

Continuous Integration im medizinischen Bereich

Ein praktisches Beispiel

Gliederung

1. Einleitung
2. Relevanz im medizinischen Bereich
3. Continuous Integration
4. Continuous Integration System des BZMM
5. openOR Buildsystem
6. openOR Testsystem
7. Ergebnisse
8. Snapshot: Build Tool Chain

Gliederung

1. Einleitung
2. Relevanz im medizinischen Bereich
3. Continuous Integration
4. Continuous Integration System des BZMM
5. openOR Buildsystem
6. openOR Testsystem
7. Ergebnisse
8. Snapshot: Build Tool Chain

Einleitung

- große Projekte ohne Software Engineering Praktiken kaum möglich
- Beispiel: Continuous Integration
- Einführung im BZMM
- Einblick gewähren durch Vortrag

BZMM

- Berliner Zentrum für Mechatronische Medizintechnik
- Kooperation aus Fraunhofer und Charite
- Leitung: Prof. Dr.-Ing Erwin Keeve
- Arbeitsbereiche:
 - Intraoperative Bildgebung
 - Bildgesteuerte Navigation
 - Chirurgische Instrumente



Gliederung

1. Einleitung
2. Relevanz im medizinischen Bereich
3. Continuous Integration
4. Continuous Integration System des BZMM
5. openOR Buildsystem
6. openOR Testsystem
7. Ergebnisse
8. Snapshot: Build Tool Chain

Relevanz im medizinischen Bereich

- Medizinproduktgesetz
 - hohes Qualitätsmanagement notwendig
- eher große Projekte



Stärkenwörterbuch der Justiz **juris**

Gesetz über Medizinprodukte (Medizinproduktegesetz - MPG)

Nichtamtliches Inhaltsverzeichnis

MPG
Ausfertigungsdatum: 02.08.1994

Wörterbuch

"Medizinproduktegesetz" in der Fassung der Bekanntmachung vom 7. August 2002 (BGBl. I S. 3148), das zuletzt durch Artikel 13 des Gesetzes vom 8. November 2011 (BGBl. I S. 2178) geändert worden ist"

Stand: Neugefasst durch Bek. v. 7.8.2002 / 3148;
zuletzt geändert durch Art. 13 G v. 8.11.2011 / 2178

Näheres zur Bandangabe finden Sie im Menü unter [Links](#)

Dieses Gesetz dient der Umsetzung

- der Richtlinie 90/385/EWG des Rates vom 20. Juni 1990 zur Angleichung der Rechtsvorschriften der Mitgliedstaaten über aktive implantierbare medizinische Geräte (ABl. EG Nr. L 189 S. 17), zuletzt geändert durch die Richtlinie 93/58/EWG (ABl. EG Nr. L 220 S. 1),
- der Richtlinie 93/42/EWG des Rates vom 14. Juni 1993 über Medizinprodukte (ABl. EG Nr. L 169 S. 1), zuletzt geändert durch die Richtlinie 2001/104/EG (ABl. EG Nr. L 6 S. 50) und
- der Richtlinie 98/79/EG des Europäischen Parlaments und des Rates vom 27. Oktober 1998 über In-vitro-Diagnostika (ABl. EG Nr. L 331 S. 1).

Fußnote

+++ Textausweis ab: 01.01.2008 +++
vom Institut für Rechts der Informatik und 80 Seiten

Rechtsprechung des
Bund: 101/94 (GGBl. Nr. 101000000)
Bund: 101/94 (GGBl. Nr. 101000000)
Bund: 101/94 (GGBl. Nr. 101000000)

Rechtsprechung des
Land: 101/94 (GGBl. Nr. 101000000)
Land: 101/94 (GGBl. Nr. 101000000)
Land: 101/94 (GGBl. Nr. 101000000)

Rechtsprechung des
Land: 101/94 (GGBl. Nr. 101000000)
Land: 101/94 (GGBl. Nr. 101000000)
Land: 101/94 (GGBl. Nr. 101000000)

Durchführung des
der 101/94 (GGBl. Nr. 101000000) vgl. S. 24.7.2009 1.0000

Gliederung

1. Einleitung
2. Relevanz im medizinischen Bereich
- 3. Continuous Integration**
4. Continuous Integration System des BZMM
5. openOR Buildsystem
6. openOR Testsystem
7. Ergebnisse
8. Snapshot: Build Tool Chain

Continuous Integration

- Praxis aus der Softwareentwicklung
- Idee: Häufige “Integration“ des Codes in das Repository (daher der Name)

