

Modellbasiertes Testen auf Basis des fundamentalen Testprozesses

Tobias Eckardt, Michael Spijkerman

Software Quality Lab (s-lab)
Universität Paderborn

12. Februar 2009



Motivation

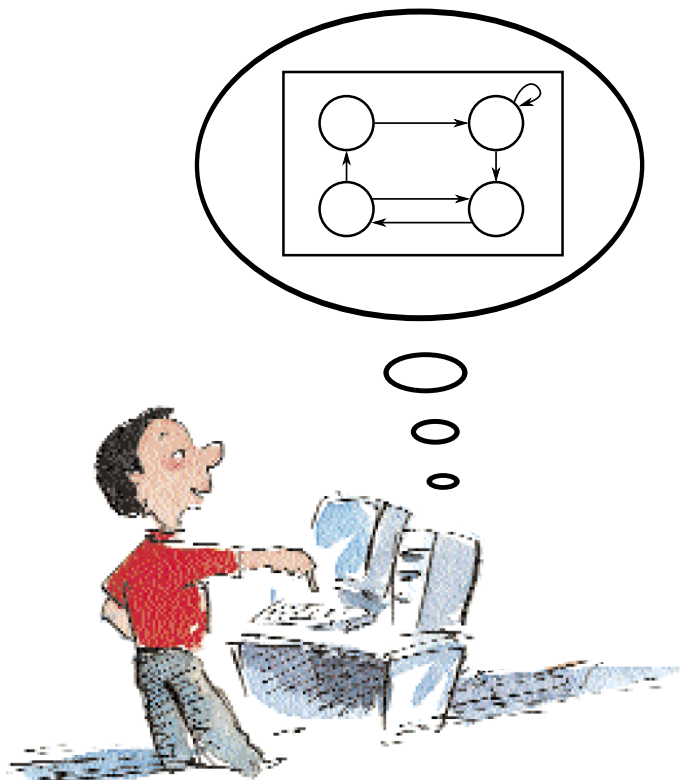
Manuelles Testen

Modellbasiertes Testen

Gegenüberstellung

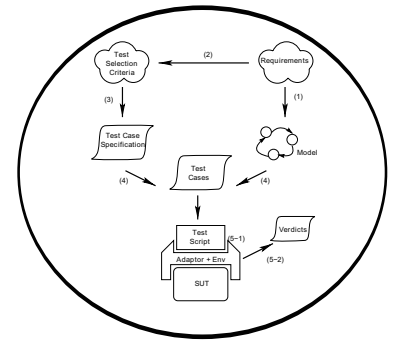
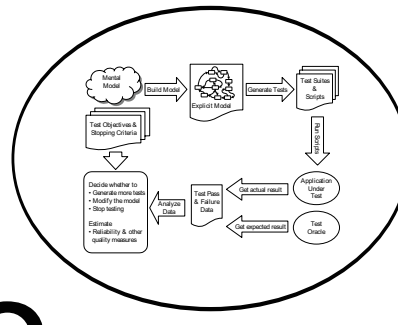
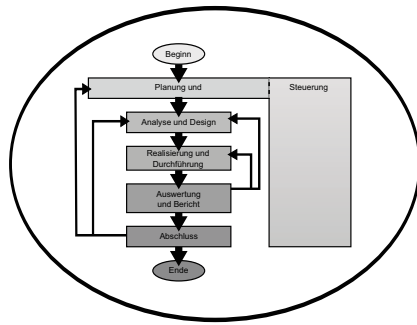
Zusammenfassung

Vorgehensmodell für Modellbasiertes Testen

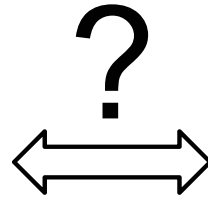


Quelle: [Kla05] / [Rob00]

