

„... wer soll denn das alles testen...“

TESTING FRAMEWORK IN DER PRAXIS

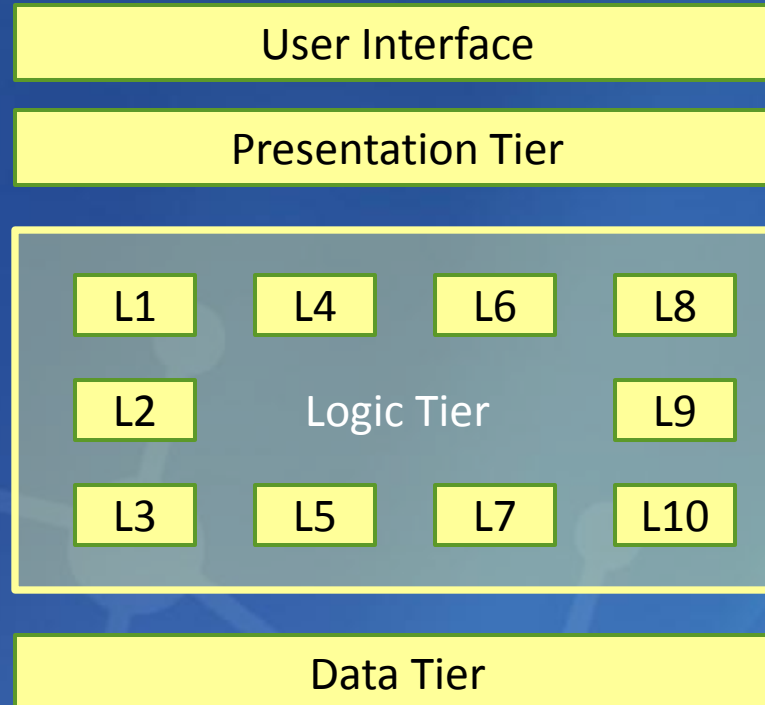
Jonas Bandi Software Architect, TechTalk

Alexander Schramek Quality Manager, Senior Project Manger

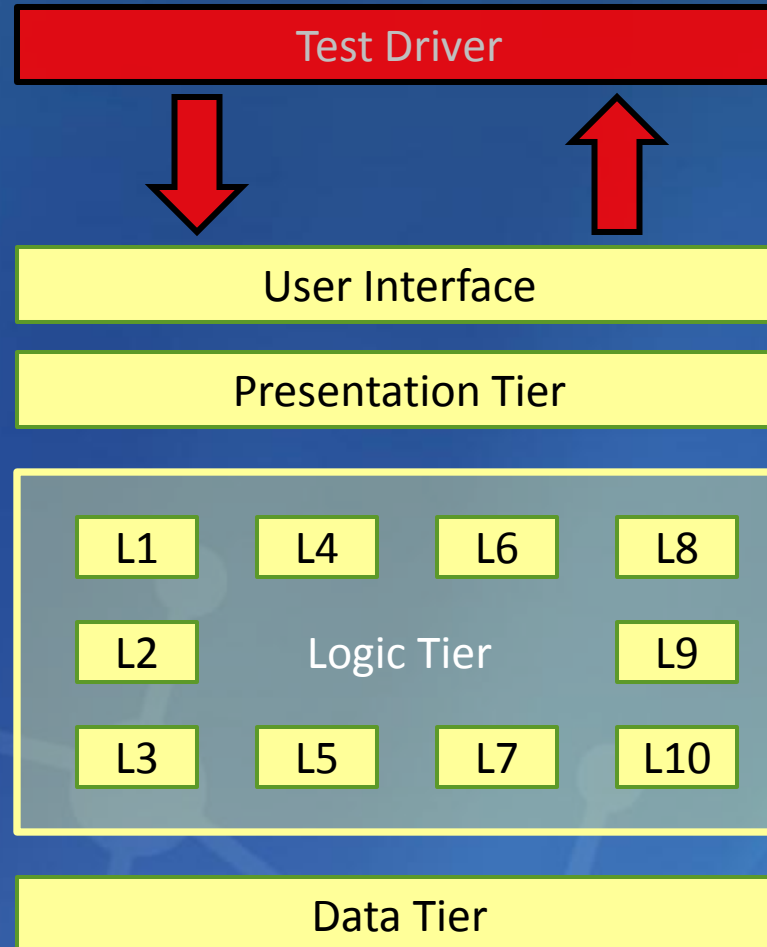
DIE „MESSAGE“ DER NÄCHSTEN 45MIN

- Scrum ist anders...
 - ... auch beim Testen!
- Warum so?
 - Gründe und Randbedingungen, die zur Struktur der Lösung geführt haben
- Was bringt's?
 - Entwickler sind gefordert
 - Gefragt: Beihilfe zum Test!
 - “Aber unter'm Strich”: Jede Menge Vorteile!