Continuous Integration

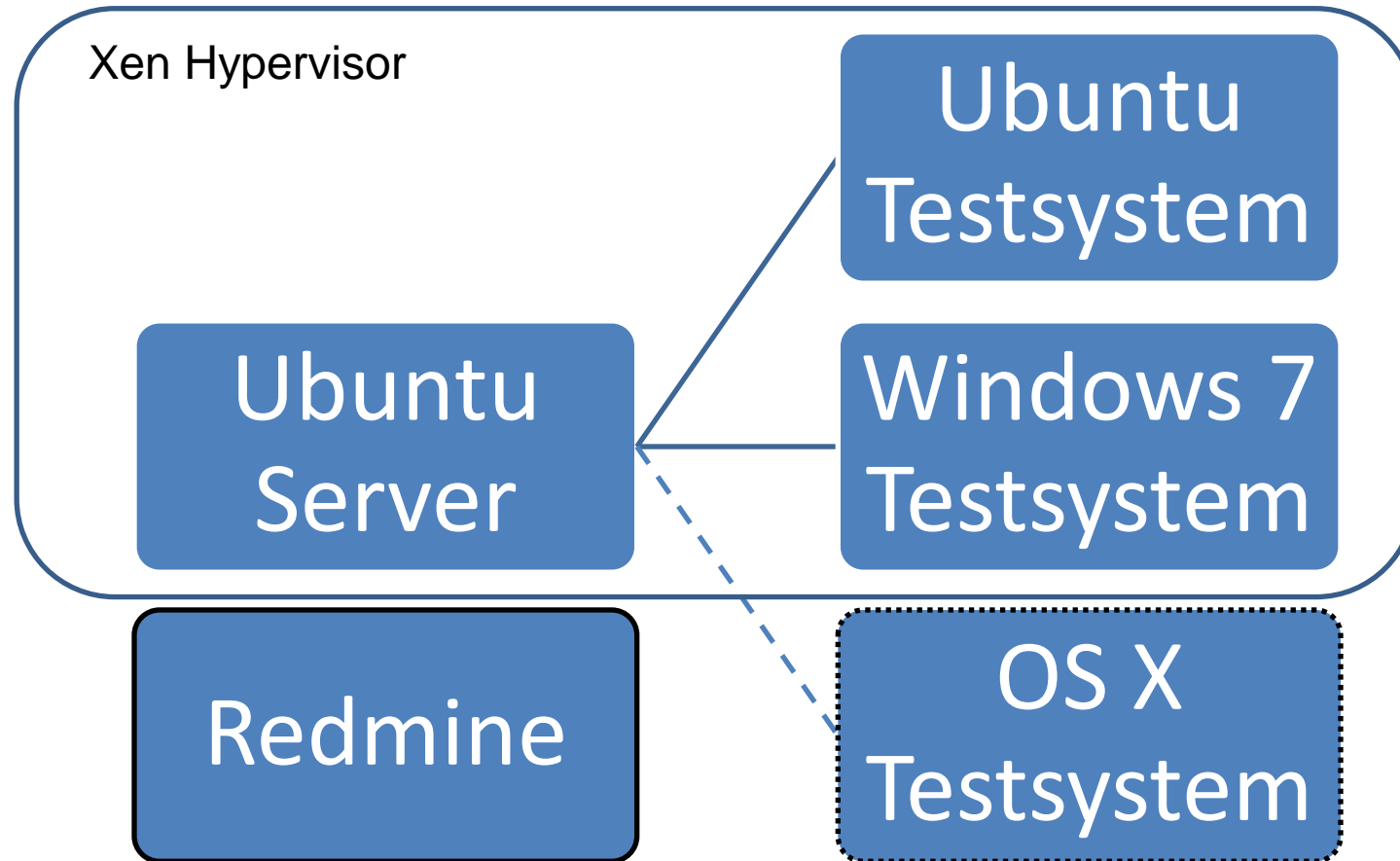
Practices:

- Single repository
- Automatic build
- Self-testing build
- Everyday everyone commits to mainline
- Every commit has to be build on the integration machine
- Fast Build
- Test environment = clone of producton environment
- Easy accessible executable
- Public result
- Automatic Deployment

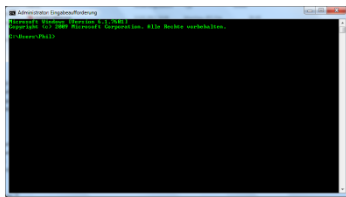
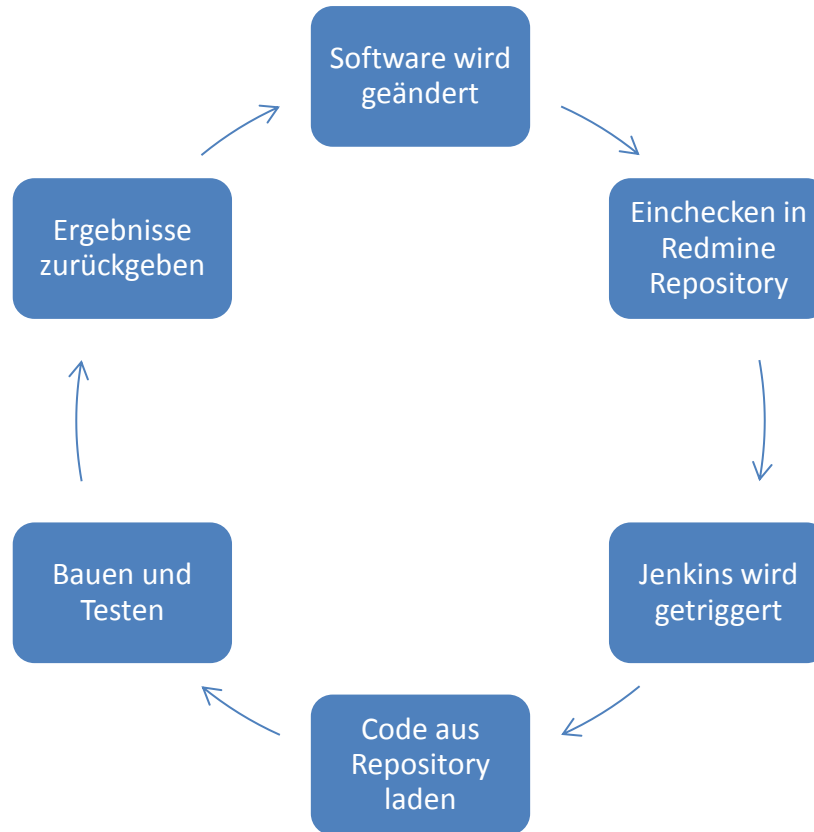
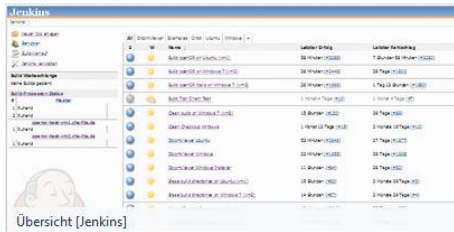
Gliederung

