

Documentation as Code

Dokumentierst Du noch, oder
programmierst Du schon?



Matthias Hölzer-Klüpfel

Dokumentationsanforderungen



Software-Dokumentation nach 62304

- ▶ Software-Entwicklungsplanung
- ▶ Softwareanforderungen
- ▶ Softwarearchitektur
- ▶ Detailliertes Softwaredesign
- ▶ Integrationstests und Testergebnisse
- ▶ Softwaresystemtests und Testergebnisse
- ▶ Freigabedokumente

Software-Dokumentation für FDA-Einreichungen

Geforderte Dokumentation	Minor LoC	Moderate LoC	Major LoC
Bestimmung des Level of Concern	X	X	X
Beschreibung der Software	X	X	X
Risikoanalyse des Gerätes	X	X	X
Softwarespezifikation	Kurzfassung	vollständig	vollständig
Softwarearchitektur		X	X
Detailliertes Design		X	X
Traceability-Analyse	X	X	X
Entwicklungsprozess		Kurzfassung	vollständig
Verifikation und Validierung	Funktions-tests	Funktions- und System-tests	vollständig
Versionshistorie	X	X	X
Liste mit bekannten Fehlern		X	X

Medical SPICE Work Products

Application Specification
Change Documentation
Change Information
Change Management Plan
Change Request
Change Review Record
Feedback
Known Anomaly List
List of Residual Anomalies
Overall Risk Evaluation
Personnel Qualification Records
Primary Operating Functions
Problem Advisory
Problem Investigation Record
Problem Report
Registry of Released Versions
Residual Anomalies Evaluation Record
Risk Analysis
Risk Management Plan
Risk Management Policy
Risk Management Process
Risk Management Report
Software Architecture Verification Record

Software Architecture
Software Build Record
Software Configuration History Record
Software Configuration Management Plan
Software Delivery Procedures
Software Detailed Design Verification Record
Software Detailed Design
Software Development Lifecycle Definition
Software Development Plan
Software Documentation Plan
Software Integration Plan
Software Integration Regression Test Record
Software Integration Test Plan
Software Integration Test Record
Software Integration Test Verification Record
Software Integration Verification Record
Software Maintenance Plan
Software Modification Kit
Software Problem Resolution Plan
Software Release Kit
Software Release Record
Software Requirements Traceability
Software Requirements Verification Record

Software Requirements
Software Risk Management Plan
Software Source Code
Software System Regression Test Record
Software System Test Plan
Software System Test Record
Software System Test Specification
Software System Test Verification Record
Software Unit Acceptance Criteria
Software Unit Verification Record
Software Verification Plan
Software Version Archive
SOUP Specification
System Architecture
System Requirements
Training Material
Usability Engineering Plan
Usability Specification
Usability Verification Record
User Interface Design
Validation Plan
Validation Record

Arten von Dokumentation

Dokumente

- ▶ unterliegen einem Lebenszyklus (Prüfung, Freigabe, Aktualisierung, ...)
- ▶ verfügen über eine Version(snummer)

Aufzeichnungen

- ▶ belegen die Konformität mit dem Qualitätsmanagement-System
- ▶ werden erstellt und aufbewahrt

Strategien



Früher

Funktionalität

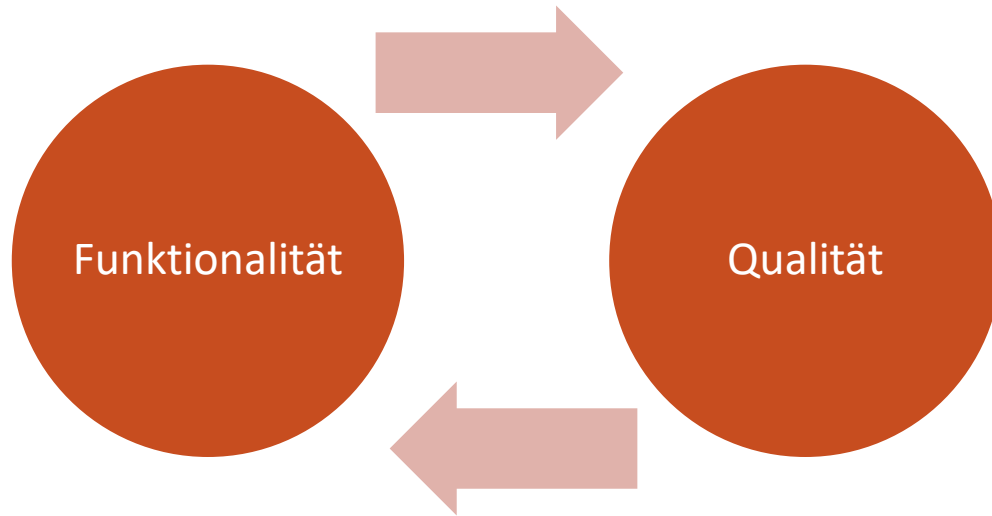


Qualität



Dokumentation

Heute



A light orange arrow points downwards from the space between the two circles above towards the "Dokumentation" box below.

