

intersoft:

intersoft AG

Continuous Integration



Der einfache Weg zu
besserer Software

Inhalt

- ▶ Was ist Continuous Integration?
- ▶ Was soll Continuous Integration?
- ▶ Wie geht Continuous Integration?
- ▶ Wie sieht Continuous Integration in der Praxis aus?
- ▶ Was sind die Probleme bei Continuous Integration?

Martin Fowler



[http://www.martinfowler.com/articles/
continuousIntegration.html](http://www.martinfowler.com/articles/continuousIntegration.html)

Was ist Continuous Integration?

- ▶ Extreme Programming fordert jede Änderung auf einem Integrationsrechner zu integrieren
- ▶ Das Integrieren wird automatisiert
- ▶ Es wird der aktuellste Sourcezustand integriert
- ▶ Automatisches Testen des Builds

Integrations Serversoftware

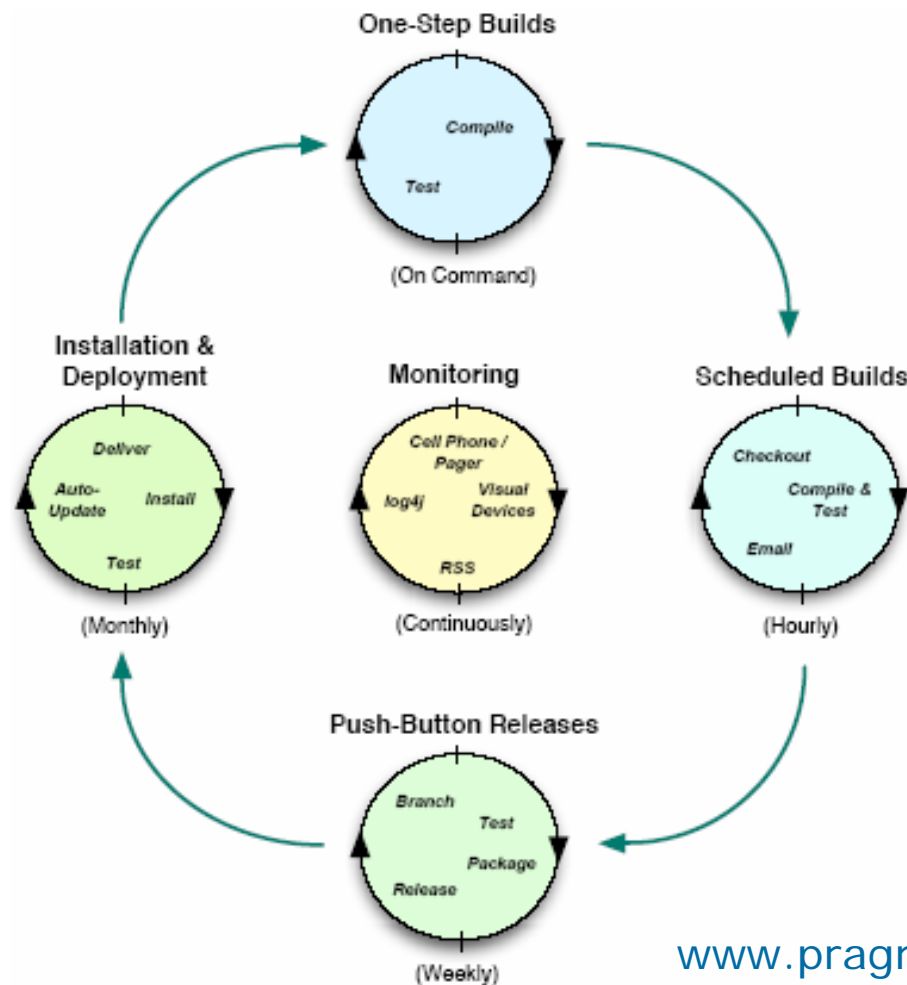
- [CruiseControl](#) - ThoughtWorks
- [CruiseControl.NET](#) - ThoughtWorks
- [DamageControl](#) - ThoughtWorks
- [Tinderbox & Tinderbox2](#) - Mozilla Projekt
- [Continuum](#) - Apache
- [Sin](#) - Tigris
- [BuildBot](#)
- [Anthill Professional](#)
- [BeetleJuice](#)
- [Luntbuild open/professional](#)
- [Gump](#)
- [Draco.NET](#)
- [Parabuild Server](#)
- [Tinderbox3](#)