1. Einleitung
2. Relevanz im medizinischen Bereich
3. Continuous Integration
- 4. Continuous Integration System des BZMM**
5. openOR Buildsystem
6. openOR Testsystem
7. Ergebnisse
8. Snapshot: Build Tool Chain

Continuous Integration System des BZMM



Continuous Integration System des BZMM



Gliederung

1. Einleitung
2. Relevanz im medizinischen Bereich
3. Continuous Integration
4. Continuous Integration System des BZMM
- 5. openOR Buildsystem**
6. openOR Testsystem
7. Ergebnisse
8. Snapshot: Build Tool Chain

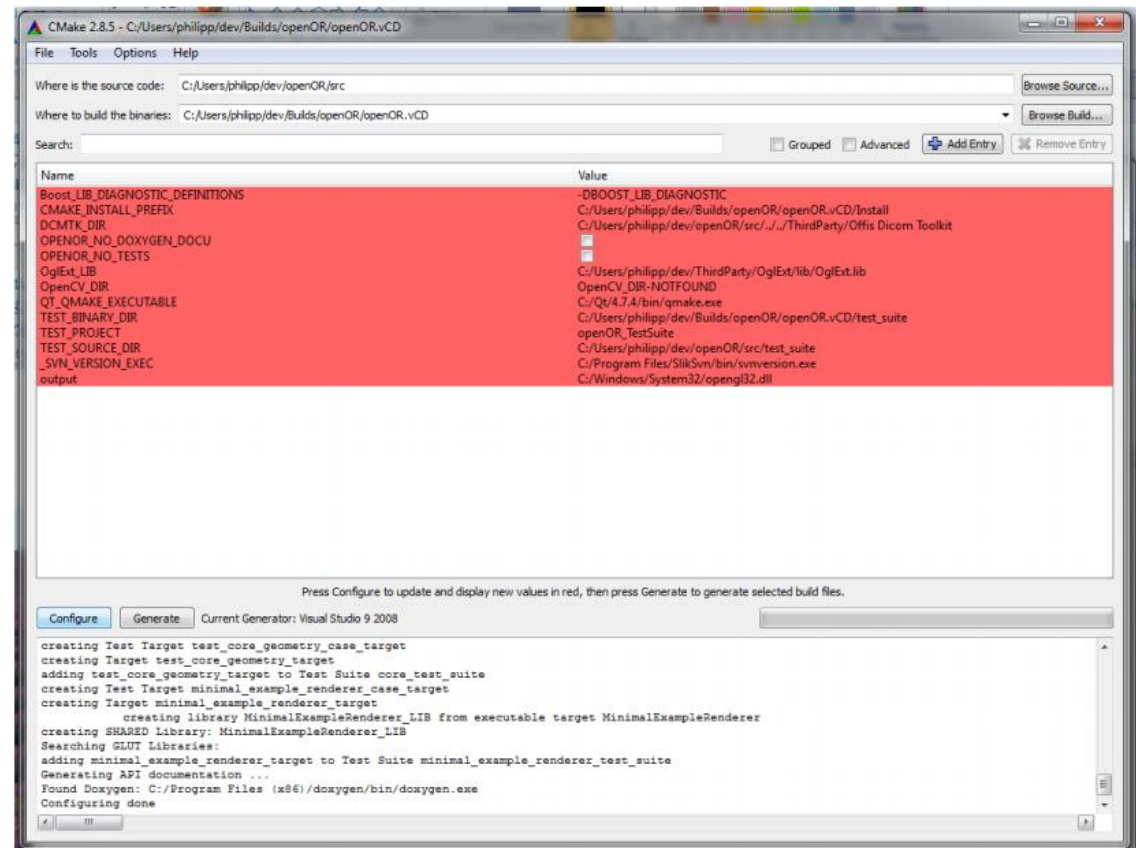
openOR

- plattformunabhängiges, modulares Framework für medizinische Anwendungen
- einfache Benutz- und Erweiterbarkeit