Vorgehensmodell für Modellbasiertes Testen

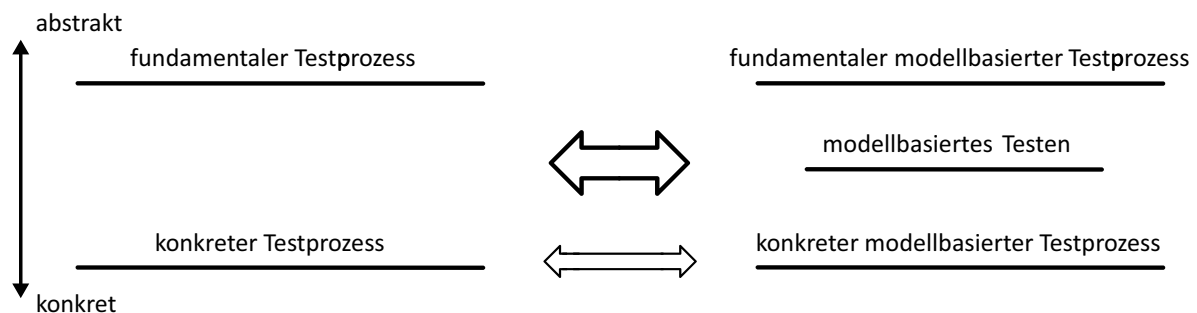


vgl. [SL05]

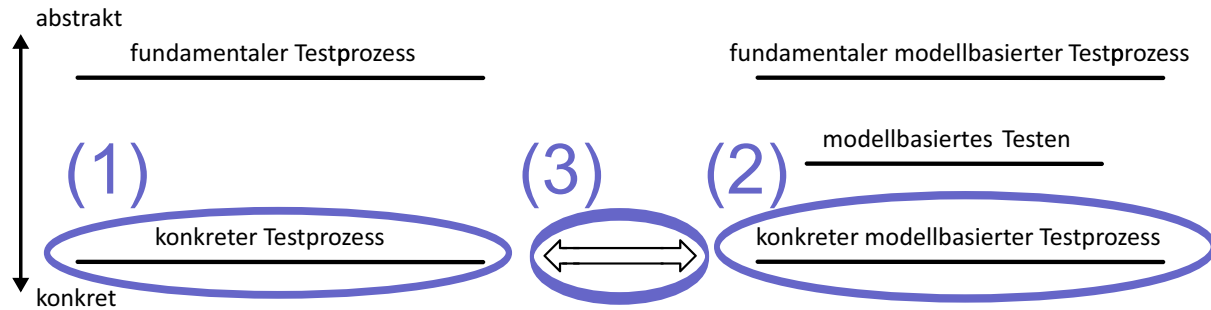


vgl. [EFW01], [UPL06]