Dokumentation

Agile Ansätze



Agiles Manifest

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

While there is value in the items on the right, we value the items on the left more.

Potentiell auslieferbares Produkt

Ziel einer Iteration sollte es sein:

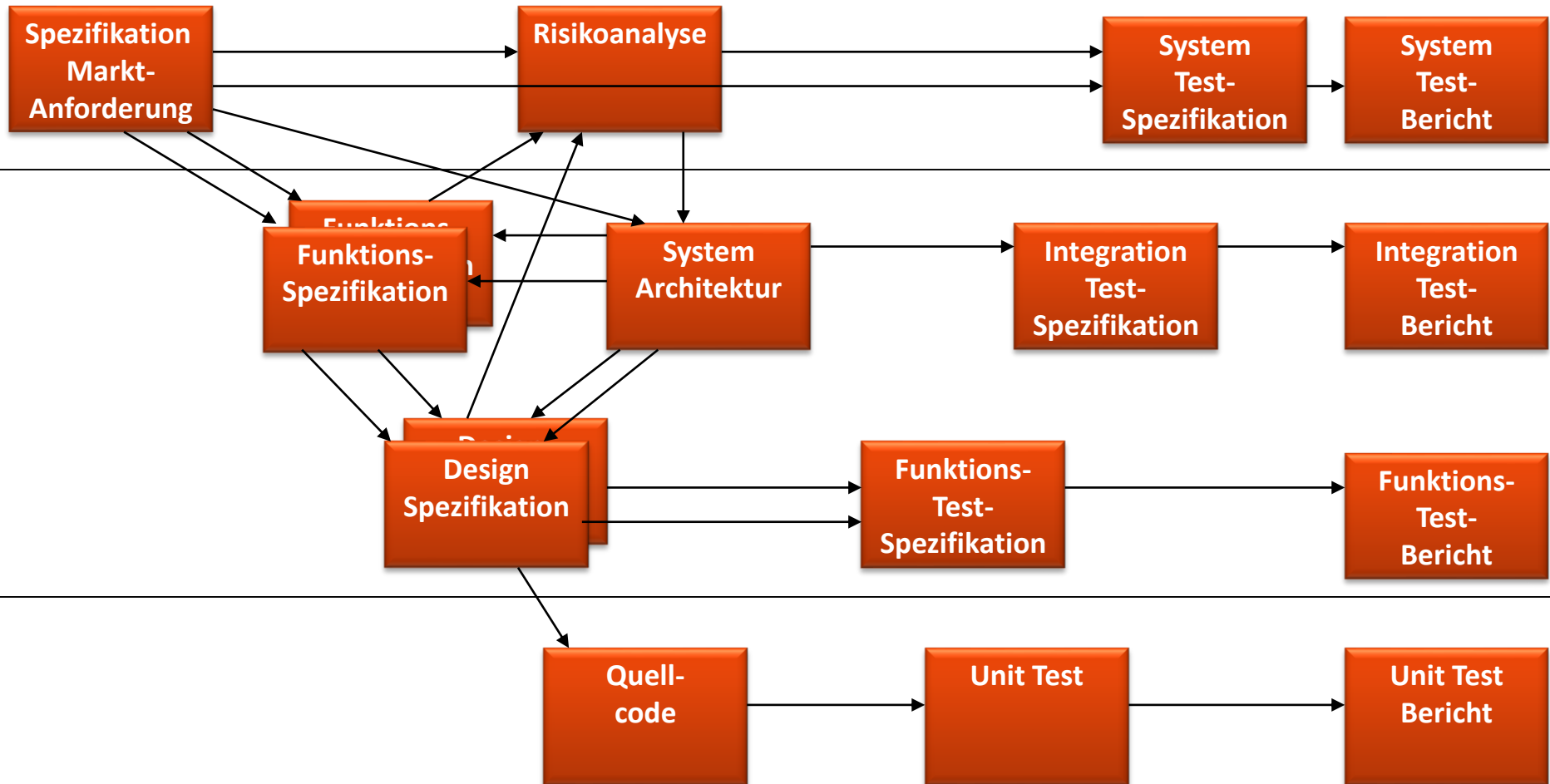
- ▶ eine lauffähige Software,
- ▶ mit erfolgreich ausgeführten Tests, und
- ▶ konsistenter technischer Dokumentation

zu liefern.