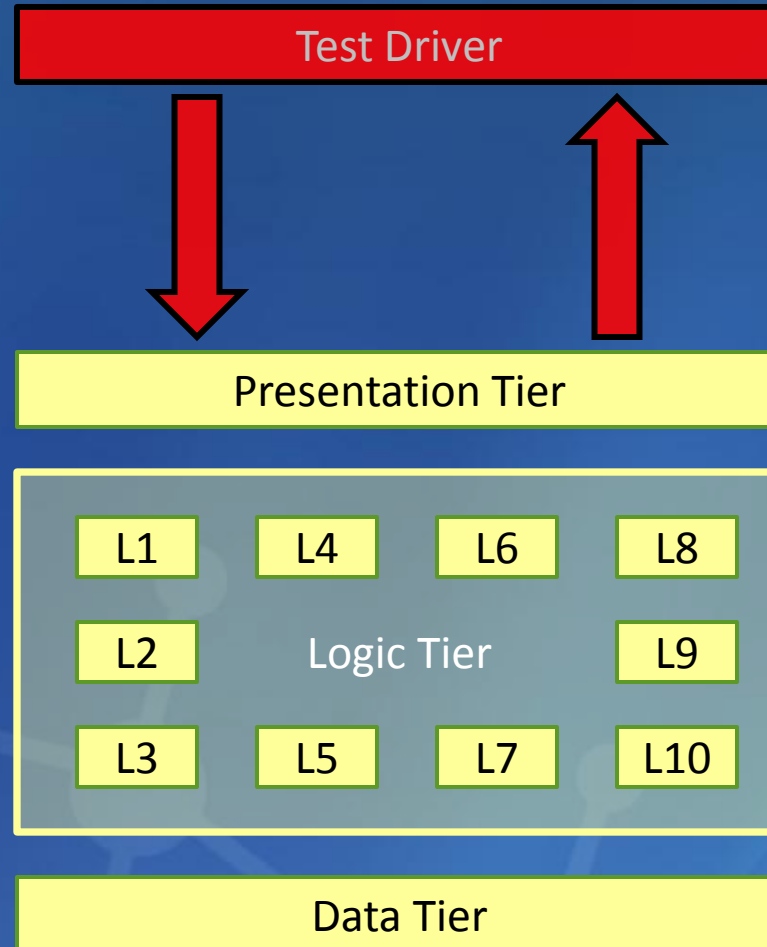
GESAMTSYSTEM



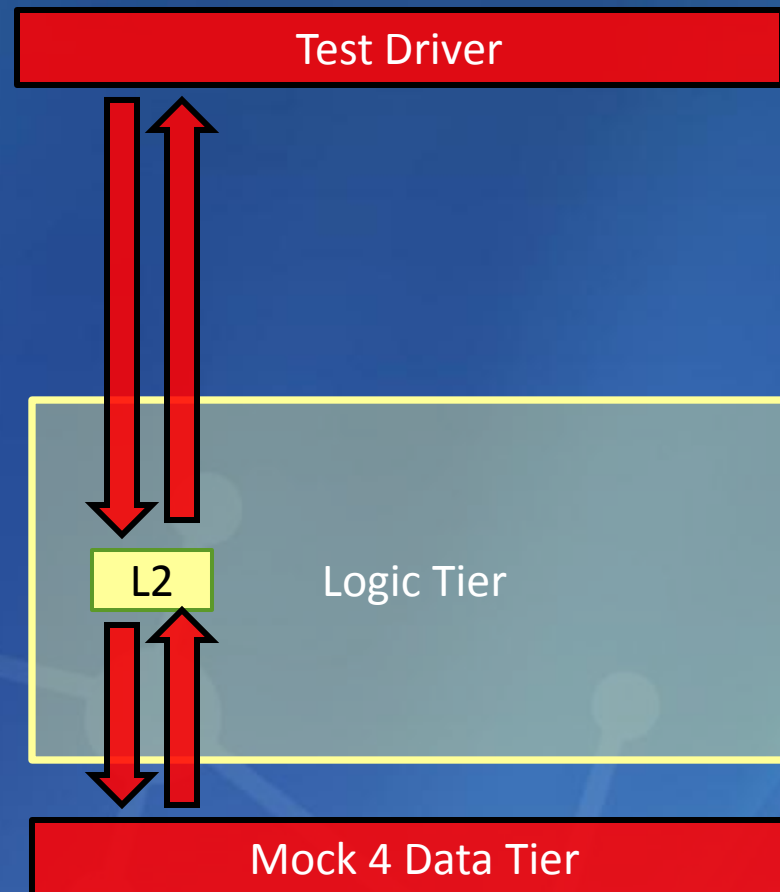
SYSTEMTEST (ÜBER UI)



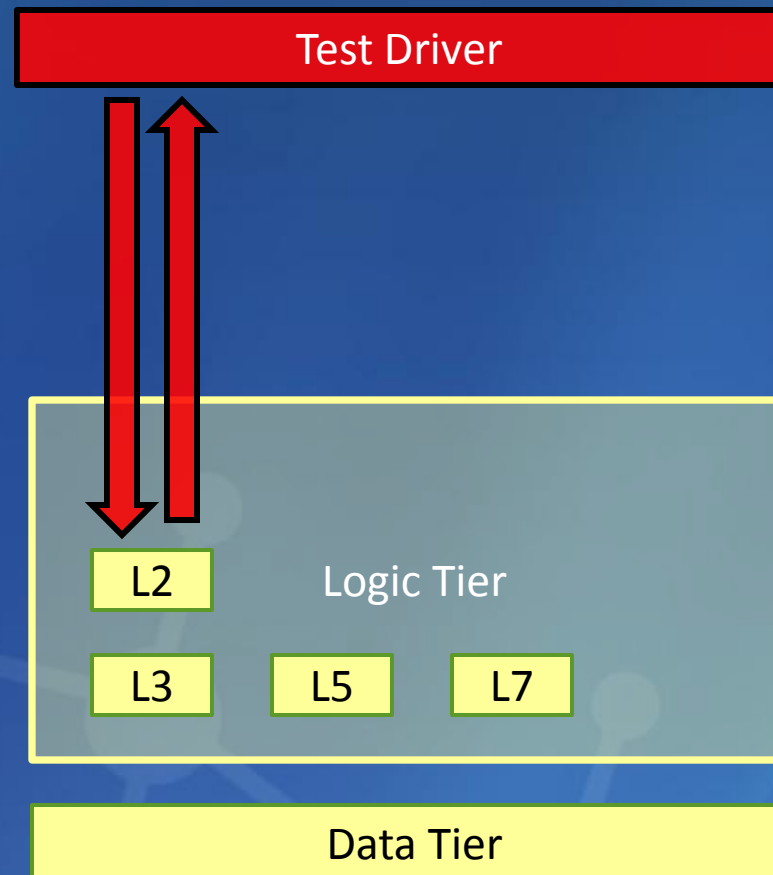
SYSTEMTEST (OHNE UI)



