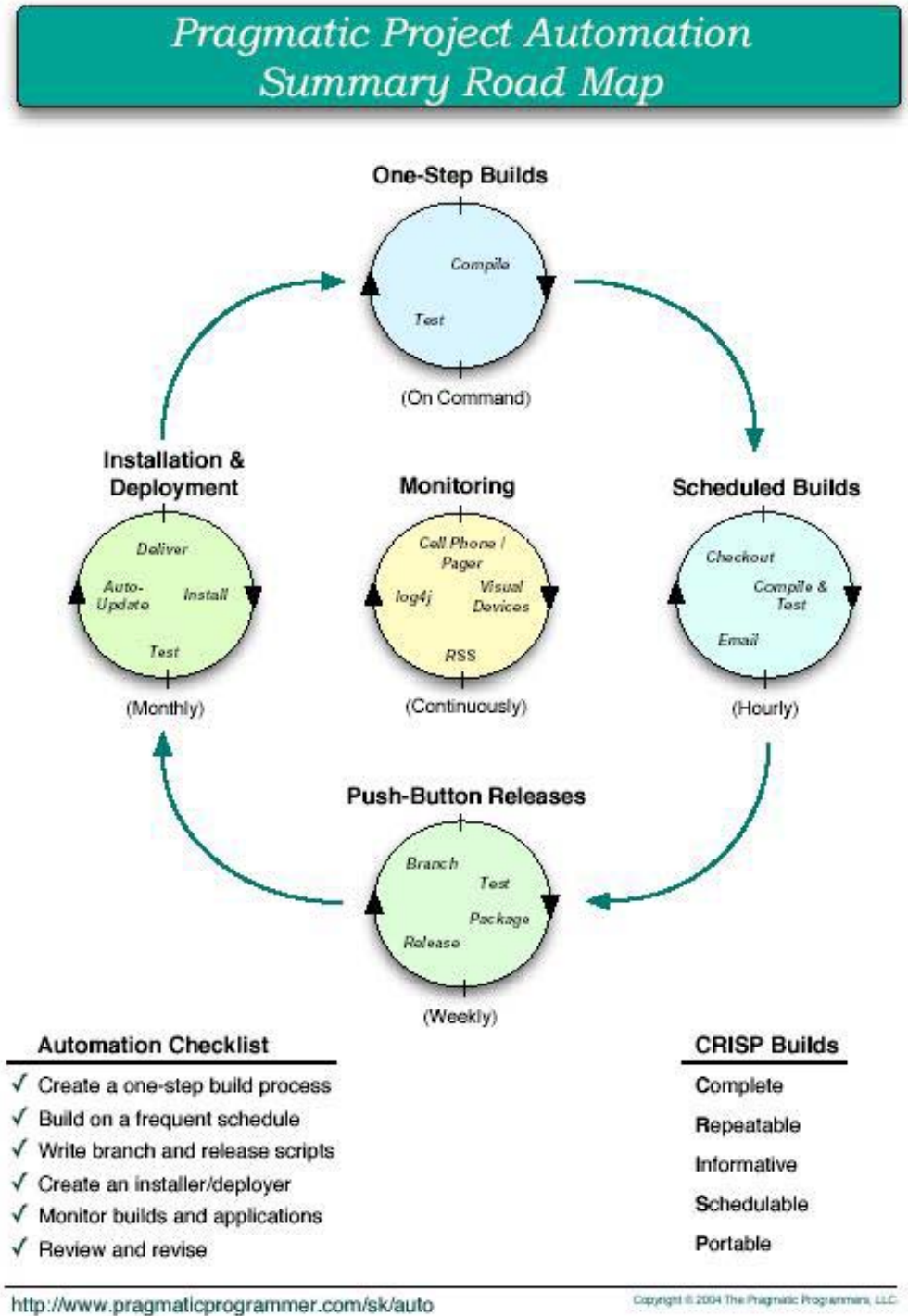
# Continuous Integration

Dr. Christoph Steindl

# The Big Picture

- Building # Compiling
- Integrate early
- Integrate often
- Make the build CRISP
- Automate
  (e.g. with Ant or Maven)
- Verify the build with tests
- Trigger build by schedule
  or event (like modified
  sources in the repository)
- Build continuously
  (e.g. with CruiseControl)