Schwierigkeiten



Traceability



Persistente vs. transiente Information

Persistent

- ▶ ist dauerhaft von Relevanz
- ▶ Beispiele:
 - ▶ Anforderungen
 - ▶ Architektur
 - ▶ Design

Transient

- ▶ ist für kurze Zeit relevant
- ▶ Beispiele:
 - ▶ Fehlerberichte
 - ▶ Backlog-Einträge

Persistente vs. transiente Information

Problematisch ist die Vermischung, z.B.

- ▶ User Stories als Backlog-Einträge **und** als Anforderungsdokumentation
- ▶ Akzeptanzkriterien als Definition of Done **und** als Testfallspezifikation
- ⇒ Transiente Information nutzen, um persistente Information zu erzeugen!

Freigabeprozesse

Problematisch ist die Vermischung, z.B.

- ▶ User Stories als Backlog-Einträge **und** als Anforderungsdokumentation
- ▶ Akzeptanzkriterien als Definition of Done **und** als Testfallspezifikation
- ⇒ Transiente Information nutzen, um persistente Information zu erzeugen!

Medienbrüche

SW-Anforderungen

SW-Systemprüfung

SW-Architektur

SW-Integrationstest

Detailliertes Design

Verifikation d. Impl.

Implementierung

Medienbrüche

DOORS

Squish

Enterprise Architect

googletest

Doxygen

Code Collaborator

Visual Studio

Medienbrüche

ALM

ALM

ALM

ALM

IDE

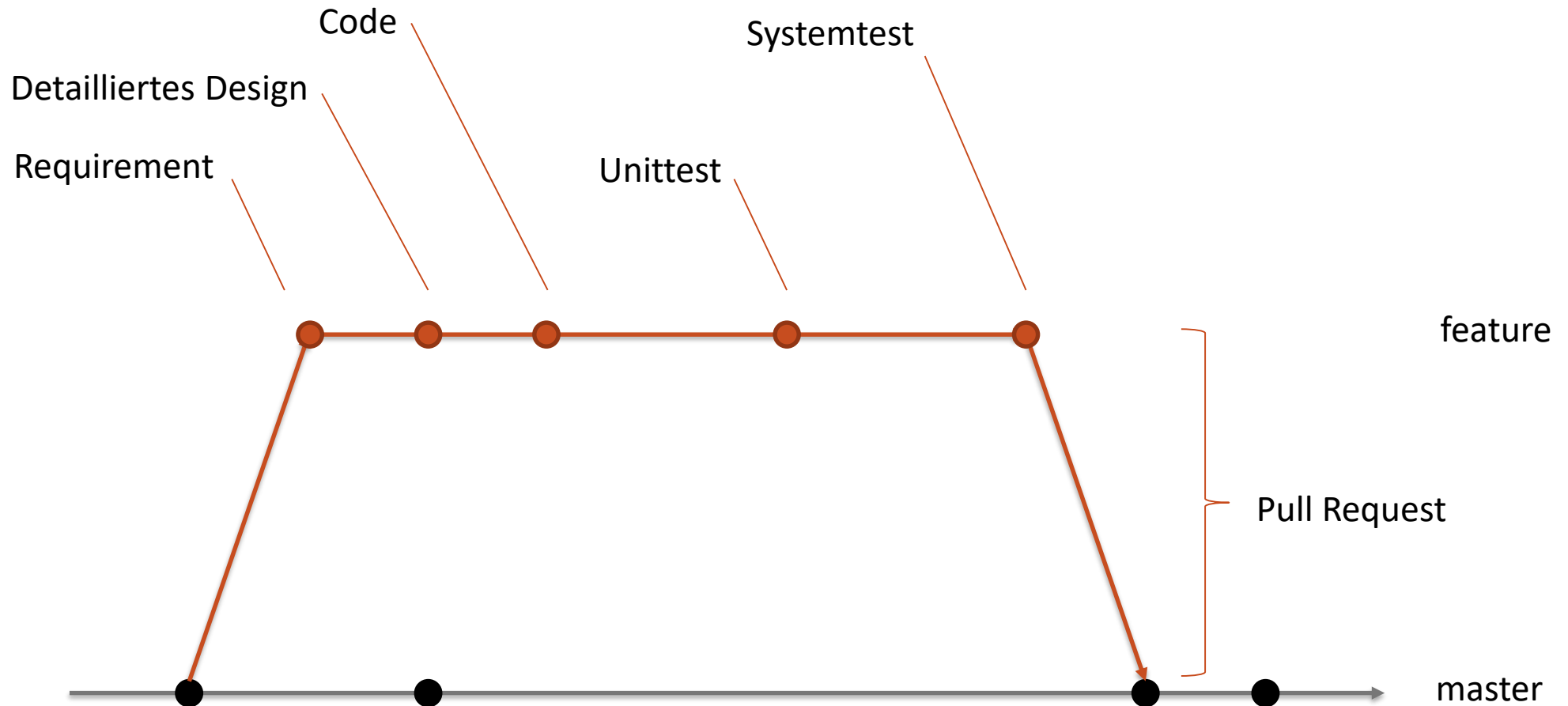
IDE

IDE

Lösungsmöglichkeiten / Projektbeispiele



Pull-Request Workflow



Pull-Request Prüfungen

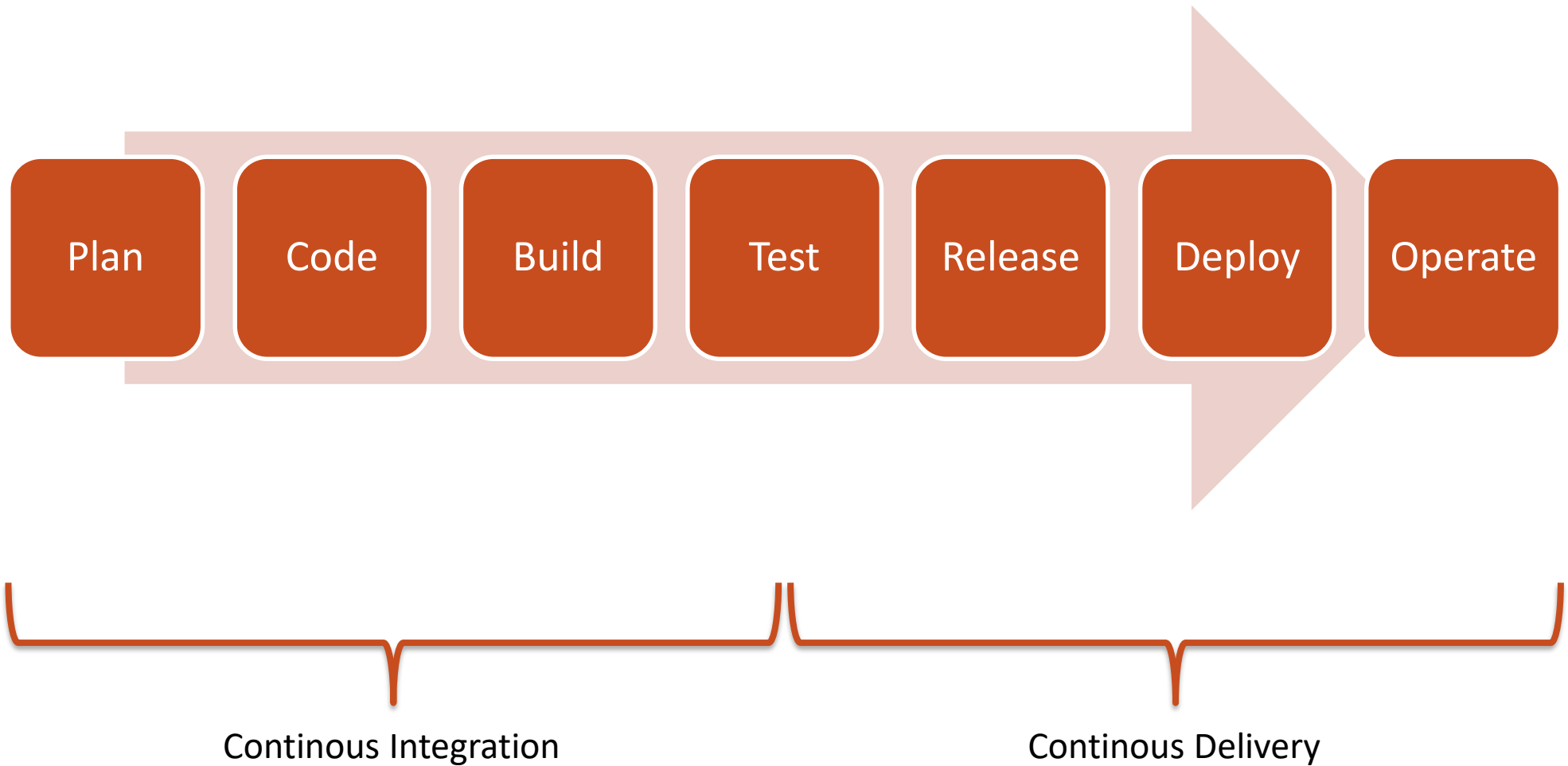
Vor dem mergen des Pull-Requests wird geprüft:

- ▶ Kompilierung
- ▶ Statische Code-Analyse
- ▶ Unit-Tests
- ▶ (automatisierte) Software-Systemtests
- ▶ Code-Review