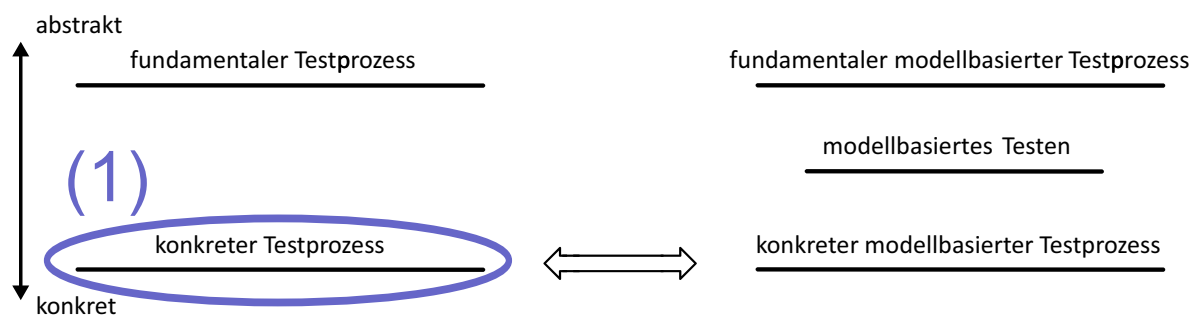
Konzept



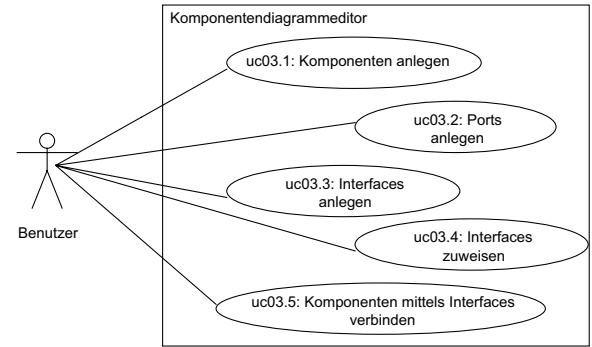
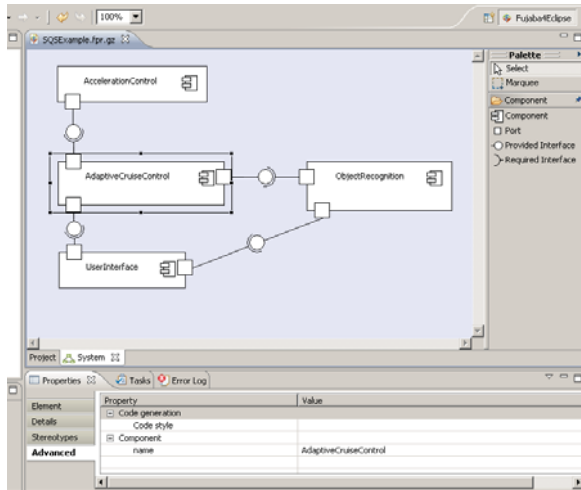
Konzept



