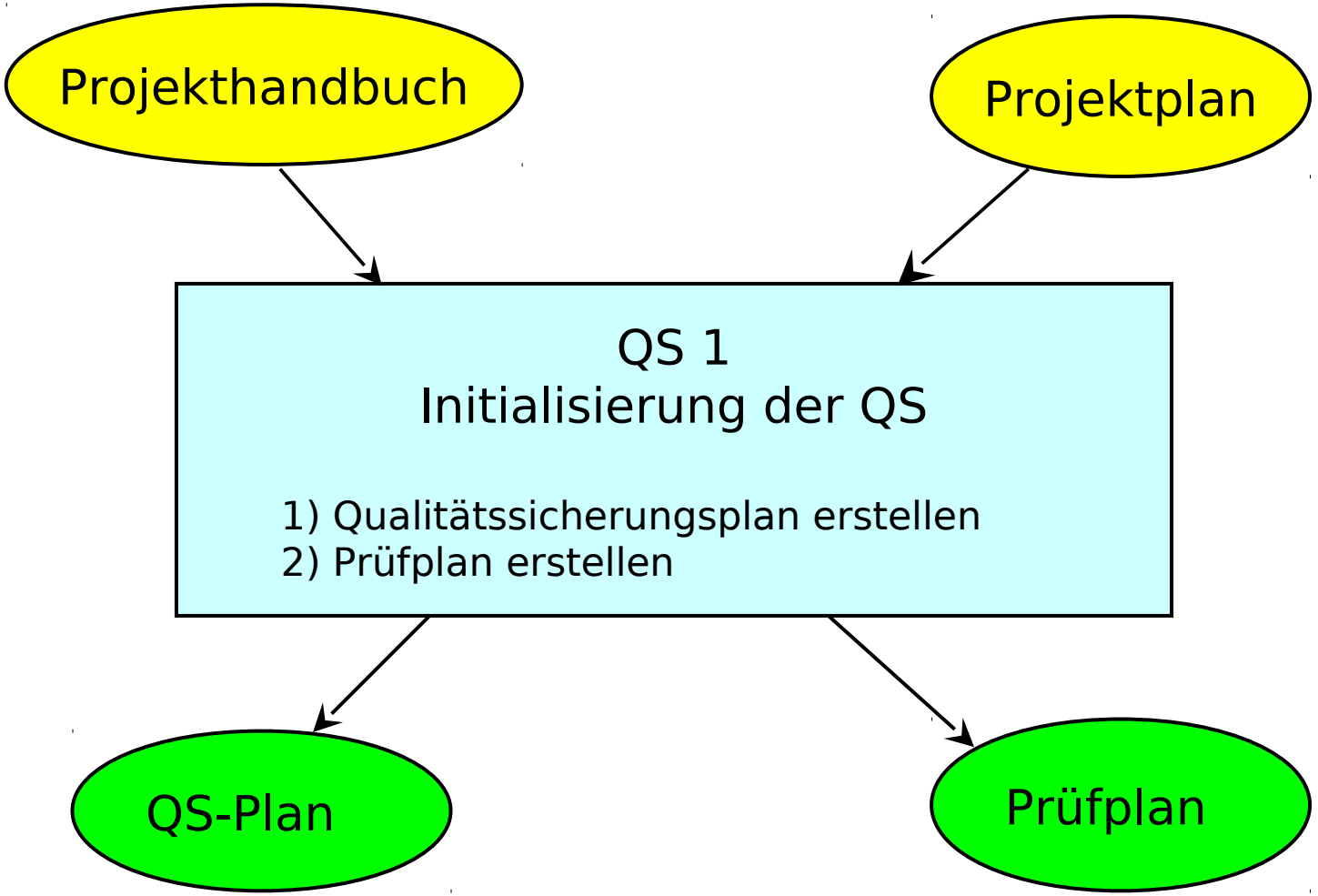
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