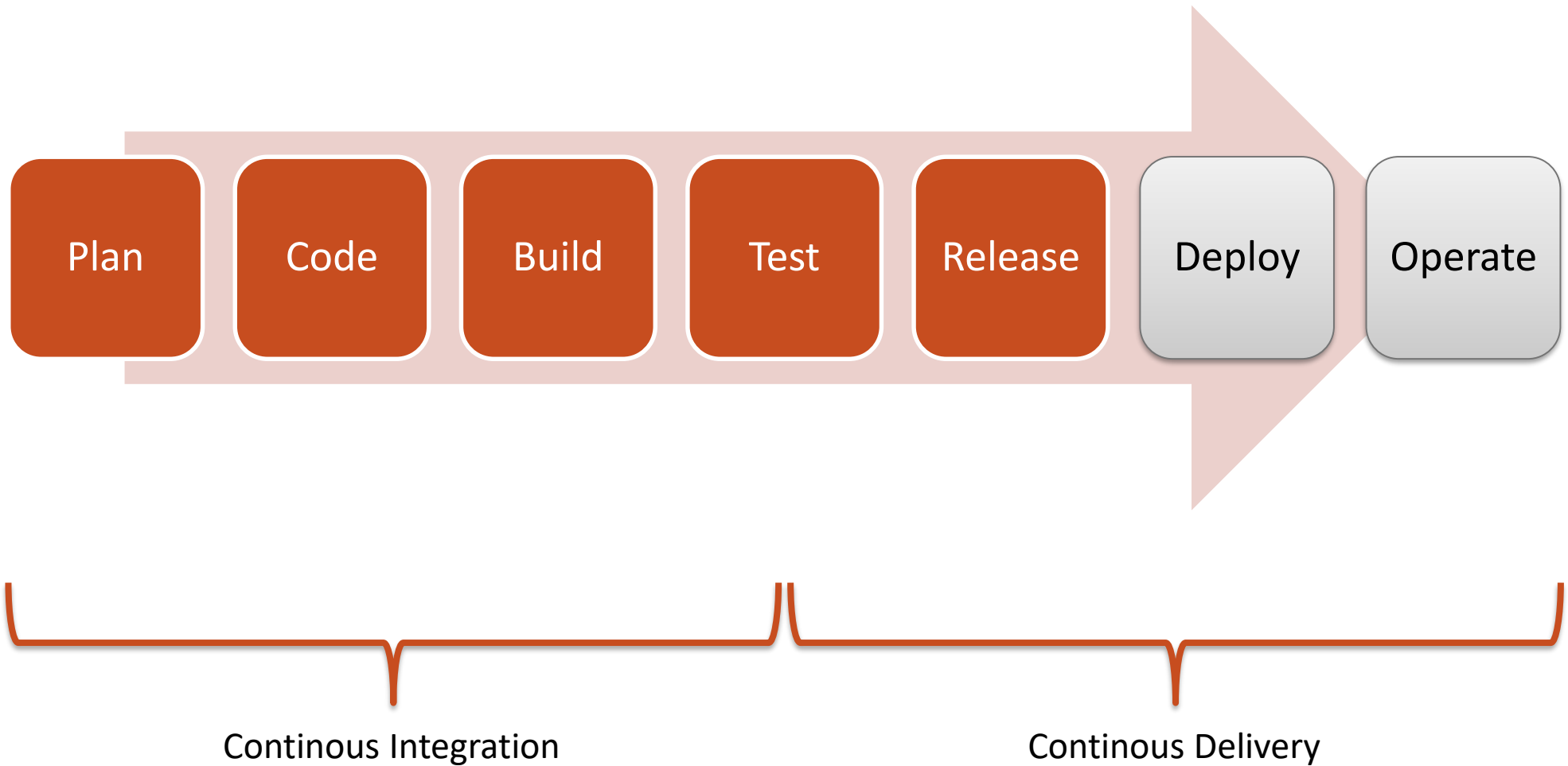
- ▶ **Dokumentengenerierung**
- ▶ **Traceability**

Prüfung der Dokumentationsinhalte im Rahmen des Approvals der Pull-Requests.