openOR Build System

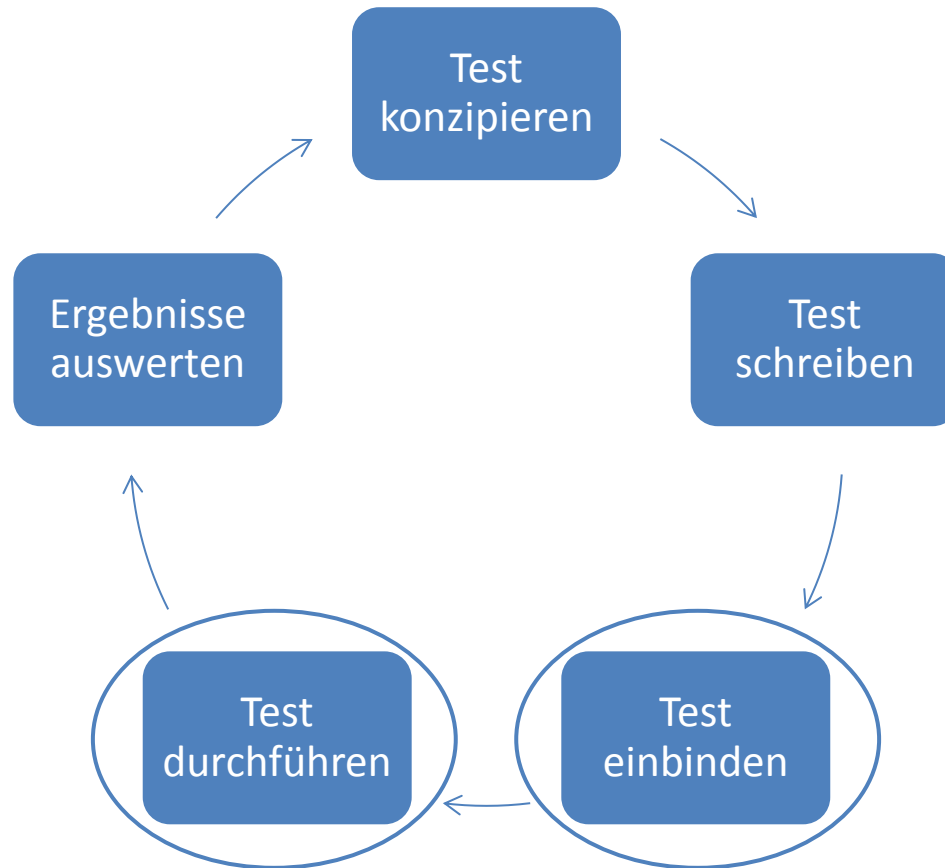
- in CMake entwickelt
- Abhängigkeiten ermitteln und auflösen



Gliederung

1. Einleitung
2. Relevanz im medizinischen Bereich
3. Continuous Integration
4. Continuous Integration System des BZMM
5. openOR Buildsystem
6. openOR Testsystem
7. Ergebnisse
8. Snapshot: Build Tool Chain

