

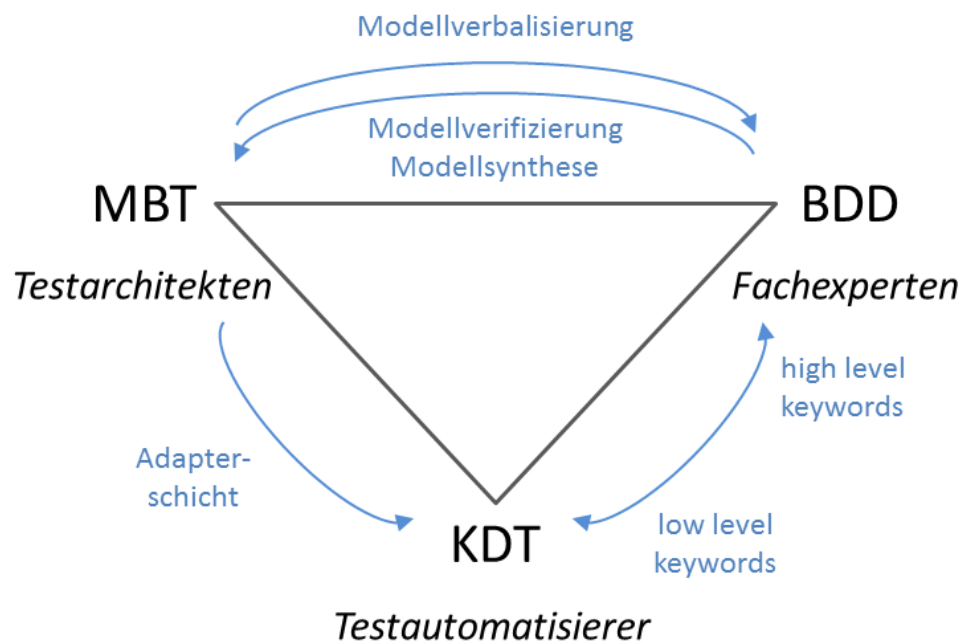
Protokoll AK MBT 10. November 2017 in Ratingen

Teilnehmer

Name
Dehla Sokenou (Protokoll)
Mario Friske
Benedikt Eberhardinger
Alexander Richter
Andrej Pietschker
Dierk Ehmke
Karsten Dörges
Wenjin Zhu

Kurzbericht aus dem AK vom Leitungsgremium

Die AK-Leitung berichtet kurz zu den Themen, die seit dem letzten Treffen auf der TAV 40 im Februar 2017 bearbeitet wurden (siehe Folien). Dabei ging es vorrangig um die Weiterarbeit an der Kombination der drei Techniken BDD, MBT und KDT: der Arbeitskreis hat das erarbeitete Dreieck durch selbst erstellte und von anderen Testexperten erarbeitete Stacks weiter verfeinert und die drei Techniken Layern in den Stacks zugeordnet. Die Stacks dienen zudem zur Untermauerung der Thesen aus dem Dreieck, das verschiedene Zusammenhänge und Kombinationsmöglichkeiten für die drei Techniken vorschlägt.



Diskussion und Gruppenarbeit

Ziel des heutigen Workshops ist die Aufstellung weiterer Stacks und die Klärung offener Fragen. Diese sind unter anderem:

- Welche Stacks kann es geben / gibt es (aus eigener Erfahrung)?
- Enthalten diese bereits MBT / BDD / KDT?
- Wo wurde/könnte MBT / BDD / KDT in den Stacks eingesetzt werden, mit welchem Nutzen?
- Gibt es erkennbare Patterns in den Stacks?
- Lassen sich die Techniken immer eindeutig dem technischen oder dem fachlichen Layer zuordnen?

Zunächst wird diskutiert, ob bestimmte Techniken für bestimmte Zielgruppen besonders geeignet sind, unter anderem BDD für Fachexperten. Darüber gibt es eine kontroverse Diskussion, da von einigen Teilnehmern berichtet wird, dass BDD auf Akzeptanzprobleme bei Fachexperten trifft. Es wird berichtet, dass bei BDD die Intention des Tests leicht verloren gehen kann und dass BDD auch quasi wie KDT eingesetzt werden kann – hier wird insbesondere eine vordefinierte Steplibrary als kontraproduktiv angesehen. Als Vorteile von BDD werden insbesondere die Möglichkeit der nachgelagert erfolgenden Automatisierung, die Zusammenfassung der gesamten Vorbedingung in einem kurzen Statement und der höhere Abstraktionsgrad genannt. Alle diese Erfahrungsberichte stützen die These aus dem Dreieck, dass es sich bei BDD um eine Art Keyword-Driven Testing handelt, nur auf einem höheren Level. Für die richtige Anwendung sind jedoch klare Regeln erforderlich, um typische Fallstricke bei der Anwendung der Technik zu vermeiden.