- **kein Global Player**

Vorteile

- ▶ Frühzeitig Integrationsprobleme aufzeigen
- ▶ Software-Qualität steigern
- ▶ Den Entwickler unterstützen
 - Rückmeldung zur eigenen Arbeit
 - Unterstützung bei der Fehlersuche
 - Entlastung vom Builden
- ▶ Wissenstransfer
 - Jeder kann jeder Zeit ein Release der Software erstellen
- ▶ Jederzeit Überblick über die Codequalität

Die Roadmap zur Buildautomation



www.pragmaticprogrammer.com

Voraussetzungen

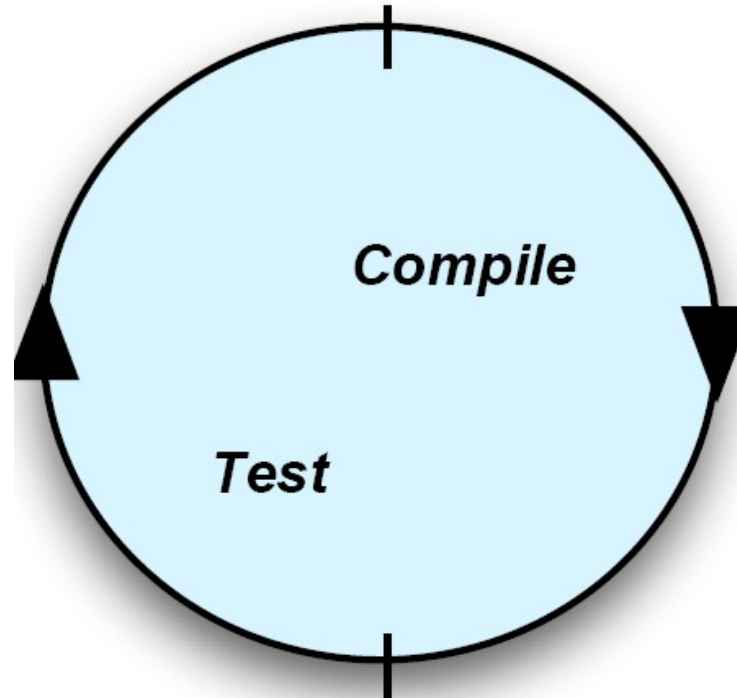
- ▶ Zentrale Codebasis
- ▶ Automatische Tests
- ▶ Leistungsfähiger Buildserver

Tools zur Buildautomation

- › Ant
- › Maven 1 und 2
- › Make
- › nAnt
- › Groovy
- › Python
- › ...

Step1

One-Step Builds



(On Command)

Automatisches Builden

```
<project default="compile" name="basis" >

  <target name="compile" >
    <javac destdir="classes" >
      <src path="src"/>
      <classpath location="xy.jar" />
    </javac>
  </target>

</project>
```

Automatisches Testen

```
<target name="test" depends="compile " >

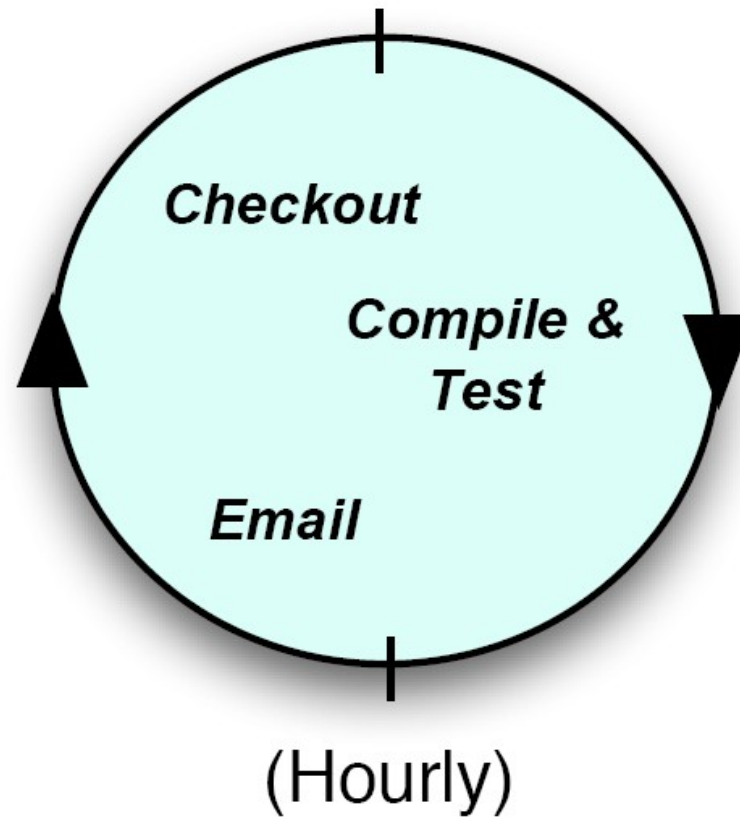
  <junit printsummary="yes" >
    <classpath>
      <pathelement location="classes"/>
      <pathelement location="xy.jar"/>
    </classpath>
    <formatter type="xml"/>

    <batchtest todir="reports">
      <fileset dir="classes">
        <include name="**/*TestSuite.class"/>
      </fileset>
    </batchtest>
  </junit>

</target>
```