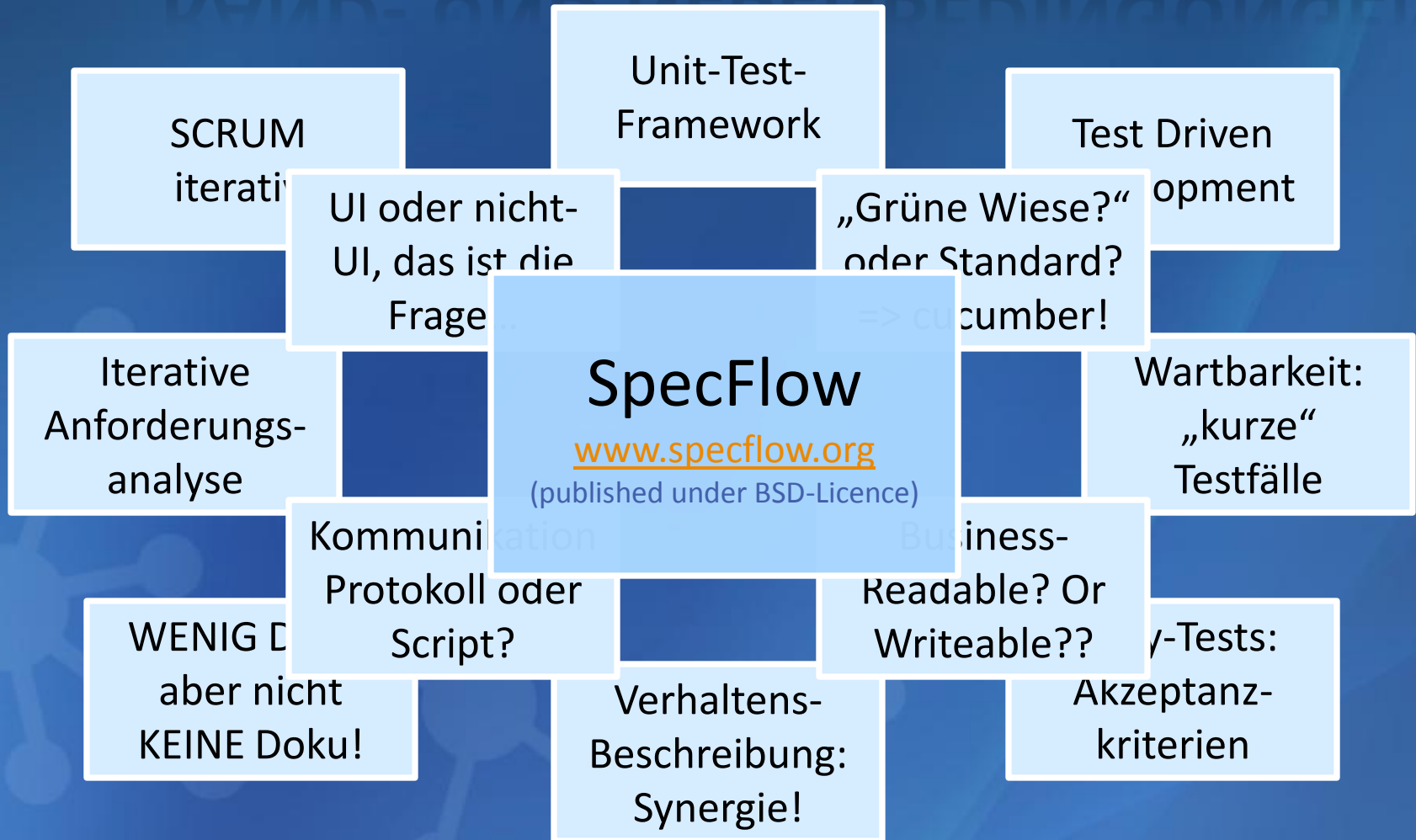
IM GEGENSATZ DAZU: UNIT TEST



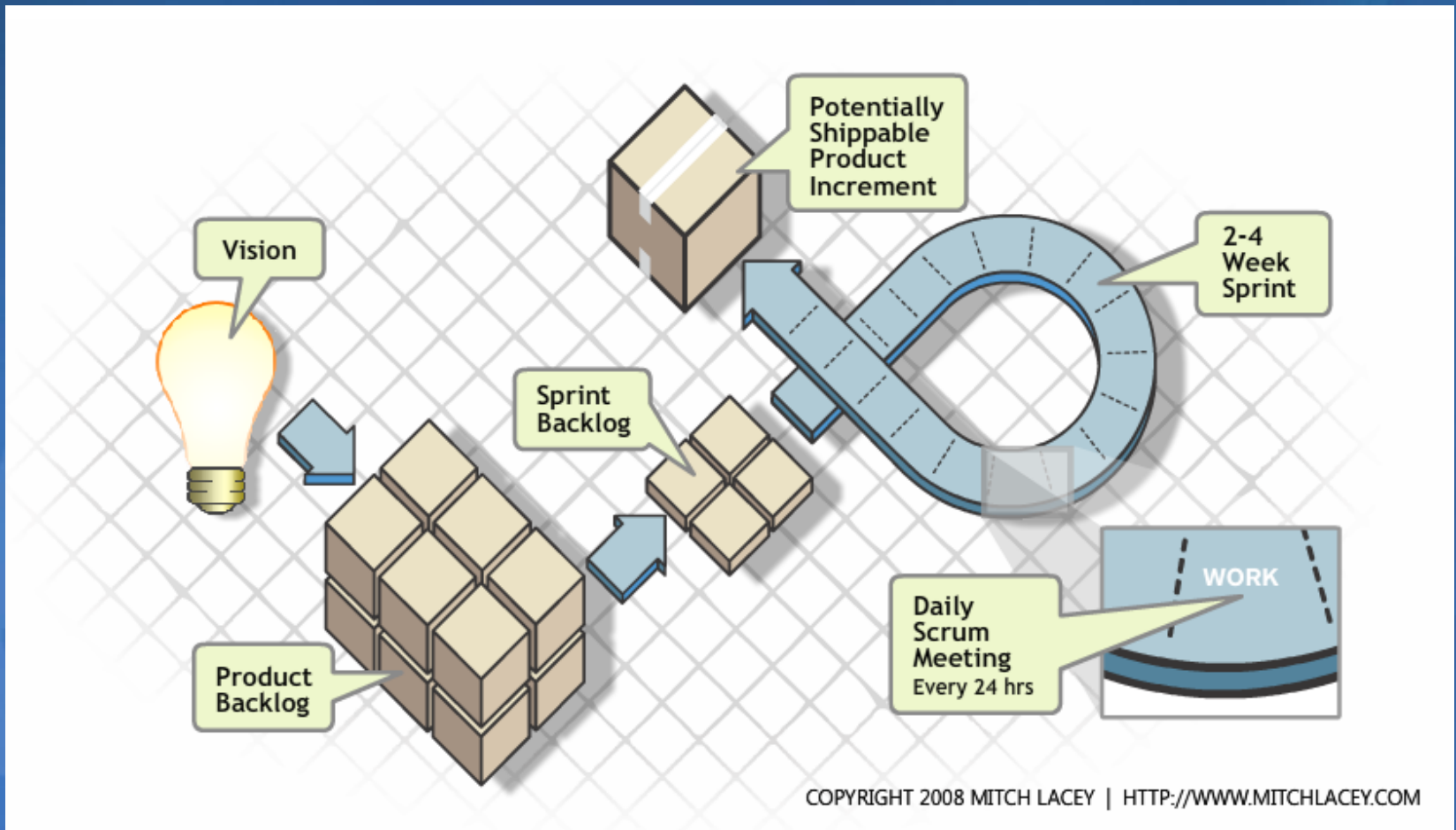
WEDER FISCH NOCH FLEISCH...