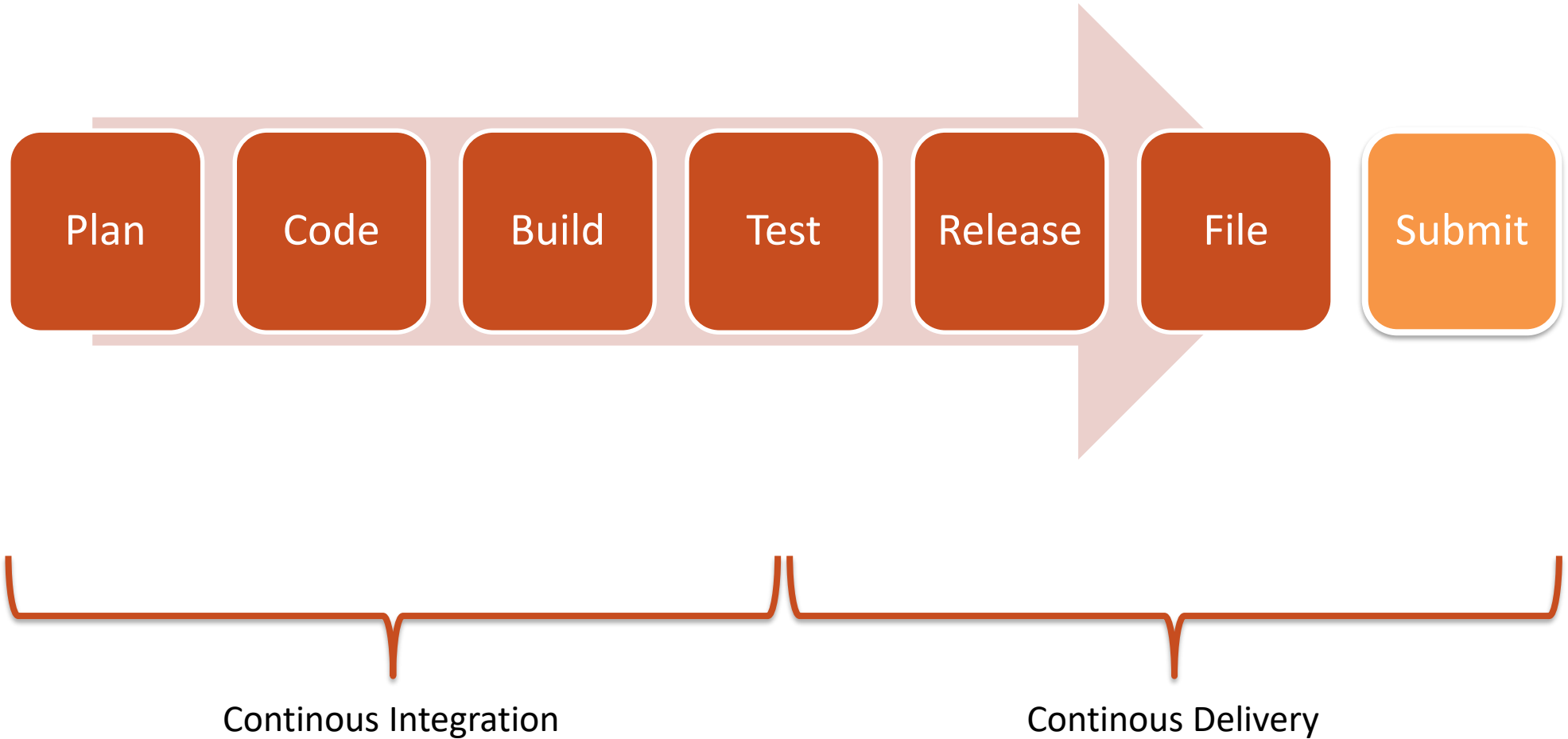
DevOps-Pipeline



DevOps-Pipeline



DevOps-Pipeline



Textbasierte Dokumentation

Artefakt	Beispiele
Requirements	Markdown, LaTeX
Architektur	PlantUML
Design	Doxygen
Code	😊
Buildsystem	Dockerfile, Jenkinsfile, azure-pipelines.xml, bitbucket-pipelines.xml
Unittests	googletest et. al.
Systemtests	Behave, Gherkin



Meist das
größte
Problem...

Dokumentengenerierung

Generierung von Dokumenten aus Source-Code, z.B.

- ▶ Anforderungsspezifikationen aus Markdown
- ▶ Detaillierte Design-Spezifikationen mit doxygen
- ▶ Testreports aus automatisierten Testergebnissen

Beispiel: Testskript

Beispiel.xml X

```
1  <?xml version="1.0" encoding="utf-8" ?>
2  <?xml-stylesheet type="text/xsl" href="test-script.xsl"?>
3
4  <testscript revision="$Revision: 123 $" version="1.0.0" xmlns="http://www.hoelzerkluepfel.de/testscript">
5
6      <sequence id="TEST1" name="Alarmfunktion">
7
8          <description>
9              Dieser Test überprüft die Funktion des visuellen Alarms.
10          </description>
11
12          <repeat count="5">
13              <wait start="490ms" timeout="510ms">
14                  <check-transition name="Alarm LED" from="Off" to="On"/>
15              </wait>
16              <wait start="490ms" timeout="510ms">
17                  <check-transition name="Alarm LED" from="On" to="Off"/>
18              </wait>
19          </repeat>
20
21          <verify requirement="REQ1234"/>
22
23      </sequence>
24
25  </testscript>
```

REQ1234: Test der Alarm LED

Nach dem POST blinkt die Alarm LED
5 mal mit 1Hz und 50% Einschaltdauer.