Step 2

Scheduled Builds



Build Automatisieren

▸ cron job

```
<target name=„checkout“ >
```

```
  <svn javahl="true">
```

```
    <checkout url="server" destpath="src"/>
```

```
  </svn>
```

```
</target>
```

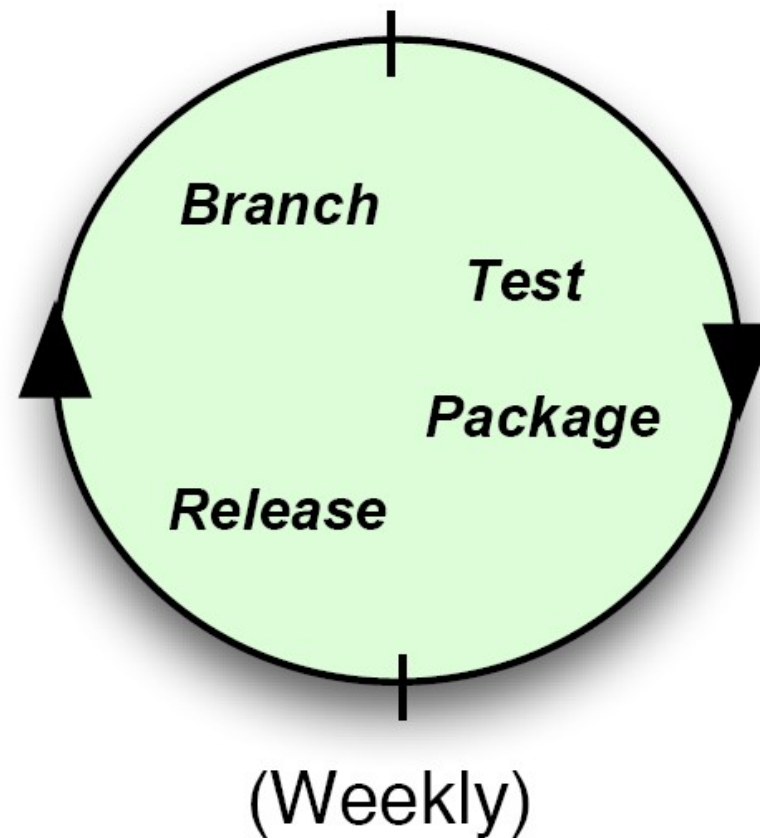
```
<target name=„email“ >
```

```
  <mail tolist=„allDev“ from=„build“ subject="build 1 fertig"/>
```

```
</target>
```