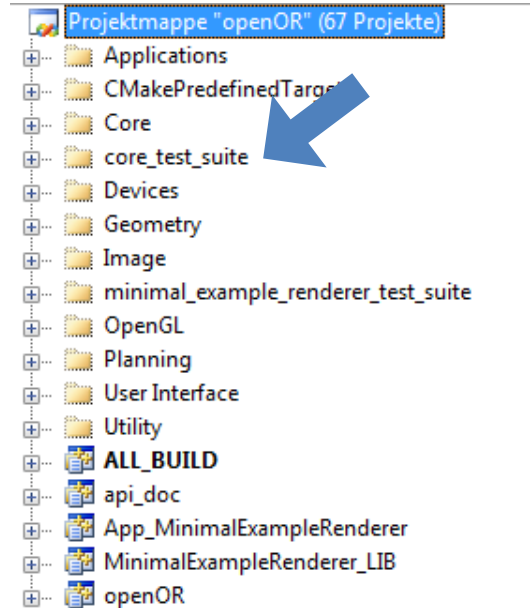
openOR Testsystem



openOR Testsystem

Test Suiten

- Besteht aus Test Targets



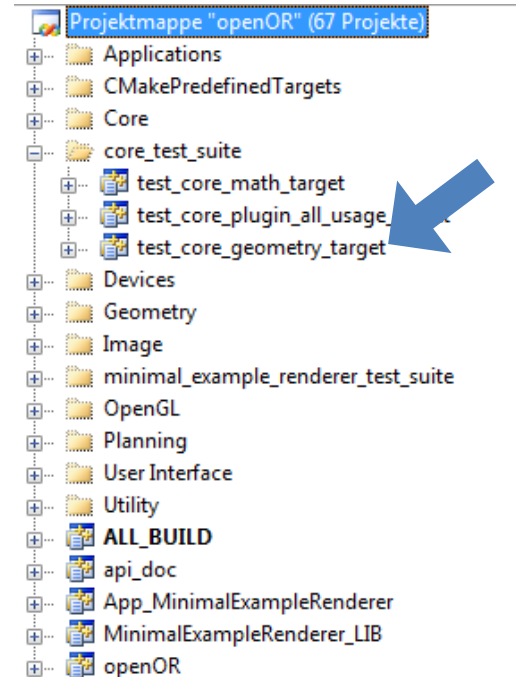
openOR Testsystem

Test Suiten

- Besteht aus Test Targets

Test Targets

- Bestehen aus Test Cases



openOR Testsystem

Test Suiten

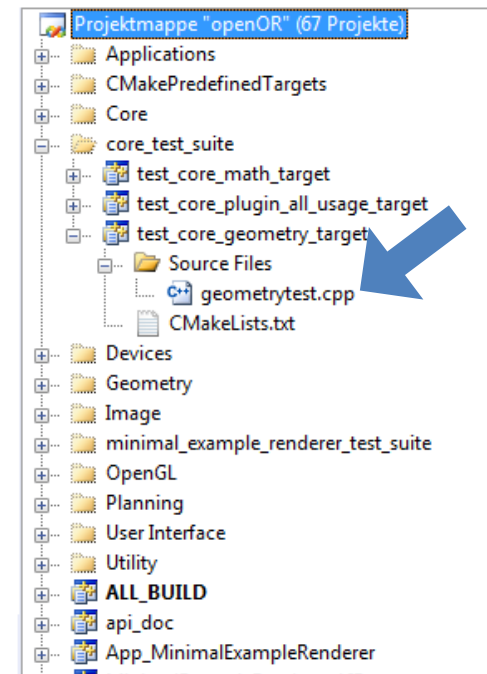
- Besteht aus Test Targets

Test Targets

- Bestehen aus Test Cases

Test Cases

- Bestehen aus Tests
- Bindet Tests ein



openOR Testsystem

Test Suiten

- Besteht aus Test Targets

Test Targets

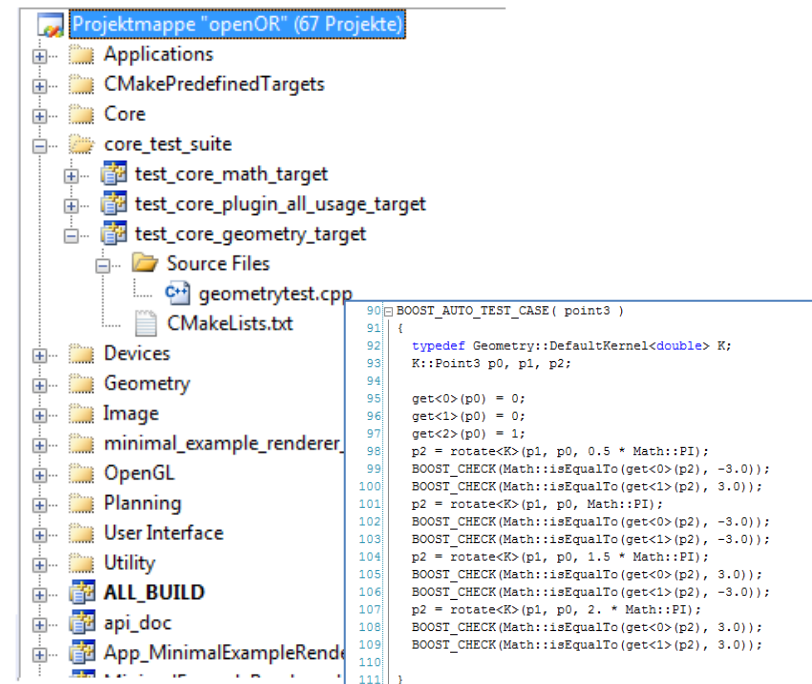
- Bestehen aus Test Cases

Test Cases

- Bestehen aus Tests
- Bindet Tests ein

Tests

- Eigentlicher Testcode

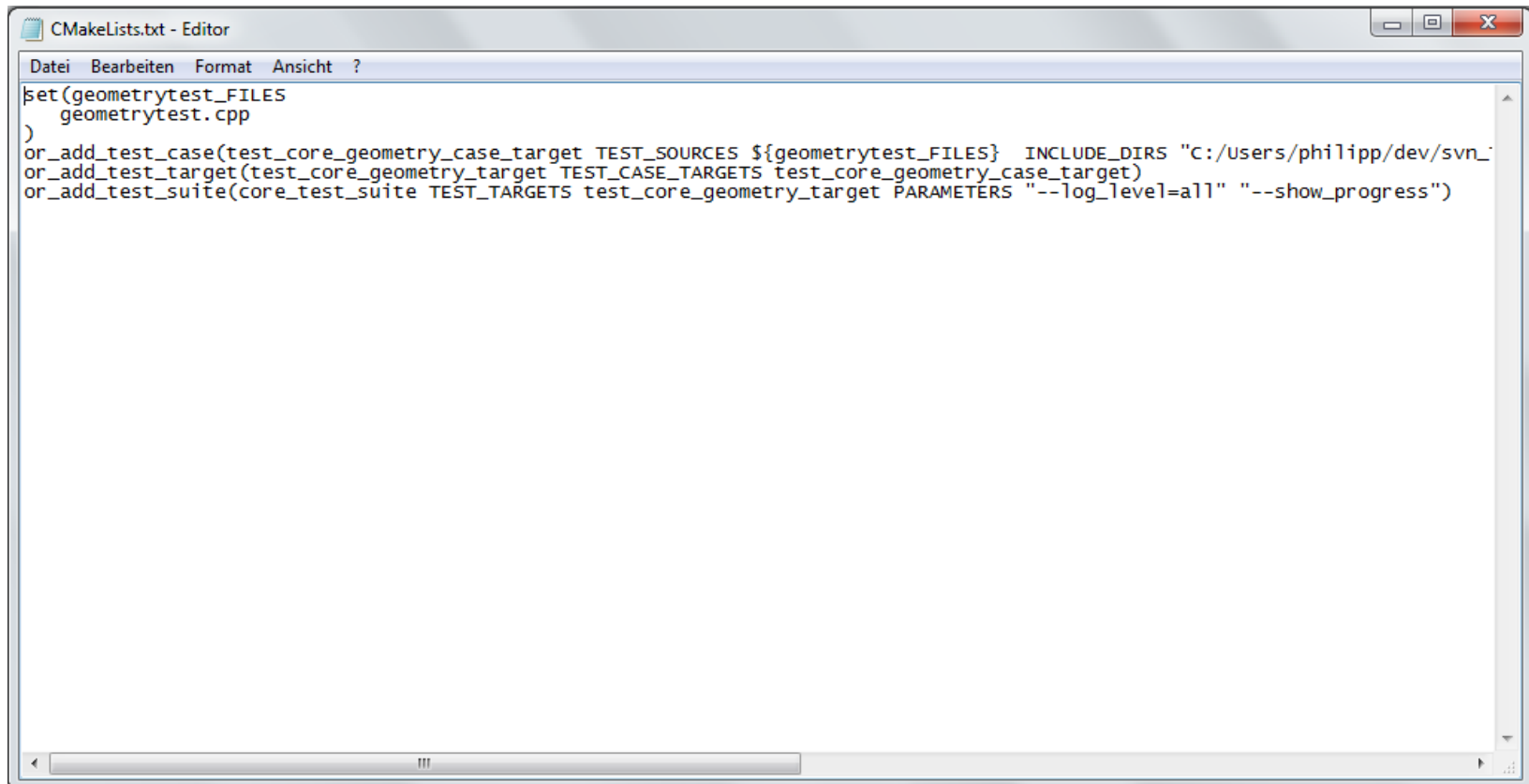


```

90 BOOST_AUTO_TEST_CASE( point3 )
91 {
92     typedef Geometry::DefaultKernel<double> K;
93     K::Point3 p0, p1, p2;
94
95     get<0>(p0) = 0;
96     get<1>(p0) = 0;
97     get<2>(p0) = 1;
98     p2 = rotate<K>(p1, p0, 0.5 * Math::PI);
99     BOOST_CHECK(Math::isEqualTo(get<0>(p2), -3.0));
100    BOOST_CHECK(Math::isEqualTo(get<1>(p2), 3.0));
101    p2 = rotate<K>(p1, p0, Math::PI);
102    BOOST_CHECK(Math::isEqualTo(get<0>(p2), -3.0));