Für die Kante MBT / KDT, in der KDT als Automatisierungswerkzeug für MBT dient, wird das Werkzeug TEDES0 genannt, das MBT auf Keywords abbilden kann.

KDT ist viel stärker an der eigentlichen Implementierung orientiert, so dass auch hier die Intention des Tests ggf. verloren gehen kann. Eine Steplibrary kann als Set of Actions verstanden werden, aus der sich durch eine Sequenz von Schritten ein modulares Szenario definieren lässt. Es stellt sich die Frage, ob es einen Mehrwert von BDD gibt, wenn KDT bereits produktiv und erfolgreich im Einsatz ist.

Für die Verbindung zwischen MBT und BDD wird als Hauptgrund eine leichtere Spezifikation und ein besseres Verständnis der Modelle genannt. Eine lesbare Spezifikation auf Basis von BDD ließe sich durch MBT besser automatisieren. Zudem lassen sich Use-Cases aus dem Modell nicht nur in Aktivitäten oder Sequenzen verfeinern, sondern auch in BDD-Szenarien. Wenn das Modell verstanden wird, wollen die Tester jedoch aus der Erfahrung der Teilnehmer mit dem Modell weiterarbeiten.

Der Unterschied zwischen Sequenzdiagrammen und BDD wird anschließend diskutiert. Dabei werden Fragen aufgeworfen wie

- Sind Sequenzen Szenarien ohne Given und When-Part?
- Ist BDD eher Higher-Level?
- Kann man BDD auf Sequenzdiagramme abbilden? Oder als State-Machine?
- Kann man BDD abbilden auf das Modell bzw. das Modell aus BDD-Szenarien synthetisieren?

Als nächstes wird die Skalierbarkeit der Techniken betrachtet. Dabei wird unterschieden zwischen technischer und fachlicher Skalierbarkeit:

- Skalierbarkeit im Hinblick auf fachliche Änderungen
 - Bei MBT muss die Änderung nur im Modell gepflegt werden.
 - Bei BDD und KDT sollte man auf Techniken zur Wiederverwendung setzen, dies hängt aber auch vom Abstraktionsgrad der Testsuite ab.

- Technische Skalierbarkeit:
 - Erfahrung mit großen Modellen sind initial oft sehr gut, doch kann bspw. Parallelität in großen Modellen die Testfallgenerierung sprengen oder die Terminierung der Testfallgenerierung in endlicher Zeit verhindern.
 - Oft handelt es sich dabei aber um Schwierigkeiten der Testtools und ist sehr stark abhängig von Teststrategie, Tool und Modellabstraktion.
 - Als eine Beobachtung wurde genannt, dass State-Machines schwieriger als MSCs skalieren, wobei die Frage nach der Allgemeingültigkeit (strengere Ordnung skaliert schlechter?) in den Raum gestellt wird.

Die wichtigsten Einsichten zur Skalierbarkeit der Techniken sind:

- Aus der Erfahrung der Teilnehmer skalieren BDD-Testsuiten in jeder Richtung, das Problem ist hier die Testausführung.
- KDT-Testsuiten skalieren theoretisch gut, aber je nach Abarbeitung der Testsuite und Chunking der Tests ist Testausführung das Problem.
- Ein gemeinsames Problem von KDT und BDD ist die Anzahl der verwendeten Keywords.
- Bei Kombinationen muss darauf geachtet werden, dass man nicht ein doppeltes Bottleneck bekommt.
- Bei MBT hängt die Skalierbarkeit insbesondere von den Modellen und ihrer Größe ab.

Ein weiterer wichtiger Aspekt ist die Traceability, welche das Tool explizit unterstützen muss:

- Bei MBT ist dies meist schon in irgendeiner Form im Tool vorhanden (z.B. Conformiq).
- Beim Team-Foundation-Server ist die Abbildung von Epic auf die BDD-Szenarien gegeben.
- Bei KDT ist Traceability in der Regel selbst zu implementierten (Beispiel: Tracing von Anforderung zu Testfall, zurück nur durch Annotation am Testfall).

Als ein Fazit lässt sich herausarbeiten, dass unabhängig von der verwendeten Methode das Tooling für die Akzeptanz beim Nutzer entscheidend ist.