Step 3

Push-Button Releases



Release Automatisieren

▸ Branch/Tag

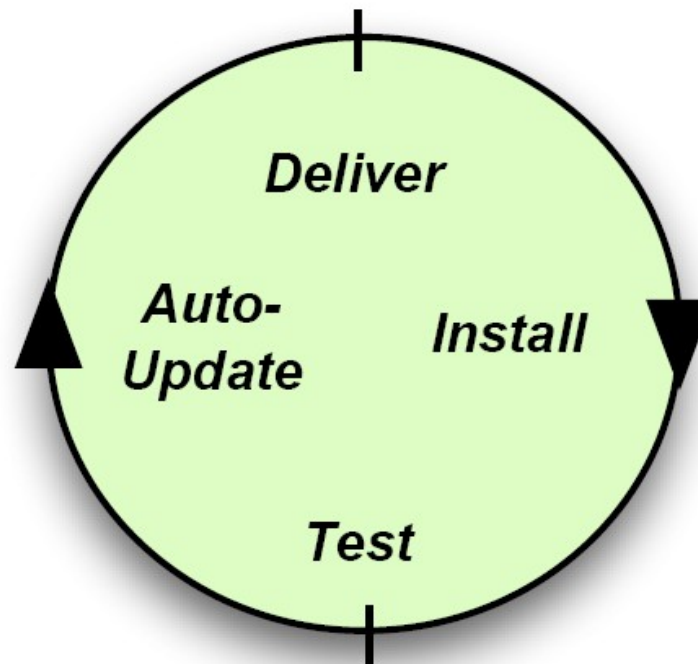
```
<target name="tag" >  
  <svn javahl="true" >  
    <copy srcPath="src" message="build_1"  
      destUrl="server/tags/build_1" />  
  </svn>  
</target>
```

▸ Jar / ear Task

```
<target name="package" >  
  <jar destfile="basis.jar" basedir="classes" />  
</target>
```


Step 4

Installation & Deployment



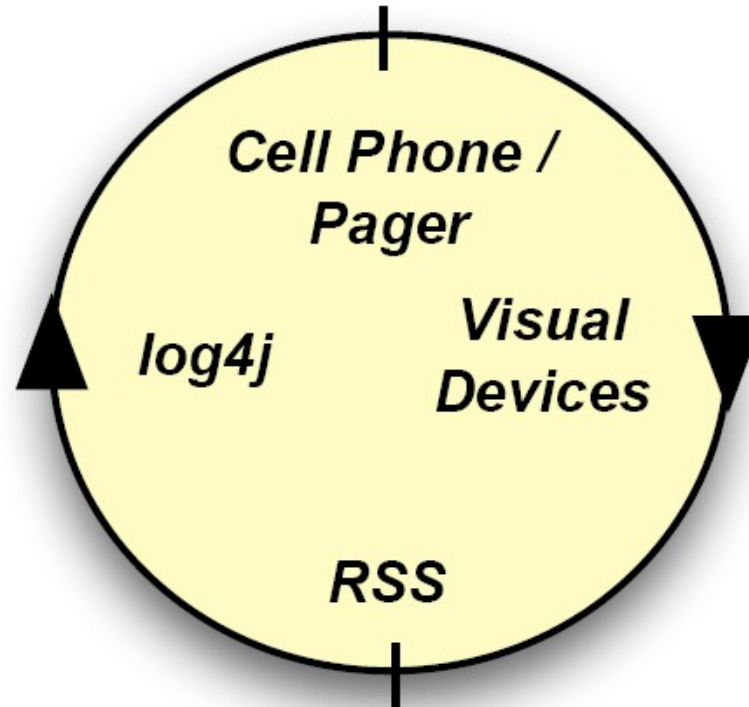
(Monthly)

Deployment Automatisieren

- ▶ ftp-Task
- ▶ DBUnit-Test
- ▶ Anwendungstest

Die Kür

Monitoring



(Continuously)

Was bieten Continuous Integration Server?

- ▶ Zeitsteuerung
- ▶ Modifikationssteuerung
- ▶ Email-Benachrichtigung
 - Gruppen
 - Entwickler die etwas geändert haben
- ▶ Web-Statusübersicht
- ▶ RSS-Feeds
- ▶ Instant Messaging
- ▶ XFD z.B. Lavalampen
- ▶ Deployment

Welchen Server nehmen?

- Geschmackssache
- Die meisten sind spezialisiert
 - Ant
 - .net
 - SVN
- intersoft AG nutzt CruiseControl
- CruiseControl ist sehr umfassend
- CruiseControl Oberfläche hat Nachholbedarf
 - Das aktuelle Release hat sehr aufgeholt