6. Beispiel: Qualitätssicherung im V-Modell

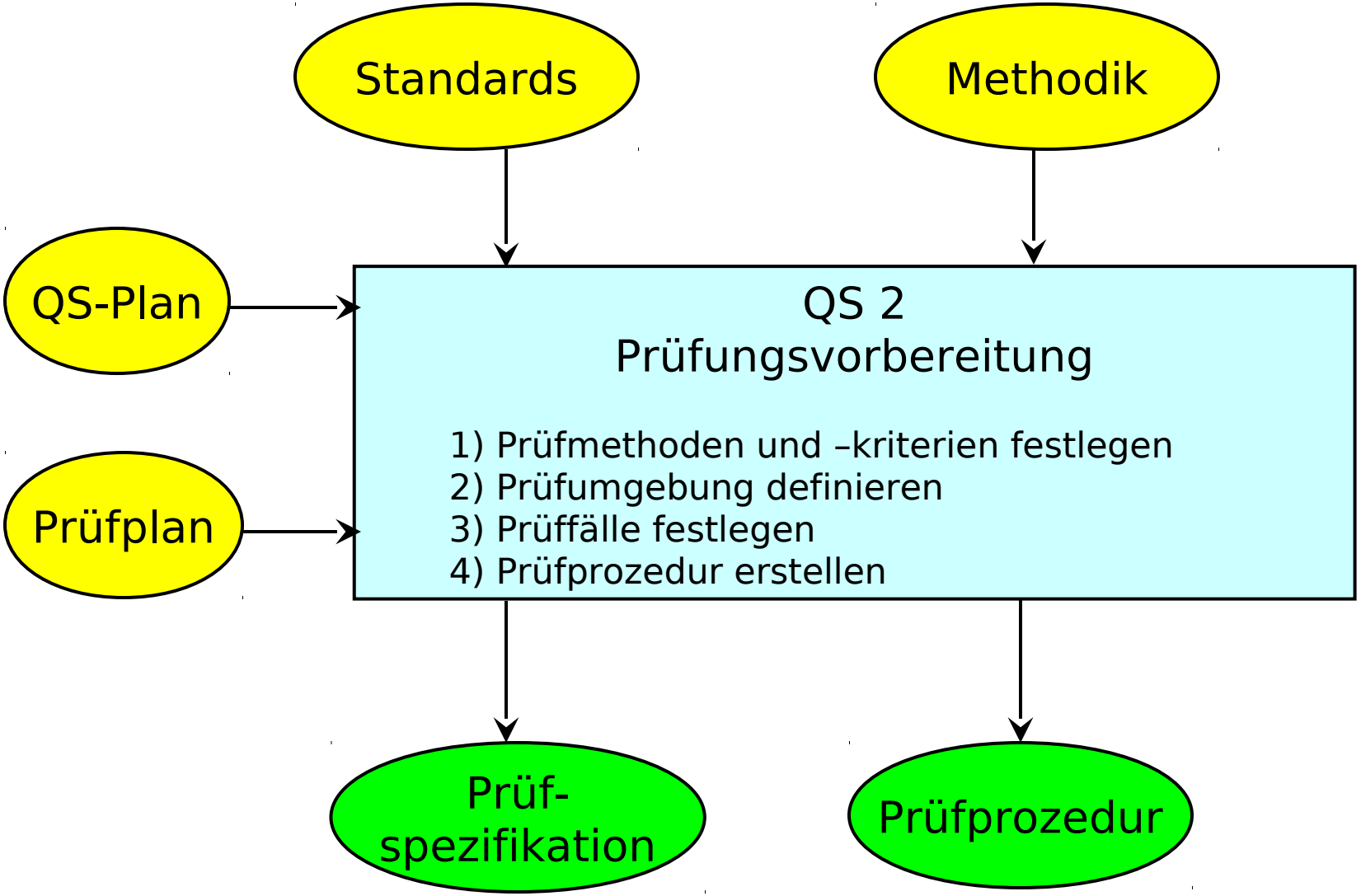
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme




Projekthandbuch
Projektplan
QS 1
Initialisierung der QS
1) Qualitätssicherungsplan erstellen
2) Prüfplan erstellen
QS-Plan
Prüfplan