Stacks

Neben den von der AK-Leitung vorgestellten Stacks aus verschiedenen Quellen (siehe Folien) werden weitere Stacks in gemeinsamer Arbeit im Workshop erstellt und darüber diskutiert. Im Folgenden 3 Fotos mit Ergebnissen.

Teststeps

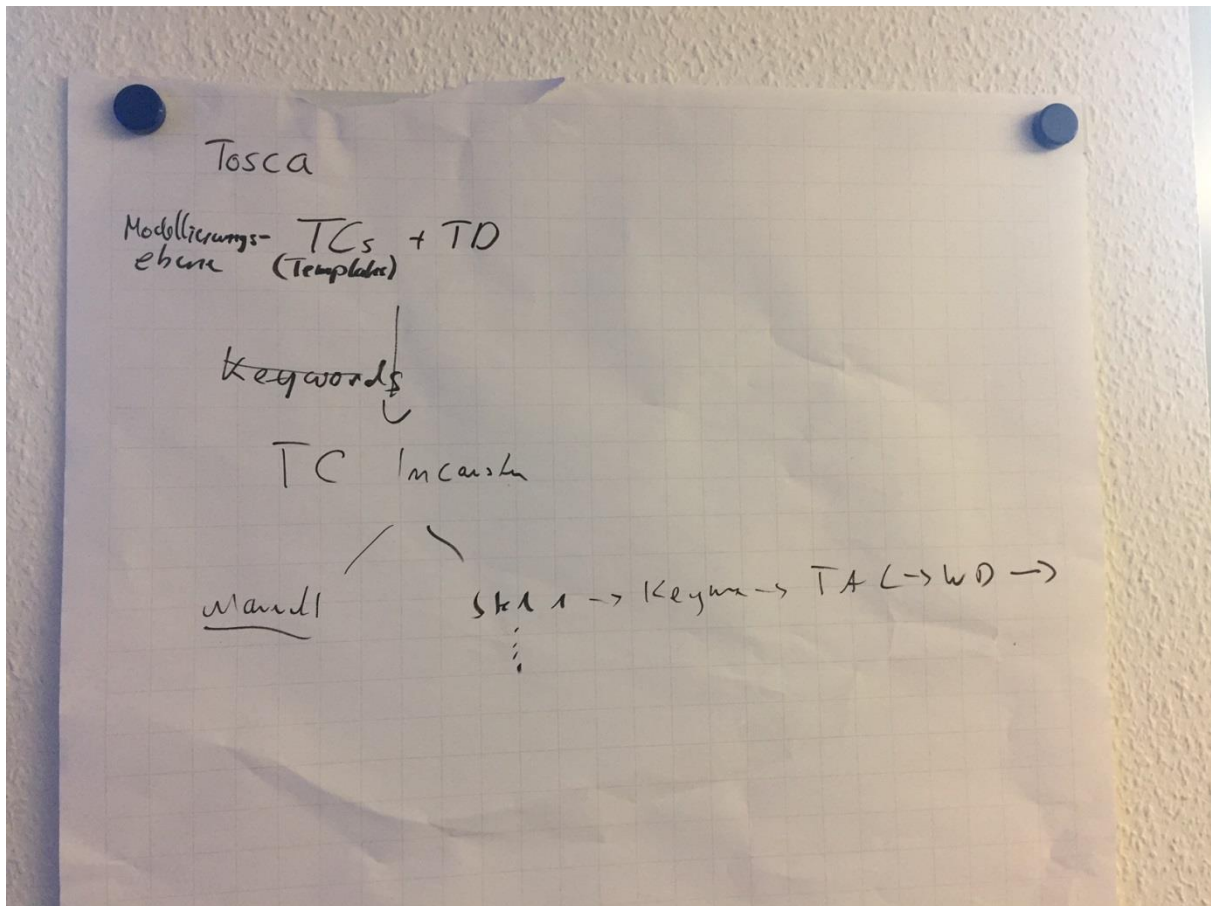
Mapping Teststep \rightarrow Fixture $\begin{matrix} TS_x \rightarrow F_x \\ TS_y \rightarrow F_x \end{matrix}$

attr. [' (new|alt)']