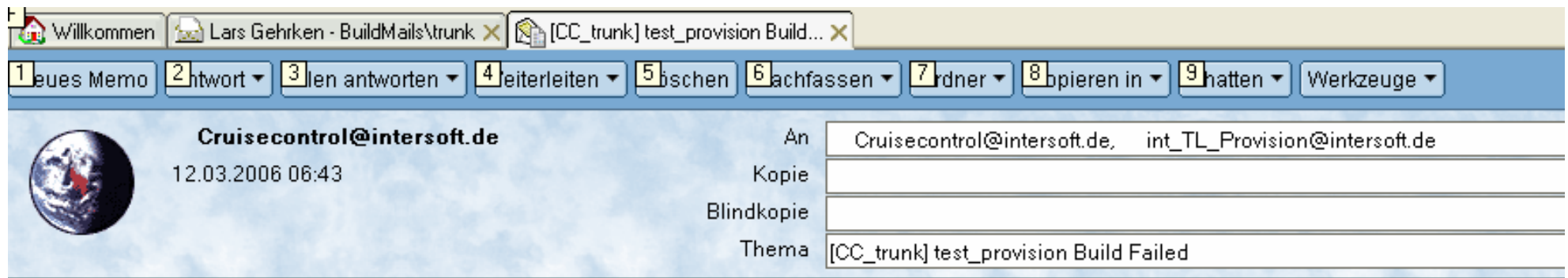
Konfiguration

```
<project name="basis" buildafterfailed="false">
  <bootstrappers>
    <svnbootstrapper localWorkingCopy="../${branch.name}/module/build_tools"/>
  </bootstrappers>
  <modificationset quietperiod="120">
    <svn RepositoryLocation="${svnservlet}/lifestream/${branch.path}/${project.name}"/>
    <buildstatus logdir="../cc_logs/util"/>
  </modificationset>
  <schedule interval="3600">
    <ant target="dist" buildfile="../${branch.name}/module/build_tools/release.xml" multiple="1" />
  </schedule>
  <log dir="../cc_logs/${project.name}" encoding="ISO-8859-1">
    <merge dir="../build/cvs_release_basis/build/basis/reports/test-result"/>
  </log>
  <publishers>
    <email mailhost="" returnaddress="Cruisecontrol@intersoft.de" defaultsuffix="@intersoft.de" subjectprefix="[CCtrunc]"
      buildresultsurl="http://cruise.intersoft-ag.de/cruisecontrol/buildresults/${project.name}">
      <failure address="int_TL_Basis@intersoft.de" reportWhenFixed="true"/>
    </email>
  </publishers>
</project>
```

Wie sieht Continuous Integration in der Praxis aus?

- ▶ [Übersichtsseite](#)
- ▶ Detailansicht

Wie sieht Continuous Integration in der Praxis aus?



View results here ->

http://cruise.intersoft-ag.de:8901/cruisecontrol/buildresults/test_provision?log=log20060312063830

Probleme in der Praxis

- ▶ Die Entwickler verlassen sich auf CruiseControl
- ▶ Unfertiger Code blockiert andere
- ▶ Lange Laufzeiten des Builds
- ▶ Abhängigkeiten zwischen den Modulen
 - Es muss in der richtigen Reihenfolge gebaut werden
 - Paralleles bauen birgt Timingprobleme
- ▶ Häufige falsche Fails, lassen Entwickler nachlässig werden
- ▶ Blockierte Buildqueue

Probleme in der Praxis

- ▶ ant ist keine Programmiersprache
 - Fehlerbehandlung schwierig
 - oft manuelle Eingriffe nötig
 - schnell schlecht wartbar
- ▶ Externe Tools lassen sich oft schwer integrieren
- ▶ Unittests die keine sind

Unittests die keine sind

I have these rules that I use for unit tests, primarily because I encounter so many teams that start writing end to end tests, call them unit tests, and give up because "testing takes too long".

To me, a test is not a unit test if:

- It talks to a database
- Communicates across a network
- Touches the file system
- You can't run it at the same time as any of your other unit tests.
- You have to do special things to your environment to run it (like editing configuration files).

Tests that do those things are okay, but to me they aren't unit tests, and they should be segregated from true unit tests.

http://www.developertesting.com/archives/individual_weblogs-jeffrey_fredrick-index.html

Ausbaumöglichkeiten

- Erweitern der Überprüfungen
 - Codestyle prüfen
 - Testabdeckung prüfen
- Coolere Ergebnis anzeigen
- Verteiltes Builden
- Scriptsprache statt oder mit ant
- Dependency Management
 - ivy
 - maven
- Autodeployment

Weitere Informationen

- ▶ <http://www.martinfowler.com/articles/continuousIntegration.html>
- ▶ <http://cruisecontrol.sourceforge.net>
- ▶ Pragmatic Project Automation ISBN 0-9745140-3-9
- ▶ <http://www.oio.de/cruisecontrol.htm>
- ▶ <http://damagecontrol.codehaus.org/Continuous+Integration+Server+Feature+Matrix>

Fragen