# Prerequisites and Benefits

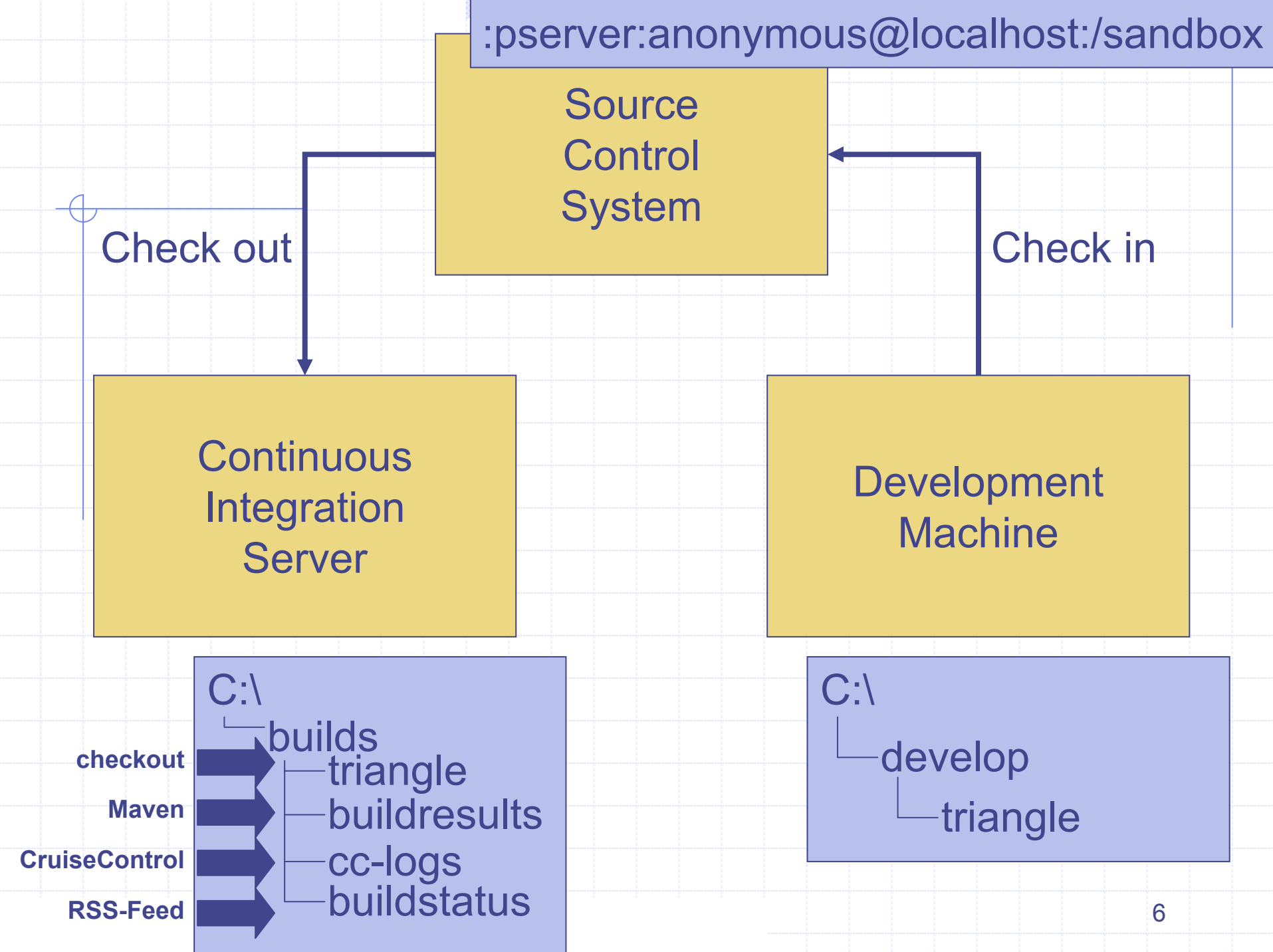
- Prequisites:
  - Keep a single place where all the source code lives and where anyone can obtain the current sources from (and previous versions).
  - Automate the build process so that anyone can use a single command to build the system from the sources.
  - Automate the testing so that you can run a good suite of tests on the system at any time with a single command.
  - Make sure anyone can get a current executable which you are confident is the best executable so far.
- Benefits
  - Most integration bugs manifest themselves the same day they were introduced.
  - Additionally you typically know where to look for the reason of the problem (or you don‘t add the buggy feature to the product).