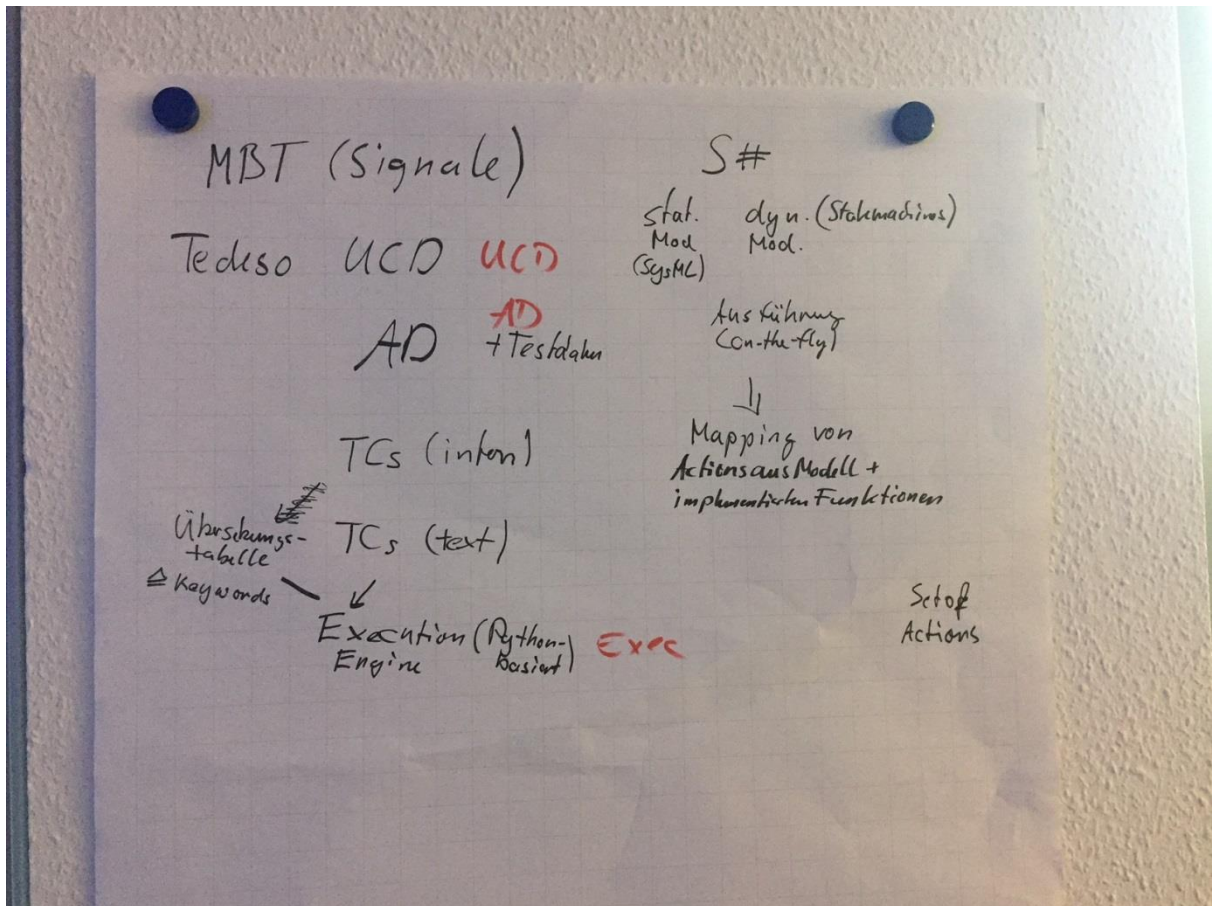
Fixtures (StepLibrary) (string per)

{
-
}

Das erste Foto zeigt die Abbildung von Teststeps (fachlich) auf Fixtures, wie sie technisch in KDT-Frameworks zur Bindung des Testcodes an die zu testende Anwendung definiert werden. Dabei wird ein Teststep durch eine Fixture realisiert. Durch dieses Mapping entsteht eine Steplibrary, die sowohl auf fachlicher als auch auf technischer Ebene definiert ist.



Das zweite Foto zeigt die Abbildung von MBT-Testfällen auf Steps und schließlich Keywords, die mit Hilfe eines Testausführungslayers (TAL) umgesetzt sind. Dies stützt die These aus dem Dreieck, dass KDT sich als Adapterschicht für MBT eignet.



Das dritte Foto stützt diese These weiter, hier werden Modelle für den Modellbasierten Test auf Aktionen gemappt, so dass schließlich ein Set von Aktionen entsteht. Die Modelle bestehen in diesem Fall aus Use-Cases-Diagrammen (UCD), Aktivitätsdiagrammen (AD) sowie in S# definierten Zustandsautomaten.

Die nächsten Schritte des Arbeitskreises sind vertiefende Diskussionen und eine weitergehende Interpretation der erstellten Stacks.