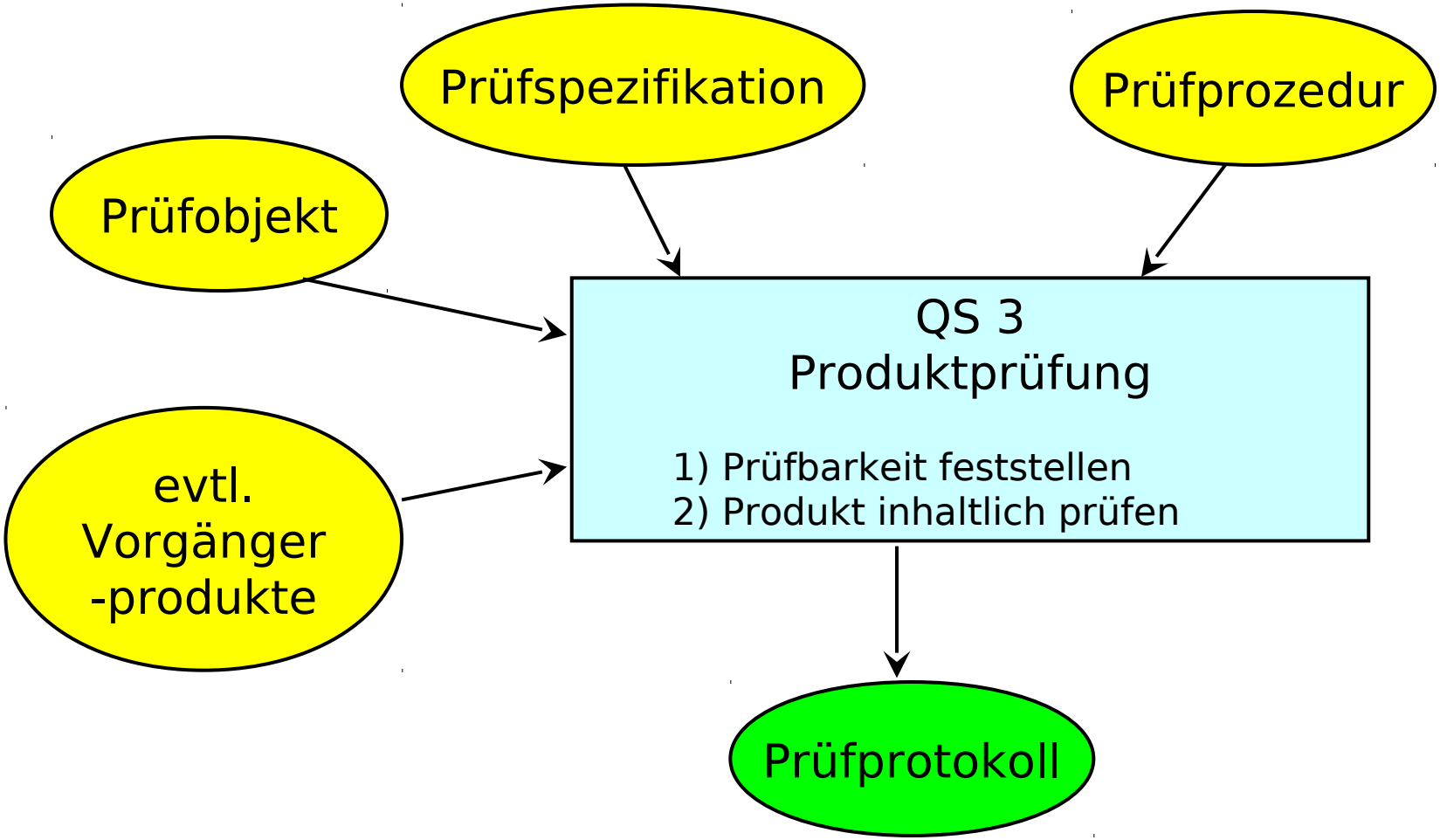
6. Beispiel: Qualitätssicherung im V-Modell