103    BOOST_CHECK(Math::isEqualTo(get<1>(p2), -3.0));
104    p2 = rotate<K>(p1, p0, 1.5 * Math::PI);
105    BOOST_CHECK(Math::isEqualTo(get<0>(p2), 3.0));
106    BOOST_CHECK(Math::isEqualTo(get<1>(p2), -3.0));
107    p2 = rotate<K>(p1, p0, 2. * Math::PI);
108    BOOST_CHECK(Math::isEqualTo(get<0>(p2), 3.0));
109    BOOST_CHECK(Math::isEqualTo(get<1>(p2), 3.0));
110
111 }

```

openOR Testsystem



```

CMakeLists.txt - Editor
Datei Bearbeiten Format Ansicht ?
set(geometrytest_FILES
  geometrytest.cpp
)
or_add_test_case(test_core_geometry_case_target TEST_SOURCES ${geometrytest_FILES} INCLUDE_DIRS "C:/Users/philipp/dev/svn_
or_add_test_target(test_core_geometry_target TEST_CASE_TARGETS test_core_geometry_case_target)
or_add_test_suite(core_test_suite TEST_TARGETS test_core_geometry_target PARAMETERS "--log_level=all" "--show_progress")
  
```

Gliederung

1. Einleitung
2. Relevanz im medizinischen Bereich
3. Continuous Integration
4. Continuous Integration System des BZMM
5. openOR Buildsystem
6. openOR Testsystem
7. Ergebnisse
8. Snapshot: Build Tool Chain

Ergebnisse

- Anfangs oft rote Punkte
 - Verbesserung über die Zeit
- weniger Kompatibilitätsprobleme
 - Insbesondere bei Linux