RAND- UND NEBENBEDINGUNGEN:



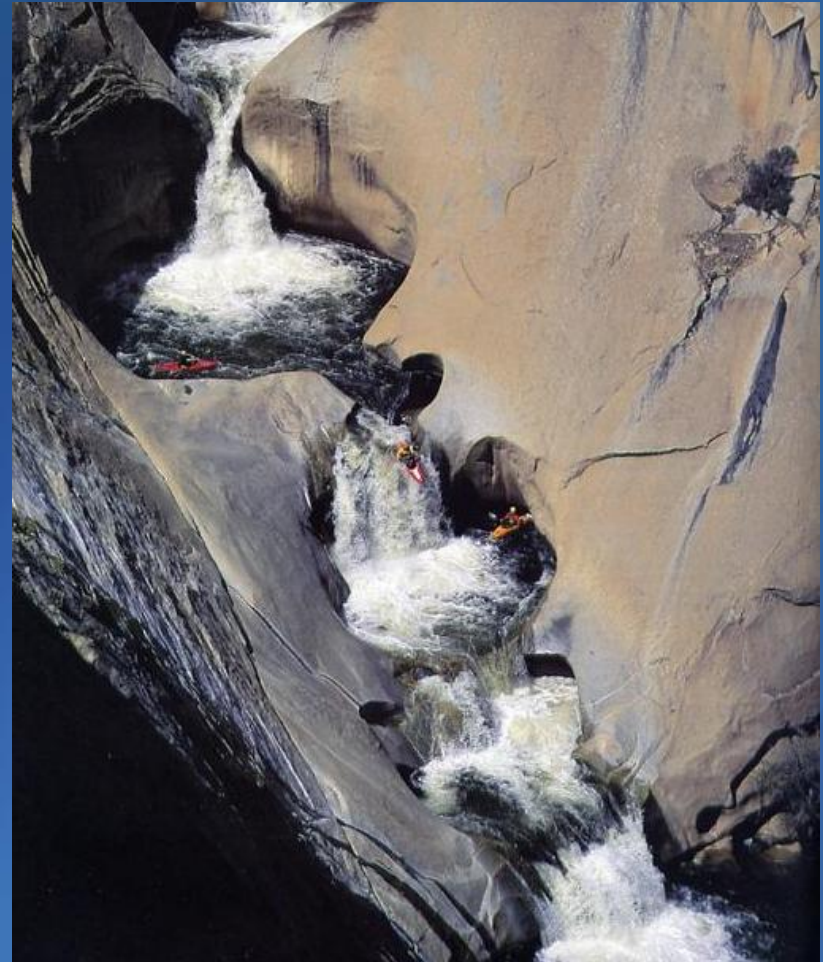
DER AUSLÖSER...: SCRUM!!!



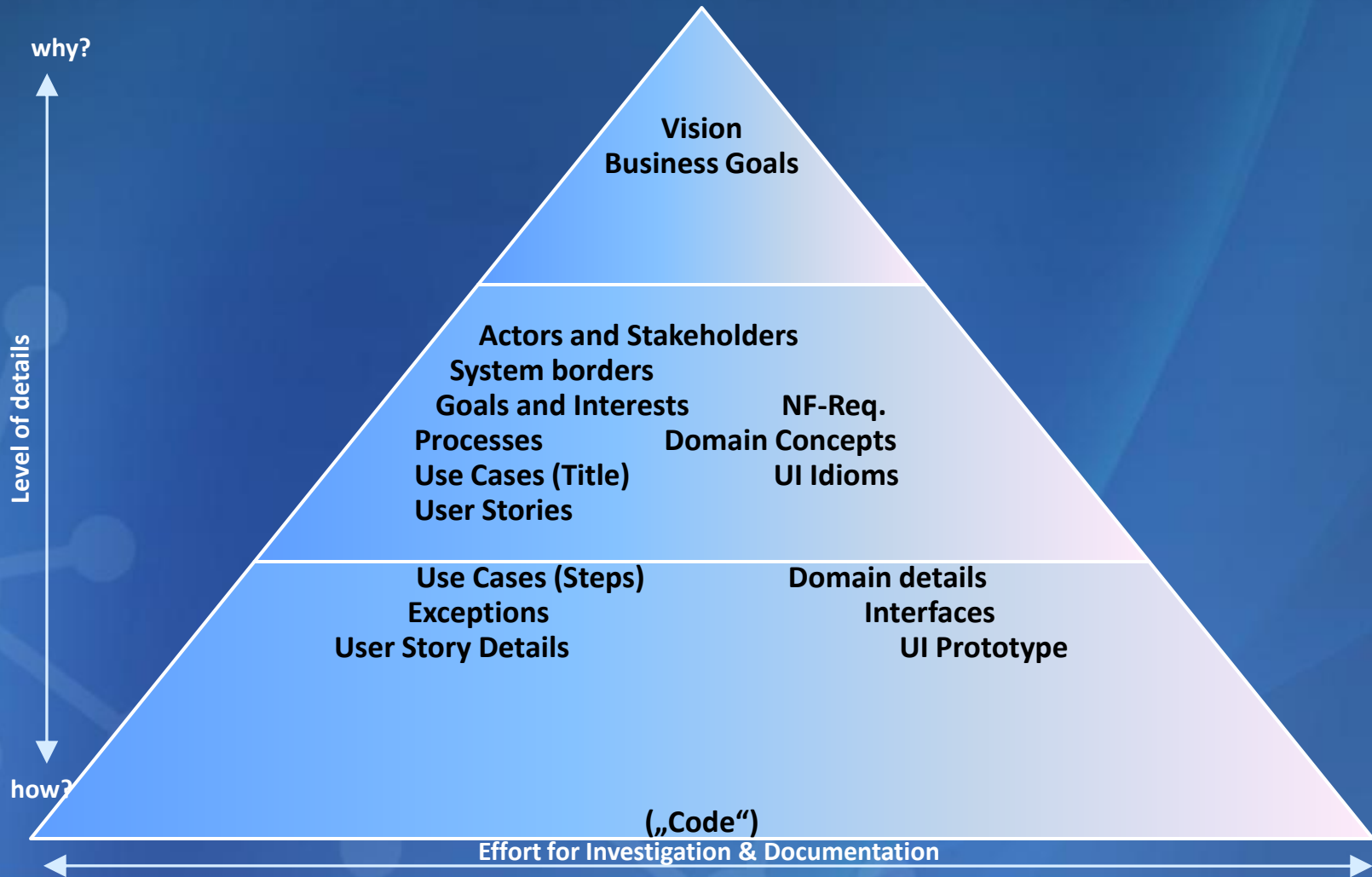
WAS ÄNDERT SCRUM AM TESTEN?

Wie war's denn früher?