Manuelles Testen am Beispiel



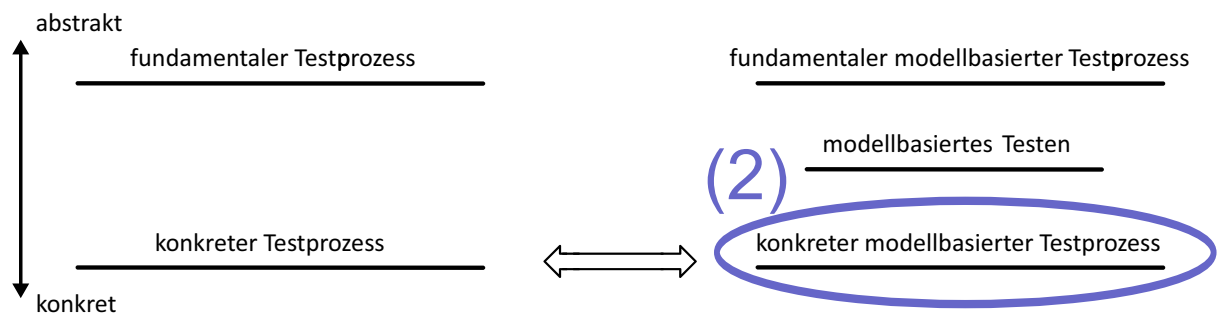
Manuelles Testen am Beispiel



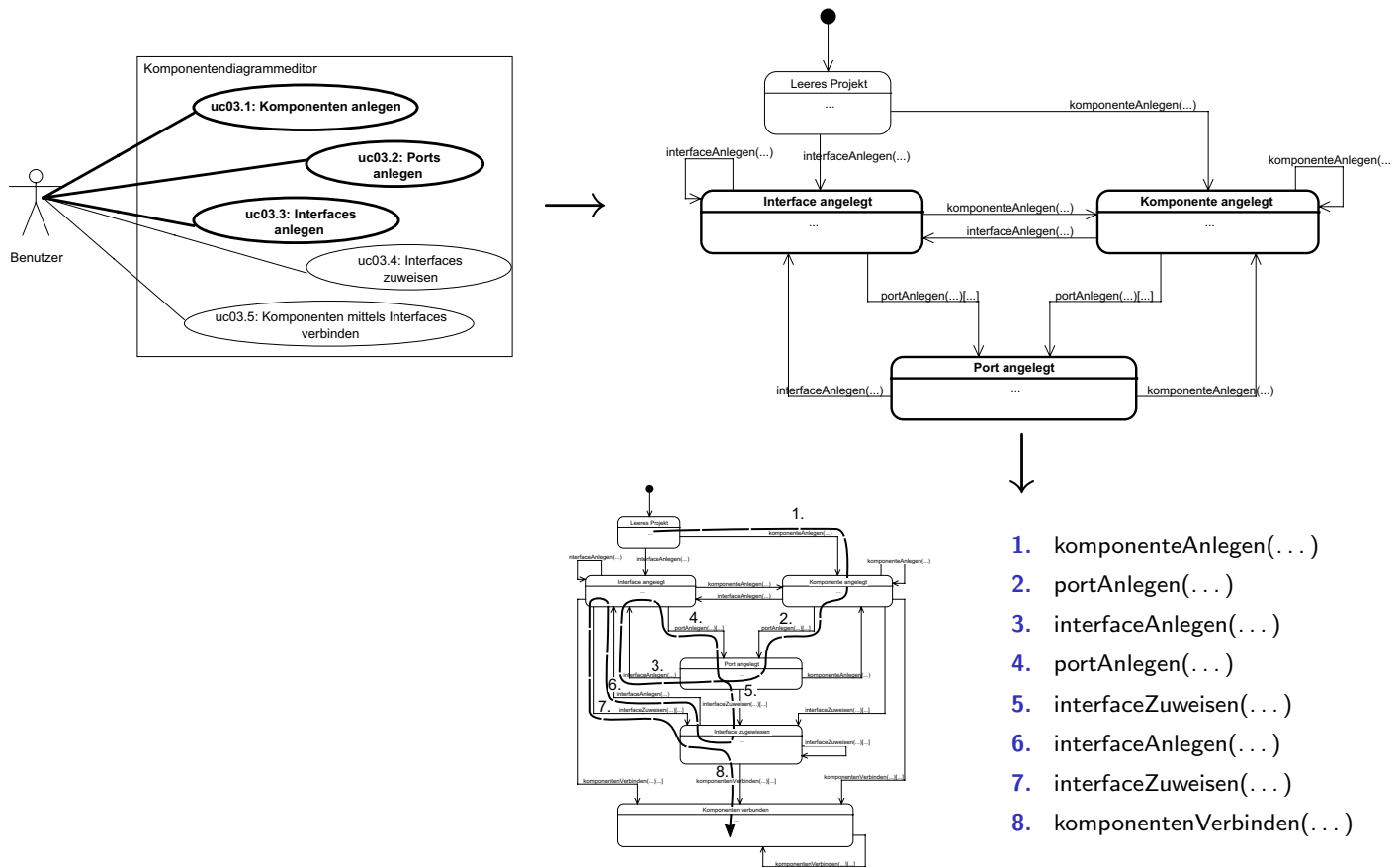
Schritt	Aktion	Erwartetes Ergebnis
1	Erstelle ein Komponentendiagramm mit Namen Test04 und lege einen Port auf der Komponente an.	Das Projekt enthält ein Komponentendiagramm mit Namen Test04 und die darin enthaltene Komponente enthält einen Port.
2	Klicke im Project Explorer mit der rechten Maustaste auf das Komponentendiagramm Test04 und wähle „Löschen“ aus dem Kontextmenü.	Es öffnet sich ein Dialog, der das Statechart der Komponente, das Klassendiagramm zu der Komponente und das Protokollstatechart des Port zur Auswahl anbietet.
3	Wähle alle Diagramme aus dem Dialog aus und klicke OK.	Das Dialogfenster schließt sich und das Komponentendiagramm, sowie die drei ausgewählten Diagramme werden aus dem Projekt entfernt.