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



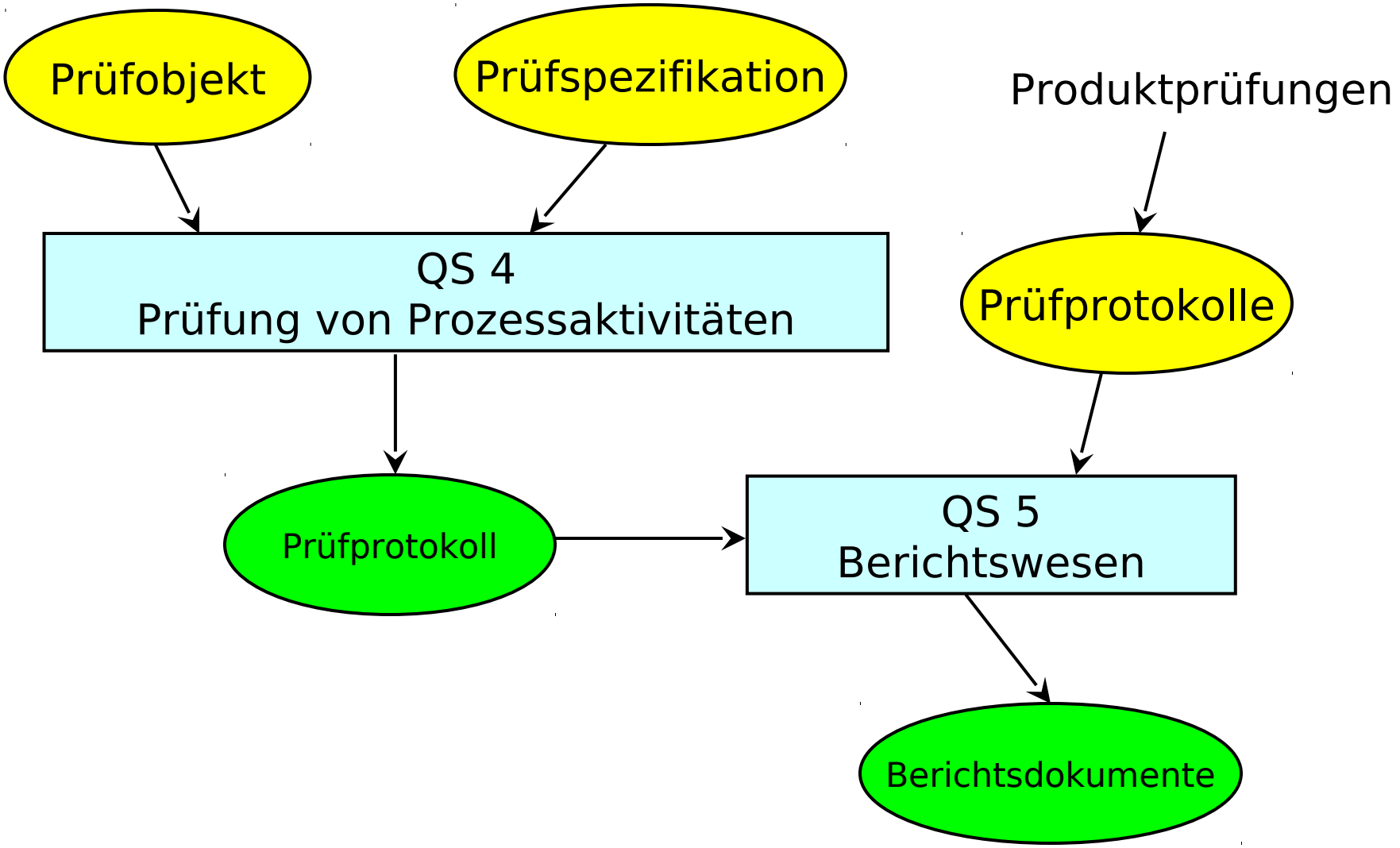
6. Beispiel: Qualitätssicherung im V-Modell

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



6. Beispiel: Qualitätssicherung im V-Modell

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



Zusammenfassung

- Festlegung und Sicherung von Qualitätsanforderungen ist eine erstrangige Managementaufgabe, welche in einem **Qualitäts-Managementprozess** durch ein Bündel von **Qualitäts-Sicherungsmaßnahmen** auf der Basis eines **Qualitätsplans** operationalisiert wird und sich auf Produkte und/oder Prozesse beziehen können.
- **Konstruktive QM** sorgen dafür, dass das Produkt gewisse Eigenschaften a priori besitzt
- **Analytische QM** messen das existierende Q.-Niveau und identifizieren Ausmaß und Ort von Defekten
- Alle Maßnahmen des **QM** werden in einem (prozess-orientierten) **QS-Plan** fixiert, welcher durch einen (produkt-orientierten) **Prüfplan** untersetzt ist.
- Auf der Managementebene bildet ein **Qualitätsmanagement-System** den Rahmen für alle qualitätssichernden Maßnahmen und Strategien.

Qualitätsmanagement erfordert Aktivitäten in folgenden Bereichen:

- Q.-**Planung**
  - Initiale Festlegung von projektbezogenen Q.-Anforderungen in überprüfbarer Form
  - Orientierung an den sechs Grundprinzipien der QS in Softwareprojekten
- Q.-**Lenkung** (prozessbezogene QS)
  - Überwachung und Steuerung des Entwicklungsprozesses mit dem Ziel, die vorgegebenen Q.-Anforderungen zu erfüllen
  - Nachweisführung über den Erfüllungsstand von Q.-Anforderungen
- und Q.-**Prüfung** (produktbezogene QS)
  - Erfassung der Ist-Parameter der Q.-Indikatoren entsprechend der Q.-Planung sowie Kontrolle der Einhaltung der konstruktiven QM