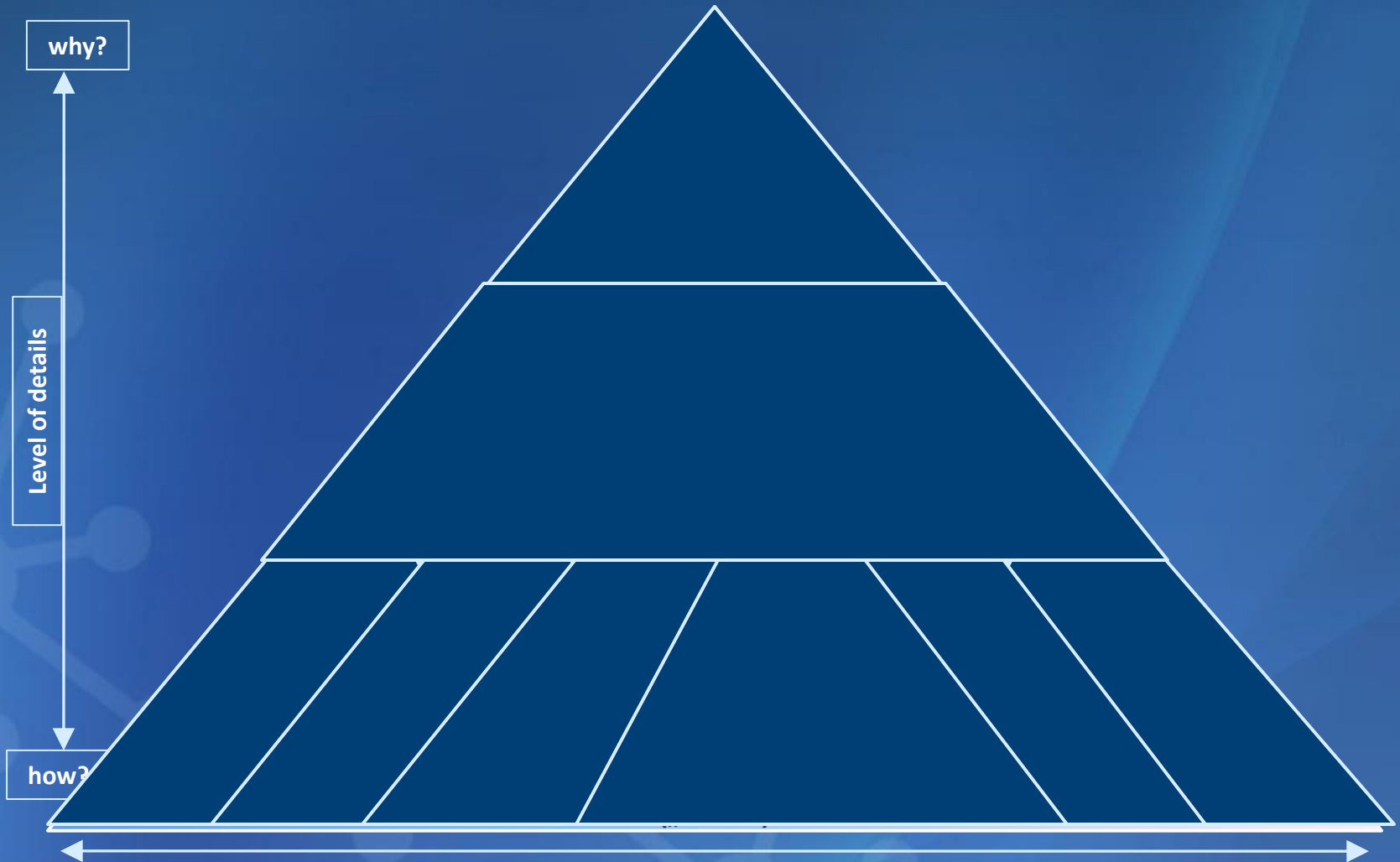
- Analyse, Design, Coding und Test
 - „Schritt für Schritt“
 - Jede Menge Doku...



WAS SOLL ICH ÜBERHAUPT TESTEN???



ANALYSE IM PROJEKTVERLAUF



SCRUM'SCHE HERAUSFORDERUNGEN:

- Detailanalyse
 - In jeweiliger Iteration
 - Kommunikation mit PO versus Dokumentation
 - „MÖGLICHST WENIG Doku“ ist nicht „KEINE Doku“
 - => was soll schon dokumentiert werden?
 - => wie soll es dokumentiert werden?
- Regressionstests – Again and again and...

AUTOMATISCHE TESTS - CHANCEN

- Zeitnahes Feedback für Entwickler
- Absicherung von Funktionalität
- Test- und Behaviour Driven Development
- Beschreibung von Verhalten

TDD - TEST DRIVEN DEVELOPMENT

Write a
failing
unit test



Make
the test
pass



refactor



AUSWIRKUNGEN VON TDD

- 15-35% Zusatzaufwand in der Entwicklung
- 80-90% weniger Aufwand bei
Wartung und Erweiterung
- 40-90% weniger Fehler