Testspezifikation

1 TEST1 - Alarmfunktion

1.1 Introduction

Dieser Test überprüft die Funktion des visuellen Alarms.

1.2 Test Steps

Step	Action	Expected Result									
#1	Repeat the following steps for 5 times:	All repeated commands pass									
	<table><tr><th>Step</th><th>Action</th><th>Expected Result</th></tr><tr><td>#1</td><td>Wait for the following condition/s to become true: - The 'Alarm LED' transitions from 'Off' to 'On' </td><td>Condition true in less than 510ms but not before 490ms</td></tr><tr><td>#2</td><td>Wait for the following condition/s to become true: - The 'Alarm LED' transitions from 'On' to 'Off' </td><td>Condition true in less than 510ms but not before 490ms</td></tr></table>		Step	Action	Expected Result	#1	Wait for the following condition/s to become true: The 'Alarm LED' transitions from 'Off' to 'On'	Condition true in less than 510ms but not before 490ms	#2	Wait for the following condition/s to become true: The 'Alarm LED' transitions from 'On' to 'Off'	Condition true in less than 510ms but not before 490ms
	Step		Action	Expected Result							
	#1		Wait for the following condition/s to become true: The 'Alarm LED' transitions from 'Off' to 'On'	Condition true in less than 510ms but not before 490ms							
#2	Wait for the following condition/s to become true: The 'Alarm LED' transitions from 'On' to 'Off'	Condition true in less than 510ms but not before 490ms									
Finished verification for requirement REQ1234											

1.3 Traces to Requirements

This test script will verify the following requirements:

- REQ1234

Test Report

1 Test Run

1.1 Summary

1.1.1 Test Tainted

This test run is tainted! The test results are not reliable, and can not be used for formal verification!

Reason/s for this:

- The test has been run on a debug version of test automation software!
- Test Board Firmware is not a released version

1.1.2 Warnings

This test run issued the following warnings:

- One or more tests have been marked as tainted, the results of this session are unreliable

1.1.3 Test Results

This test run passed: Script successfully concluded

The results of the individual tests in this test run where as follows:

Test	Result	
Alarmfunktion	This test passed.	



1.2 Execution Details

Attribute	Value
Test Execution Start	2013-10-28 12:32:37.6154
Test Script	Executable Tests\Beispiel.xml
Test Driver Version	1.3.43
Test Script Revision	\$Revision: 123 \$
Test Script Language Version	1.0.0
Tester	mhk
Test Computer	test03
Operating System	Microsoft Windows NT 6.1.7601 Service Pack 1
Common Language Runtime	4.0.30319.239
Command Line	"C:\SW Test Automation\Test Driver.exe" "Executable Tests\Beispiel.xml"
Test Report	TestReport.xml
Control Software Version	INFORMAL
Safety Software Version	INFORMAL

Externes Trace-Tool

Überprüfung der Traceability auf der Basis der Dokumente:

- ▶ Erstellen eines Trace-Modells
- ▶ Extraktion der Schlüssel
- ▶ automatische Überprüfung