Testfall: „Komponente Löschen“

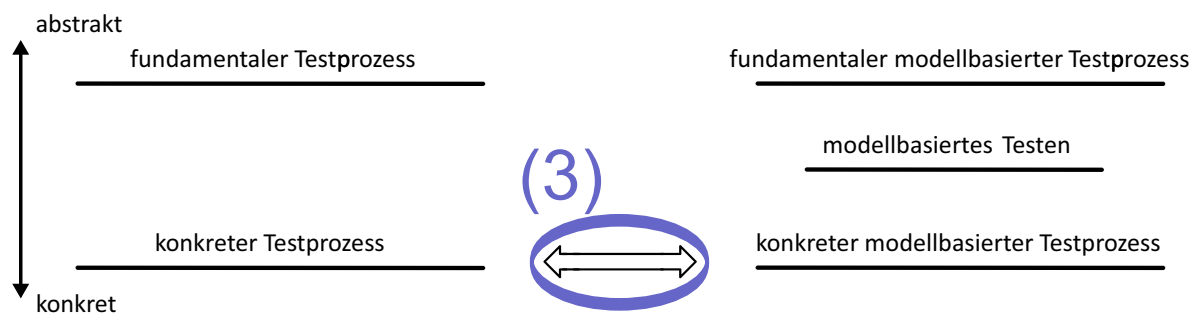
Modellbasiertes Testen am Beispiel



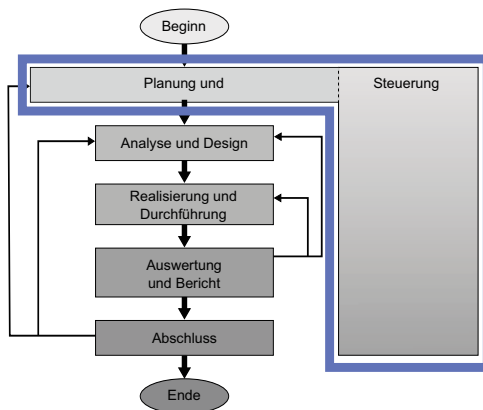
Modellbasiertes Testen am Beispiel



Gegenüberstellung mit dem Fundamentalen Testprozess

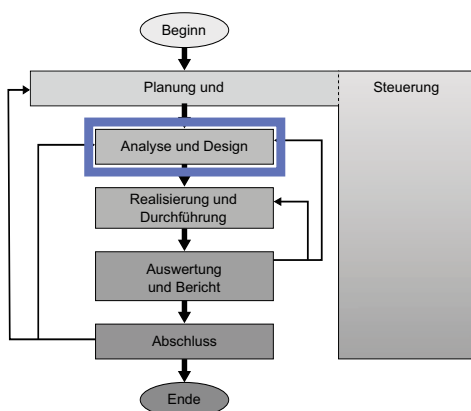


Planung und Steuerung



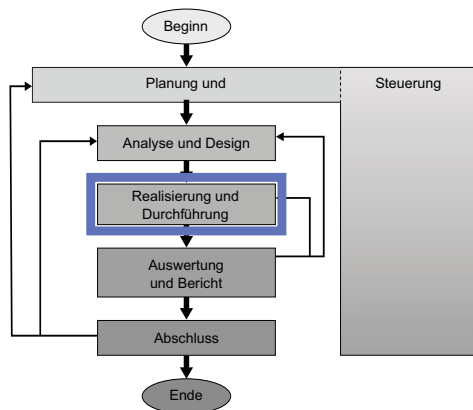
- ▶ Ressourcenplanung: Mehr Zeit/Mitarbeiter/Werkzeuge für Modellerstellung, weniger für Testfallerstellung
- ▶ Testmanagement: zusätzl. Planung/Werkzeuge zur Verwaltung des Modells
- ▶ Teststrategie: Verankerung im Modell
- ▶ Steuerung: Überwachung des Modells

Analyse und Design



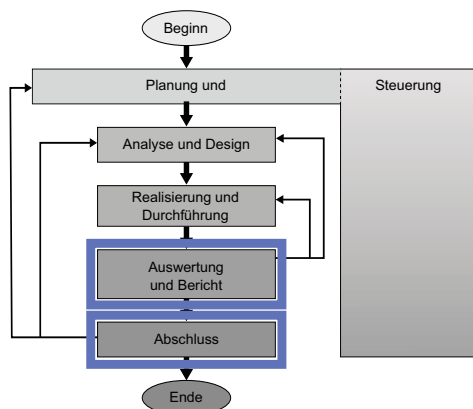
- ▶ Analyse der Testbasis
 - ▶ Modelltyp passend zu gewünschten Testfällen (z.B. bzgl. Abstraktionsgrad)
- ▶ Testplan: Mehr Zeit für das Modell, weniger für die Testfälle
- ▶ Kein manuelles Erstellen logischer Testfälle → Erstellung des Modells:
 - ▶ Integration des Testorakels
 - ▶ Testfälle für unerwartete Eingaben
 - ▶ Erstellung des Algorithmus zur Testfallgenerierung
 - ▶ Generierung logischer Testfälle