Quelle: <http://www.infoq.com/news/2009/03/TDD-Improves-Quality>

BDD-BEHAVIOUR DRIVEN DEVELOPMENT

Write a failing
acceptance
test

Write a
failing
unit test

Make
the test
pass

refactor

REQUIREMENTS == TESTS

*Traditionally Requirements: at the beginning,
tests at the end*

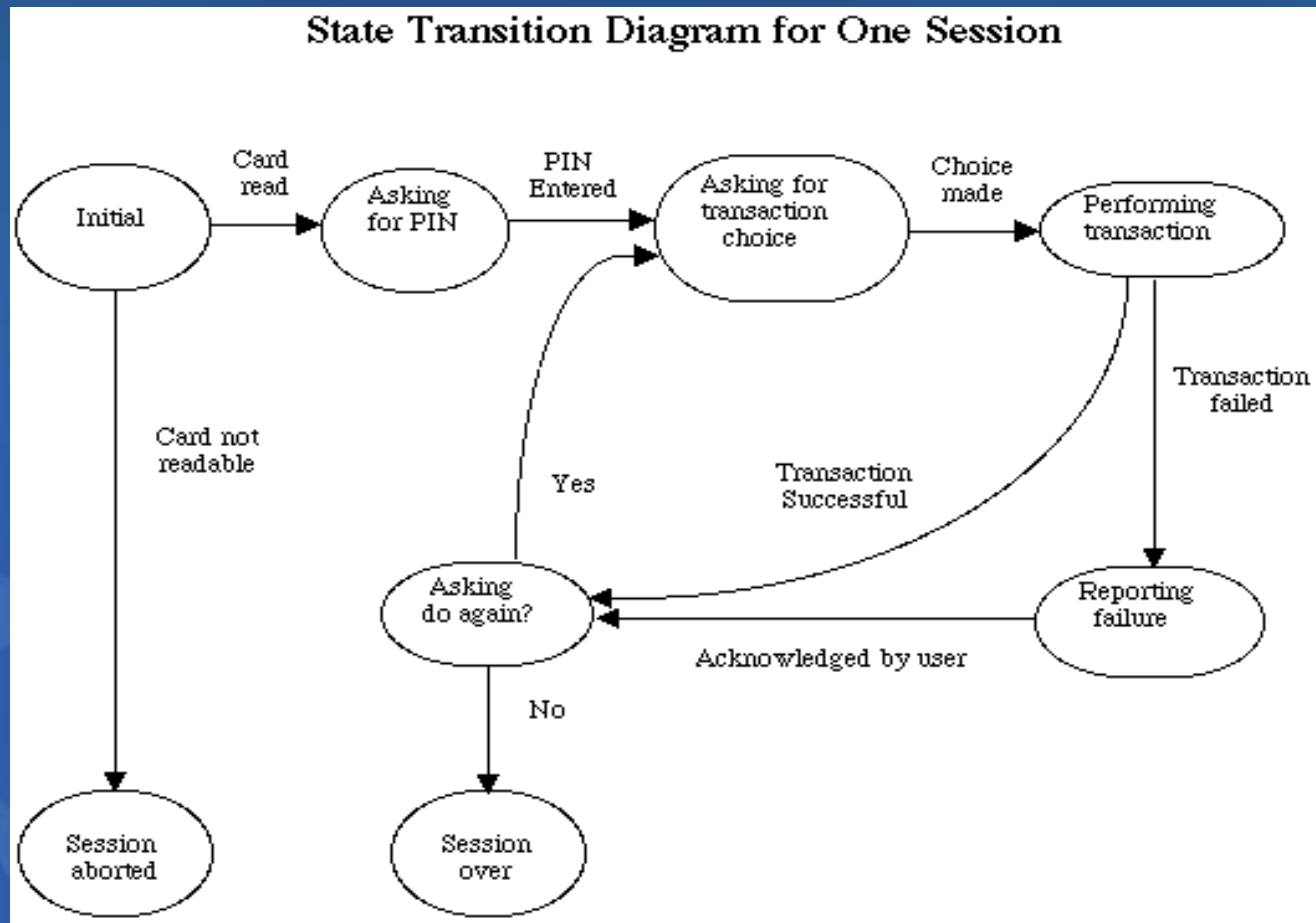
*Both: Describe, how the system is used
as formality increases, tests and requirements
become indistinguishable requirements are
equivalent.*

Equivalence hypothesis, Robert C. Martin

BUSINESS WRITEABLE TEST SCRIPTS

- Ermöglicht BDD: Write Acceptance Test first!
- Dokumentation des Verhaltens (Detail-Spec)
- Automatisch ausführbar: Verifiziert Verhalten

„KURZE“ VS. „LANGE“ TESTFÄLLE



„KURZE“ VS. „LANGE“ TESTFÄLLE

- „KURZ“
 - „single responsibility“
 - Traceability: Stories-Akzeptanzkriterien
 - Bessere Wartbarkeit
 - Anzahl: Geringerer Ordnung
- „LANG“
 - „Kombinatorische Explosion“
 - Höhere Änderungswahrscheinlichkeit

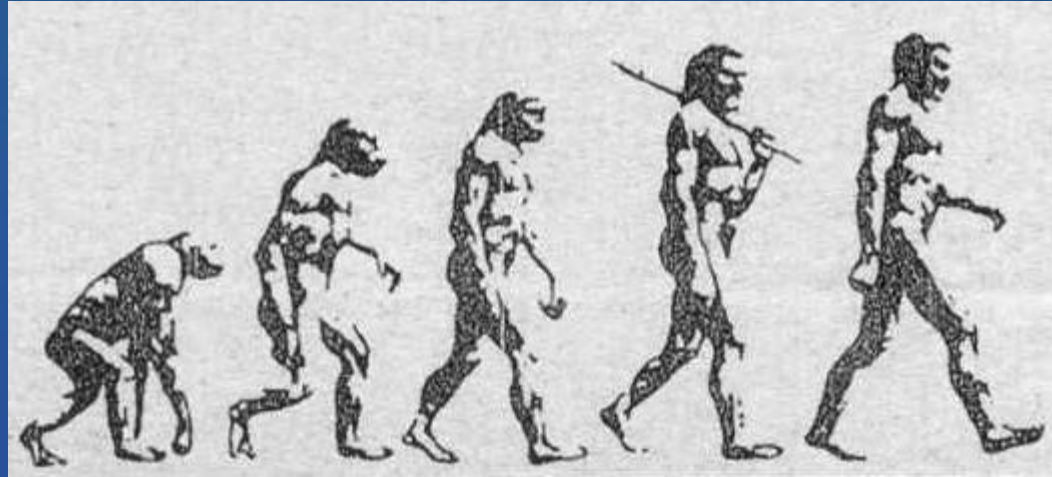
UI ODER NICHT-UI, DAS IST DIE FRAGE...

- Wenn UI, dann..
 - Nur „kurze“ Testfälle
 - Problem: Vorbedingungen schaffen?
 - Captured Script aus Unit-Test-Framework aufrufen
- Funktionalität im UI
 - Validierung, etc: Möglichst Vermeiden („Design 4 Testability“)
- Probleme am UI
 - Resultate verifizieren (Listen, ...)

UI ODER NICHT-UI, DAS IST DIE FRAGE...

- So viel wie nötig...
 - Client-seitige Validierung, etc
 - Eingabelänge, erlaubte Zeichen, etc
 - Optische Erscheinung, Usability
- ... so wenig wie möglich!
 - Testziel auch über Business-Logic
 - Daten-Variationen, Funktionalität
 - Bessere Performance

EVOLUTION DER TESTAUTOMATISIERUNG



- Unit Test Framework
 - Capture and Replay
 - Browser Control via Unit-Test
 - Keyword / Script driven