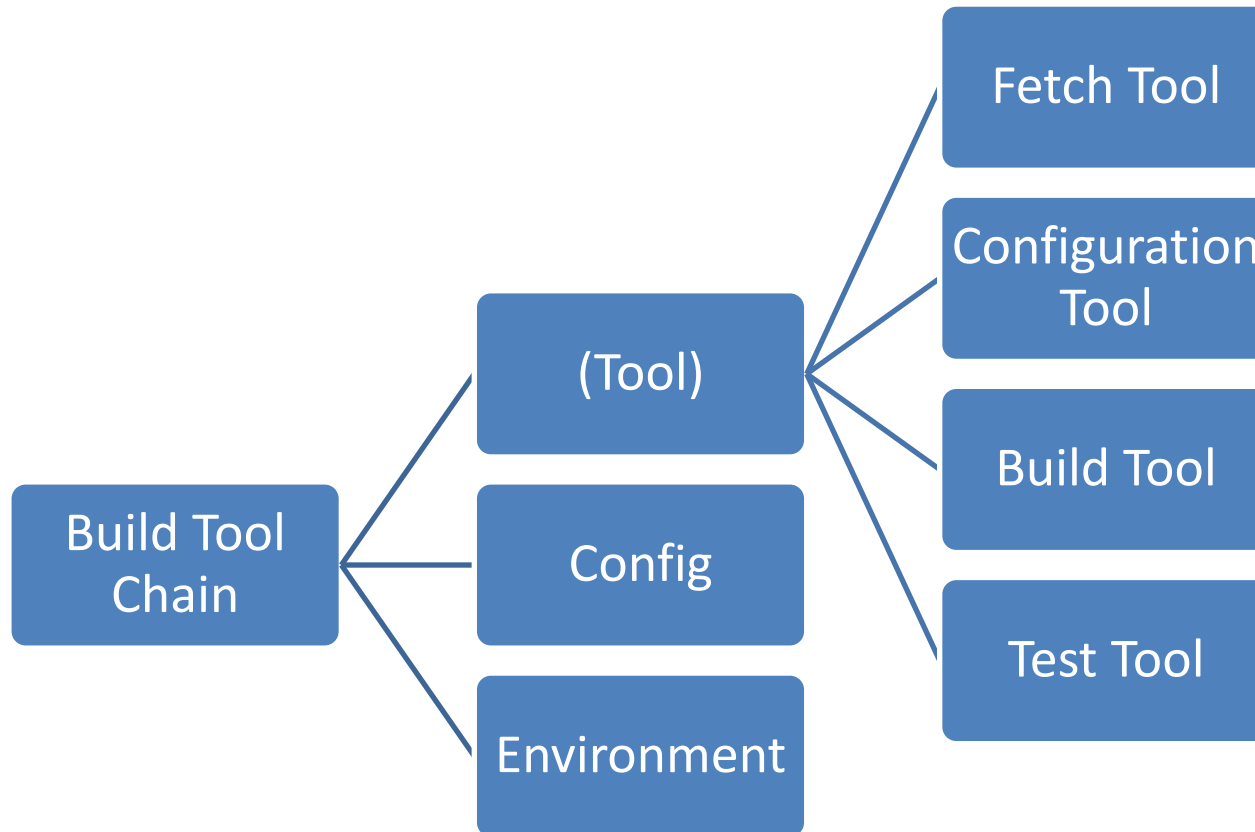
Gliederung

1. Einleitung
2. Relevanz im medizinischen Bereich
3. Continuous Integration
4. Continuous Integration System des BZMM
5. openOR Buildsystem
6. openOR Testsystem
7. Ergebnisse
8. Snapshot: Build Tool Chain