Realisierung und Durchführung



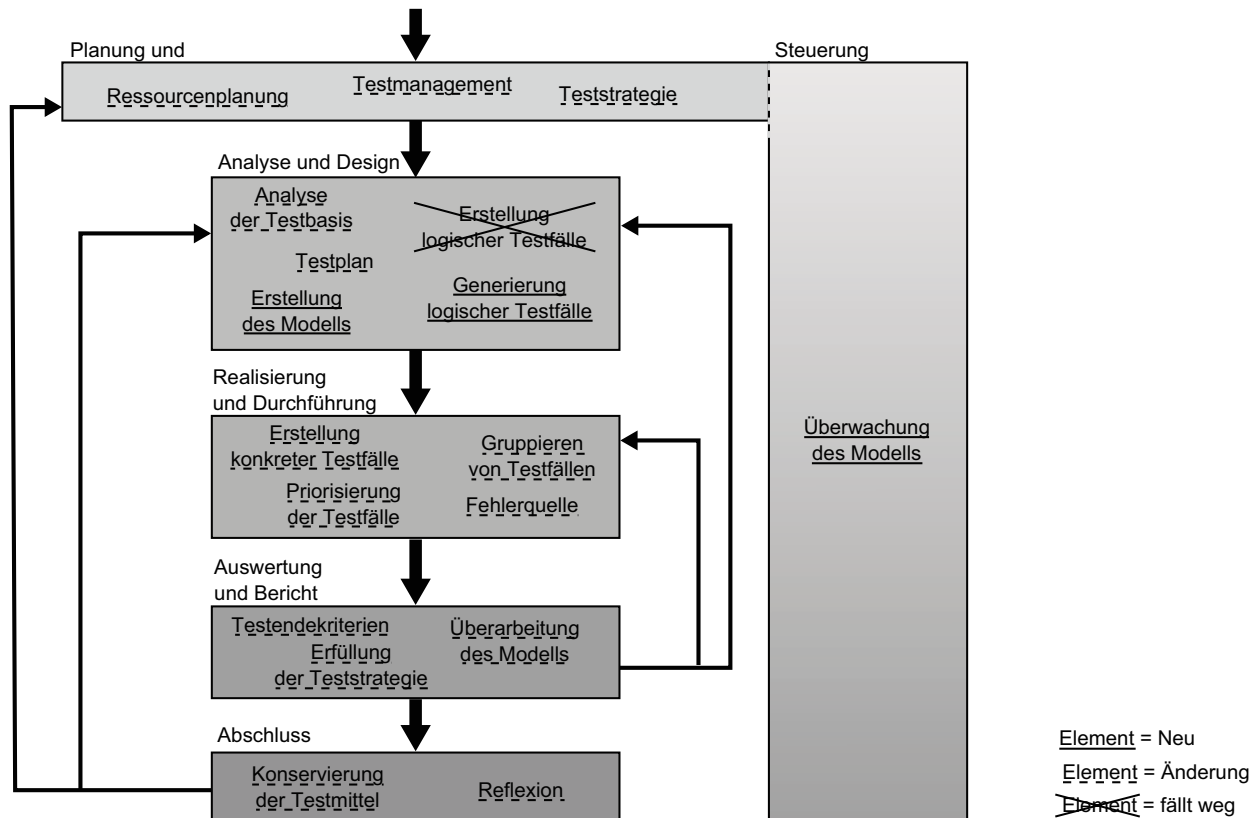
- ▶ Berücksichtigung von Prioritäten (eventl. im Modell/Algorithmuss integriert)
- ▶ Konkretisierung der Testfälle eventl. nicht mehr notwendig
- ▶ Gruppierung von Tests automatisch
- ▶ (Durchführung)
- ▶ Fehler gefunden:
 - ▶ Fehlerquelle im Modell?
 - ▶ Klassifizierung über das Modell möglich?