UNIT TEST FRAMEWORK

```
[TestMethod]
public void SetGFDatenTest2 ()
{
    Geschaeftsfall gf = GetNeuenFall ();
    tempObjects.Add (gf);

    GeschaeftsfallDatenParameter param = new GeschaeftsfallDatenParameter ();
    param.KatastralGemeindeNummer = 4002;
    param.AnkaufsFoerderung = 2000m;
    param.Ausleihungssumme = 20000;
    param.BauvollendungsfristVerlaengert = new DateTime (2010,1,1);
    param.Denkmalschutz = true;
    param.VersicherungAdresse = "1010 Wien, Teststrasse 4711";
    param.VersicherungName = "TestVersicherung";

    ProtokollierungDetailParameter detailParameter = new ProtokollierungDetailParameter (
    detailParameter.Einreichdatum = IPDateTime.Today.AddDays (-1);
    detailParameter.StandortEinreichung = StandOrt.QueryProvider.GetAll ().GetItem (0);
    detailParameter.ZeitSchrift = true;
    detailParameter.AntragsArt = KommunikationsArt.Post;
    detailParameter.Modellart = ModellartEnum.IP;

    param.DetailParameter = detailParameter;
    GeschaeftsfallServices.Instance.UpdateGeschaeftsfallDaten (gf, param, new List<Einlag

    Assert.AreEqual (4002, gf.GebietsInformation.KatastralGemeindeNummer);
    Assert.AreEqual (param.AnkaufsFoerderung, gf.AnkaufFoerderung);
    Assert.AreEqual (param.Ausleihungssumme, gf.Ausleihungssumme);
    Assert.AreEqual (param.BauvollendungsfristVerlaengert, gf.BauvollendungsfristVerlaenge
    Assert.AreEqual (param.Denkmalschutz, gf.Denkmalschutz);
    Assert.AreEqual (param.VersicherungAdresse, gf.VersicherungAdresse);
    Assert.AreEqual (param.VersicherungName, gf.VersicherungName);
```

UNIT TEST FRAMEWORK

- Vorteile:
 - Tool + Know-How dazu schon vorhanden
 - Hohe Wartbarkeit (bei gut gewählter Struktur...)
 - Nebeneffekt: Versions- u. Variantenkontrolle
- Nachteil:
 - Weder business-writeable noch –readable
 - Keine UI-Logik testbar
 - BDD braucht „Dolmetscher“ (= Entwickler) für Akzeptanz-Kriterium nach C#-Übersetzung

SELENIUM

Ermächtigungsdaten | Auswertungen | Administration

Geeignete Person erfassen

[Zurück](#) Test-Login Seite

Person | Befähigung §57a | Beschäftigungen in Begutachtungsstellen

[Bearbeiten](#) [Löschen](#)

Person

Nachname * **Magreither**
 Vorname * **Paul**
 Geburtsdatum * **14.07.1975** Titel

Wohnadresse

Strasse **Landstraße 37**
 PLZ **3500** Ort **Krems an der Donau**

Aktiv

Aktiv **Ja**
 Aktiv seit
 Inaktiv seit
 Inaktiv Grund
 Bildungspassnummer **BPN-123-1578**

Sonstiges

Information über Vertrauenswürdigkeit
 Bemerkungen

[Bearbeiten](#) [Löschen](#)

Selenium IDE *

Datei (F) Bearbeiten Optionen Hilfe

Base URL **http://vpc-mdbui/**

Fast Slow

Command	Target	Value
open	/noel.mdb.test/ind...	
click	link=Sachbereichsl...	
click	link=Suche Strg+S...	
select	ct100_Content_sear... label=Person	
click	//option[@value='...	
type	ct100_Content_pers... Magreither	
click	ct100_Content_ct10...	
click	link=Magreither	

Command **click**
 Target **link=Suche Strg+Shift+S** Find
 Value

Log Reference UI-Element Rollup

click(locator)

Arguments:

- locator - an element locator

Clicks on a link, button, checkbox or radio button. If the click action causes a new page to load (like a link usually does), call `waitForPageToLoad`.

SELENIUM

- Vorteile:
 - UI-Logik auch testbar
 - Open Source
- Nachteil:
 - Stand-Alone: Komplexe Testfälle (mehrere Screens)
Never-ever try this at home!!!
- Kombination mit Unit-Test Framework:
 - Capture-Tool für einzelne Testschritte

CUCUMBER

```
emacs@TTV-ATHENA
File Edit Options Buffers Tools Help
@web
Feature: Search for transactions

Scenario: Simple search by name
  Given I am on the search page
  And an order for 'max musterman' exists
  When I execute a simple search for 'abc'
  Then I should see 'max musterman' in the results
```

```
require 'selenium'

Before('@web') do
  @selenium = Selenium::SeleniumDriver.new("localhost", 4444,
  10000):
  @selenium.start
end

After('@web') do
  @selenium.stop
end

Given /^I am on the search page$/ do
  @selenium.open "/Orders/Search.aspx"
end

When /^I execute a simple search for '(.)'$/ do |search|
  @selenium.type "ctl00_DefaultContent_txtPhrase_text", search
  @selenium.type "ctl00_DefaultContent_txtPhrase", search
  @selenium.click "ctl00_DefaultContent_btnSearchReference"
  @selenium.wait_for_page_to_load "30000"
end

Then /^I should see '(.)' in the results$/ do |result|
  assert @selenium.is_text_present(result)
end

Given /^an order for '(.)' exists$/ do |name|
  pending
end
```

CUCUMBER

- Vorteile:
 - Beschreibt Systemverhalten
 - „Business Writeable“
 - „Binding“: Beliebige Ebene (UI, Presenter, ...)
- Nachteil:
 - Testautomatisierung ist auch SW-Entwicklung...
(but keep in mind: Quality is free – if you spend much money on it.)