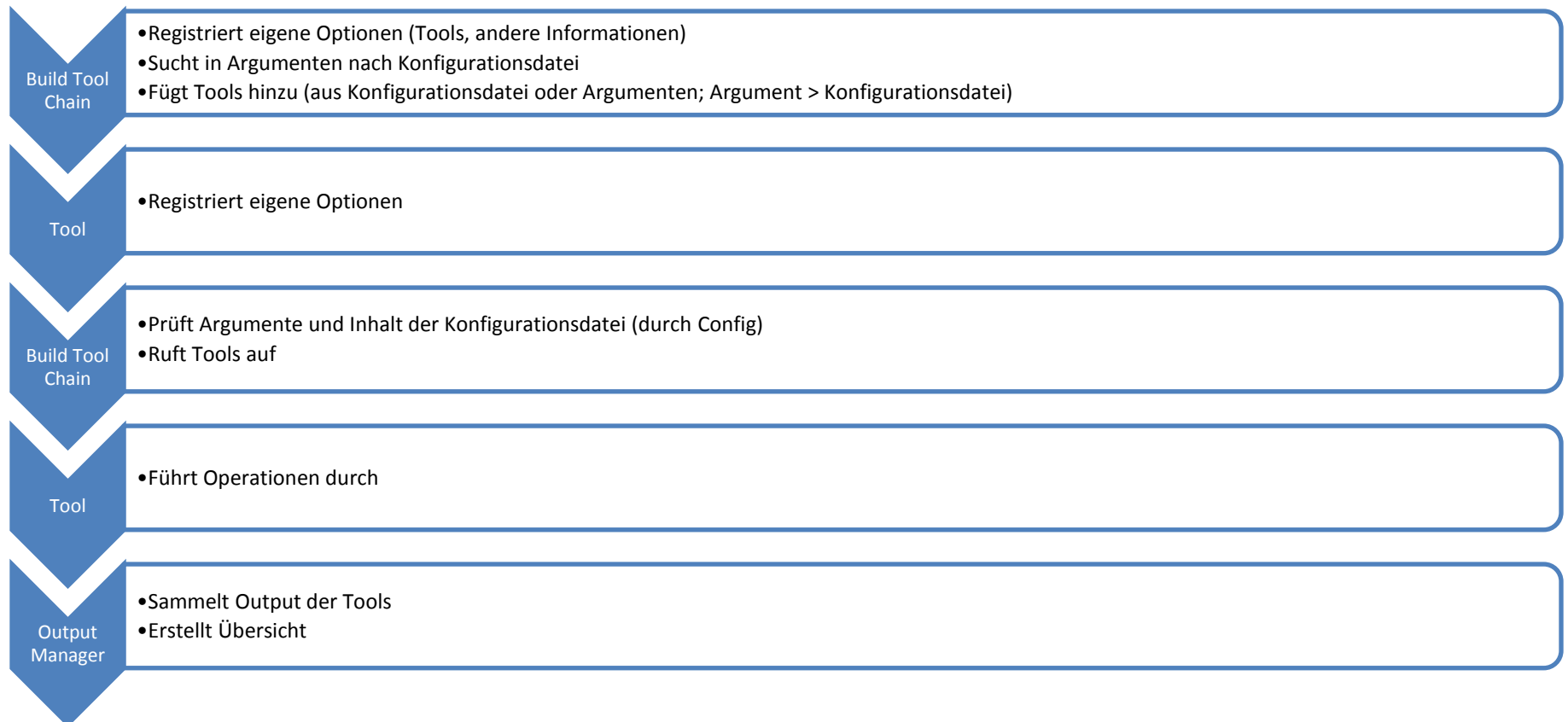
Snapshot: Build Tool Chain

- Aktuelles Projekt in der Implementierung
- Koordiniert Buildprozess
 - Momentan durch Buildskripte für jedes Betriebssystem separat
- Vereinheitlichung der Tools
 - Java -> Jenkins und Maven
 - C++ ?
 - Änderungen momentan teuer und uneinheitlich
 - Zugriff über Konfigurationsdateien und durch Übergabe von Parametern
 - Konfigurationsdateien über SVN Repository abruf- und veränderbar

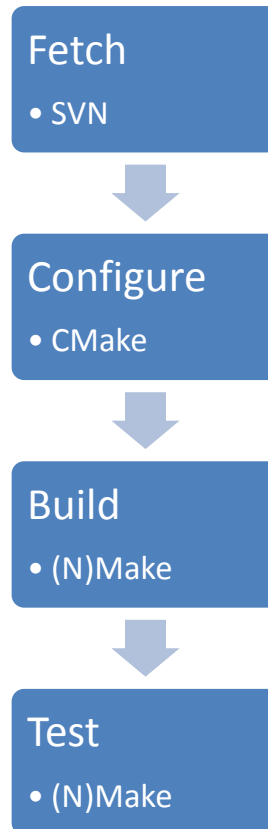
Snapshot: Build Tool Chain



Snapshot: Build Tool Chain



Snapshot: Build Tool Chain



Snapshot: Build Tool Chain

Das geht bisher

- Code auschecken, Konfigurieren, Buildprozess und Tests durchführen
- Optionen durch Konfigurationsdatei und Argumente übergeben

Das geht bisher noch nicht

- Module (generisch) aufrufen

Das ist geplant

- Build- und Testergebnisse geordnet zurückgeben und in Redmine einbinden



Vielen Dank für Ihre Aufmerksamkeit

Fragen?

Diskussion

Continuous Integration bietet hohen Mehrwert. Tatsächlich?

Quellen

Continuous Integration

- Martin Fowler: <http://www.martinfowler.com/articles/continuousIntegration.html>

Quellen

Bilder:

- <http://bzmm.charite.de/typo3temp/pics/756b3e253a.jpg>
- <http://bzmm.charite.de/typo3temp/pics/921eb6b341.jpg>
- http://bzmm.charite.de/fileadmin/user_upload/microsites/ohne_AZ/m_cc09/bzmm/bilder/Kooperationen/scopis/SCOPIS_System_middle.jpg
- <http://bzmm.charite.de/typo3temp/pics/6dddb7d00d.jpg>
- http://bzmm.charite.de/fileadmin/user_upload/microsites/ohne_AZ/m_cc09/bzmm/ogog/fraunhofer_logo.jpg
- https://upload.wikimedia.org/wikipedia/commons/e/e3/Jenkins_logo_with_title.svg
- <https://upload.wikimedia.org/wikipedia/en/7/79/Subversion.png>
- https://upload.wikimedia.org/wikipedia/commons/3/3f/Redmine_logo.svg