Auswertung und Bericht / Abschluss



- ▶ Testendekriterien: Neue Möglichkeiten auf Basis des Modells
- ▶ Passt die Testfallgenerierung zur Teststrategie? Eventl. Anpassung des Algorithmus
- ▶ Reflexion mit Einbezug des Modells
- ▶ Konservierung des Modells und entsprechender Werkzeuge

Zusammenfassung

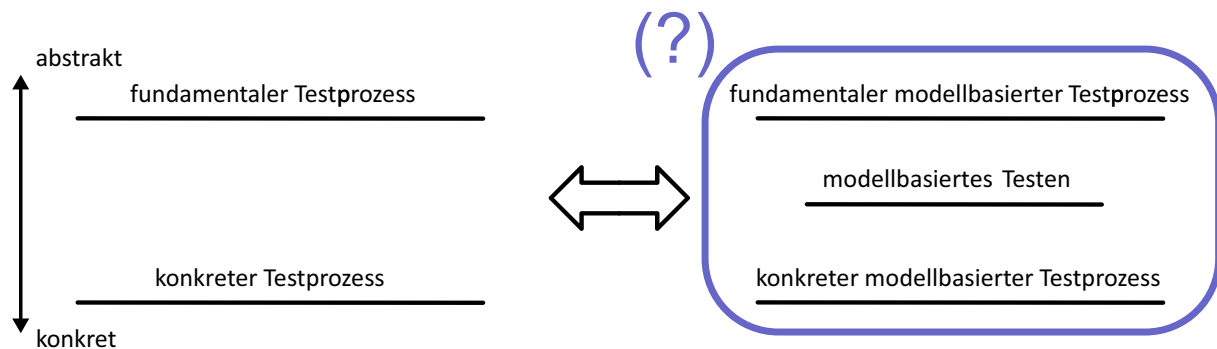


Fazit

Zusammenfassung:

- ▶ Viele Konkretisierungen und sogar einige Änderungen gegenüber dem FTP (bereits bei Analyse eines Fallbeispiels)
- ▶ Schwerpunkt der Änderungen: Analyse und Durchführung
- ▶ Aber auch in umgebenden Phasen (Planung, Steuerung, Auswertung und Abschluss) Änderungen/Konkretisierungen vorhanden

Ausblick



Folgerungen/Ausblick:

- ▶ Weitere Analysen sinnvoll und notwendig
- ▶ Definition eines fundamentalen modellbasierten Testprozesses
- ▶ Folgeprojekte in größerem Umfang und industrinahem Kontext

Quellen & Literatur I

- [EFW01] EL-FAR, Ibrahim K. ; WHITTAKER, James A.:
Model-based Software Testing.
In: *Encyclopedia on Software Engineering* (2001)
- [Kla05] KLAPDOHR, Christiane:
Modellbasiertes Testen. Ausarbeitung zum Proseminar
„Präsentation ausgewählter Problemstellungen der
Informatik“.
Universität Paderborn, 2005
- [Rob00] ROBINSON, Harry:
Intelligent Test Automation.
In: *Software Testing & Quality Engineering (STQE)*
(2000), September/Oktober, S. 24–32

Quellen & Literatur II

[SL05] *Kapitel 2.2 Fundamentaler Testprozess.*

In: SPILLNER, Andreas ; LINZ, Tilo:

Basiswissen Softwaretest: Aus- und Weiterbildung zum Certified Tester; Foundation Level nach ISTQB-Standard.

dpunkt-Verl., Heidelberg, 2005, S. 18–32

[UPL06] UTTING, Mark ; PRETSCHNER, Alexander ; LEGEARD, Bruno:

A taxonomy of model-based testing.

(2006), April, Nr. 04/2006