EVOLUTION DER TESTAUTOMATISIERUNG

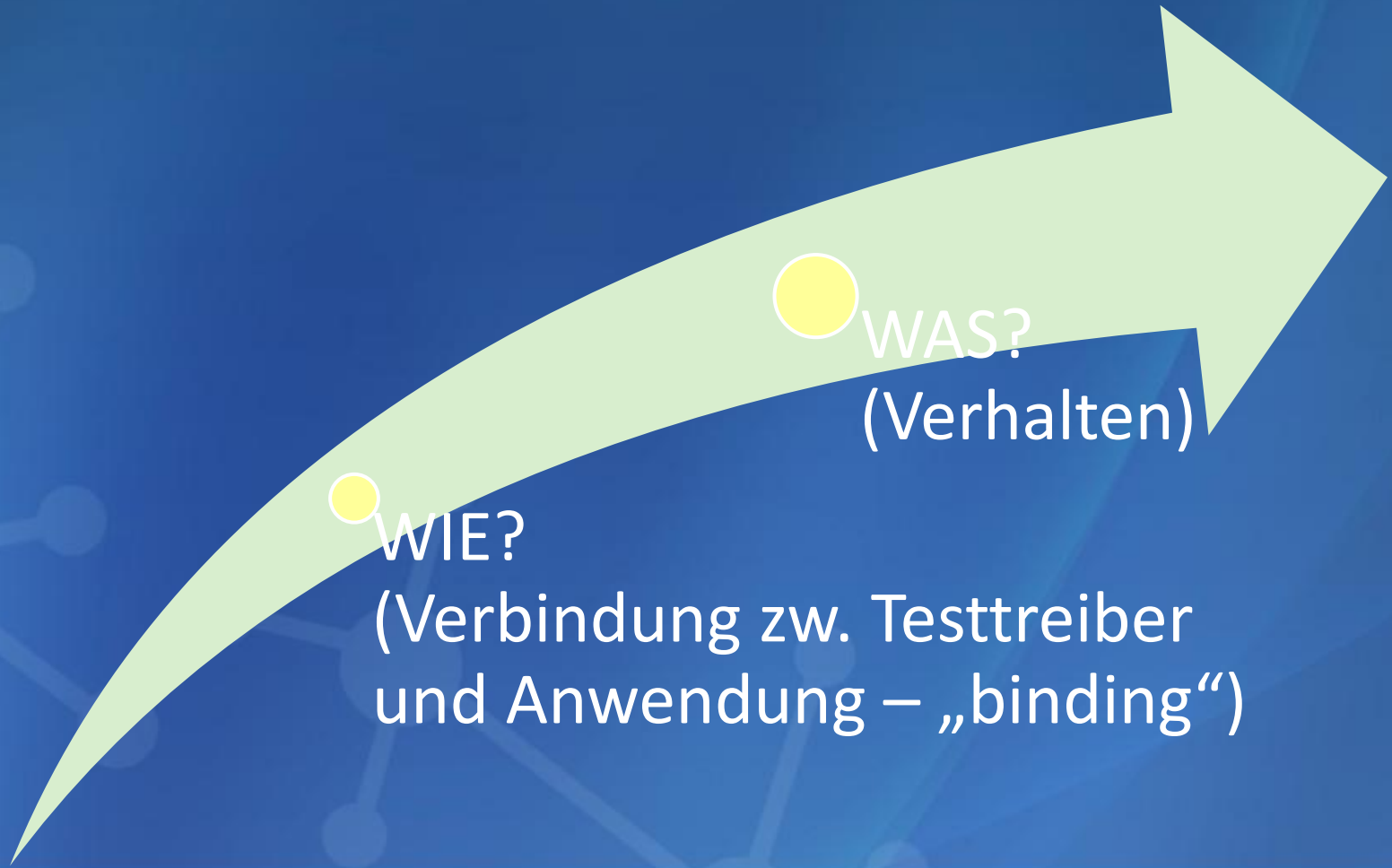
Unit Test
Framework
(Spaghetti-
Style)

Unit Test
Framework
(well
structured)

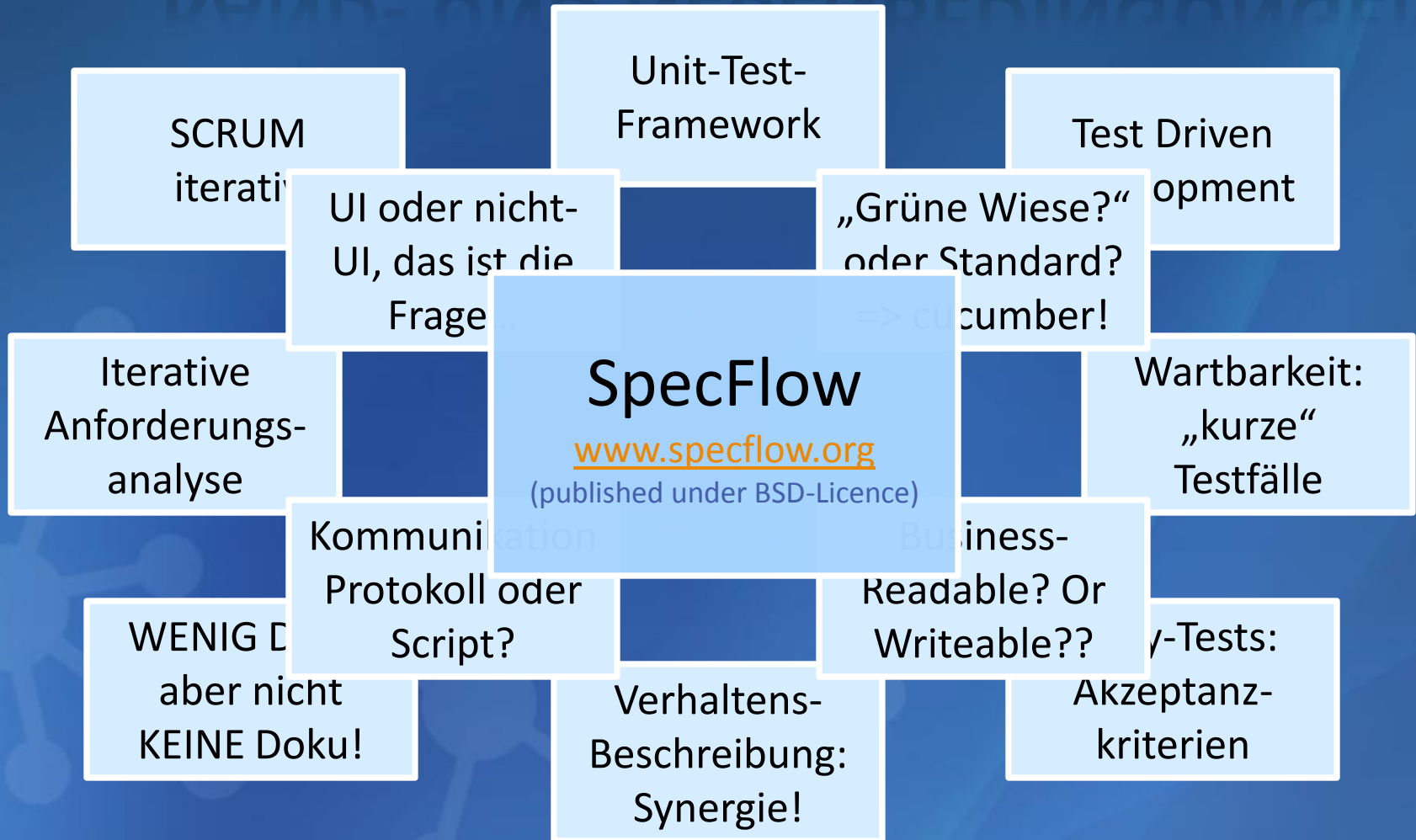
Business-Writeable
Script Language

- Visual-Studio-Komfort
(Versionskontrolle,
 - Wartbarkeit
- Kommunikation mit Product Owner und Business
- Ermöglicht TDD!

WAS VERSUS WIE:



RAND- UND NEBENBEDINGUNGEN:



... und nach der Pause geht's weiter!

DI Alexander Schramek

Quality Manager, Senior Project Manager



TechTalk Software Support HandelsgmbH

Leonard-Bernstein-Straße 10, A-1220 Wien

ADDITIONAL RESSOURCES

- www.css-web.net
- www.microsoft.com/en/us/
- <http://blogs.msdn.com/msdnat/>