Beispiel Trace

```
verbose = True

project = "Sandbox"

PATH = "documents\\products\\" + project + "\\SRS-SDS\\"

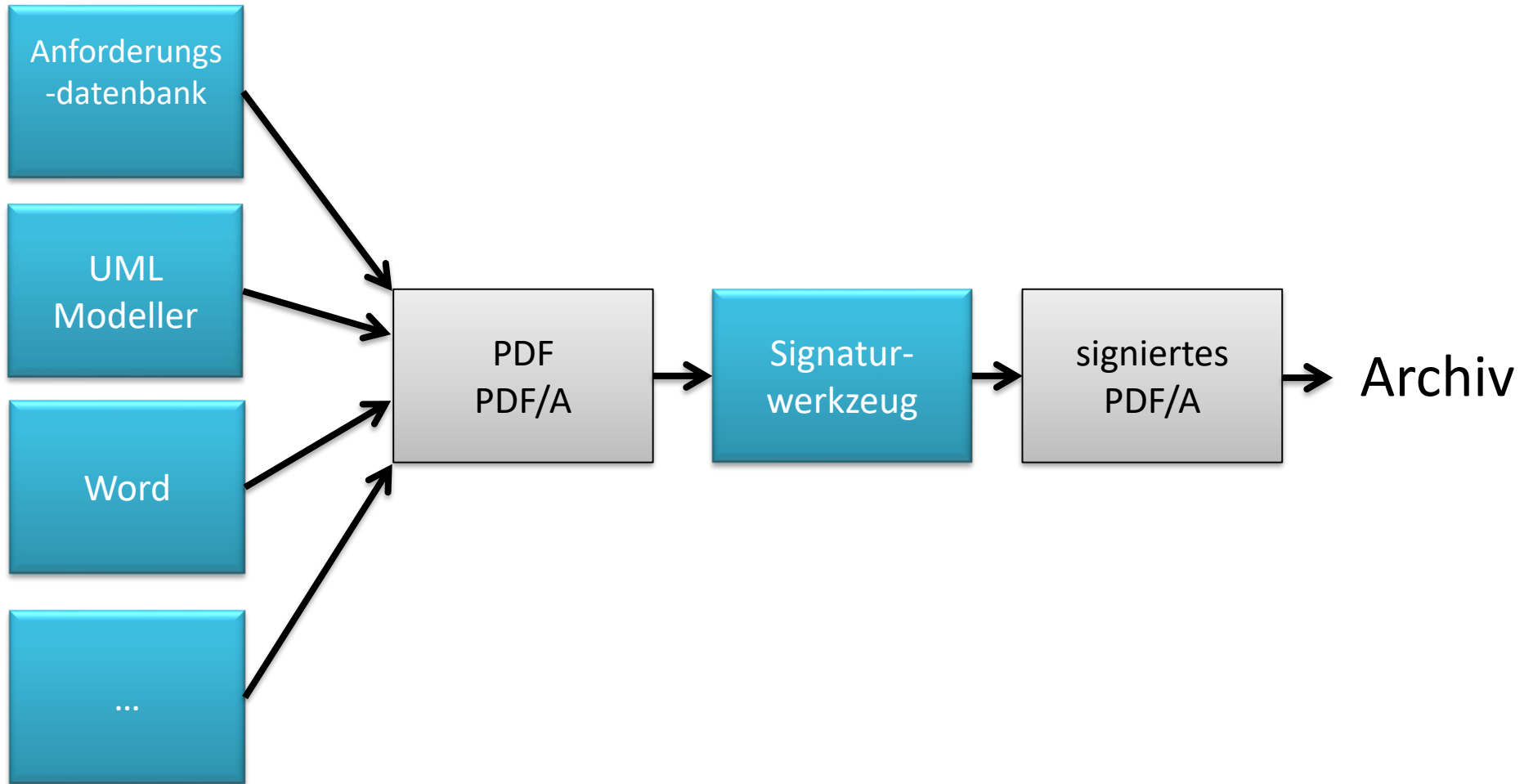
prs = PDFExtractor(PATH + project + "-SRSu.pdf", [r"(ET\d\d).?(UR\d\d)"], [])
srs = PDFExtractor(PATH + project + "-SRSp.pdf", [r"(SF\d\d).?(SR\d\d)"], [r"(\([ivx]*\))"])
prs_srs = PDFExtractor(PATH + project + "-SRSp.pdf", [r"(ET\d\d).?(UR\d\d)"], [r"(SF\d\d).?(SR\d\d)"])

report.timestamp("Trace report generated:")

missing = []
for et in prs.traces:
    if not prs_srs.traces[et]:
        missing.append(et)
report.list("ETs that are not linked to a SR:", missing, verbose)

missing = set(srs.traces.keys())
for key, items in prs_srs.traces.items():
    missing = missing - items
report.list("SRs that are not linked to an ET:", missing, verbose)
```

Elektronische Signatur



GDS-SAS-0102-01 Software Architecture Specification.pdf - Nitro Pro 10

	Generic Dialysis System	
0-6		0-6

Project	Document ID	Revision	Date of Issue	Status

	0102	0021	01	1900-01-01	Draft	
--	------	------	----	------------	-------	--

Gültigkeitsstatus

gültigkeit der Signatur: Signatur ist gültig

Dokumentveränderungen: Das Dokument wurde seit der Signierung nicht verändert oder

enthält nur zugelassene Änderungen.

Generic Dialysis System

Elektronische Signatur

Idealer Workflow:

- ▶ Integration in einen Pull-Request
- ▶ Elektronische Signatur zur Bestätigung eines Approvals
- ▶ Automatische PDF-Signatur der entstandenen Dokumente

Fazit



- ▶ Traditionell wird Dokumentation getrennt vom Software-Sourcecode behandelt
- ▶ Die Werkzeuge der Software-Entwicklung haben sich enorm weiterentwickelt; die der Dokumentation kaum
- ▶ Dokumentation analog zu Code zu behandeln erlaubt es, Workflows zu automatisieren
- ▶ Dazu ist es notwendig, neue Dokumentationsformen zu ermöglichen

Kontakt



MATTHIAS HÖLZER-KLÜPFEL DIPLOM-PHYSIKER, M.SC.

- ▶ **post** Landsteinerstraße 4
97074 Würzburg
Germany
- ▶ **tel** +49 931 32072-821
fax +49 931 32072-819
mobil +49 176 6085 7994
- ▶ **mail** matthias@hoelzer-kluepfel.de
web www.hoelzer-kluepfel.de