# CruiseControl

- Is a continuous integration server
  - Runs in the background
  - Runs the “build cycle” (at configurable intervals):
    - Reload configuration file (in case of changes)
    - Determine if a build is necessary (by querying the source control system)
    - Build
      - Check out the project files
      - Compile the sources
      - Execute the tests
      - Package the deliverables
    - Create a log file
    - Send notifications of success or failure
- Presents the build results as a web page (by the “build results JSP” which can be deployed into a Tomcat server).
- Integrates with Junit, Ant, Maven and CVS (and other source control systems).

# How to

- On the build server
  - Create a directory for the builds (e.g. „mkdir c:\builds“)
  - Change into the directory (e.g. „cd c:\builds“)
  - Check out the project (e.g. „cvs co triangle“)
  - Create a directory for the build results (optionally on another server, e.g. „mkdir c:\builds\buildresults“)
    ➔ this is where Maven will deploy the results of the build
  - Create a directory for the log files (e.g. „mkdir c:\builds\cc-logs“)
    ➔ this is where CruiseControl will create its log files
  - Let maven create the cruisecontrol.xml (e.g. „maven cruisecontrol“)
  - Add additional publishers (e.g. for RSS feeds)
  - Let maven start CruiseControl (e.g. „maven cruisecontrol:run“)
- On the developer‘s machine
  - Develop the system
  - Check in code
- The source control system acts as the intermediary between the developer and the build server.
- Relax and wait for email notifications to arrive.


:pserver:anonymous@localhost:/sandbox
Source Control System
Check out
Check in
Continuous Integration Server
Development Machine
C:\
builds
triangle
buildresults
cc-logs
buildstatus
checkout
Maven
CruiseControl
RSS-Feed
C:\
develop
triangle

# CruiseControl.xml (1/3)

```
<?xml version="1.0" encoding="UTF-8"?>

<cruisecontrol>
  <project name="testing-triangle">
    <bootstrappers>
      <currentbuildstatusbootstrapper
        file="C:\builds/cc-logs/currentbuildstatus.txt">
      </currentbuildstatusbootstrapper>
    </bootstrappers>
    <modificationset>
      <cvs localWorkingCopy="C:\builds/triangle"
        cvsroot=":pserver:anonymous@localhost:/sandbox">
      </cvs>
    </modificationset>
  </project>
</cruisecontrol>
```

- Name
- Bootstrappers
- Modification set

# CruiseControl.xml (2/3)

```
<schedule interval="60">
  <maven goal="scm:update-project|clean test|site site:fsdeploy"
    projectfile="C:\builds/triangle/project.xml"
    mavenscript="C:\PROGRA~1\APACHE~1\MAVEN1~1.0/bin/maven">
  </maven>
</schedule>
<log dir="C:\builds/cc-logs/testing-triangle">
  <merge dir="C:\builds/triangle/target/test-reports">
  </merge>
</log>
```

- Schedule interval
- Maven for build
- Log

# CruiseControl.xml (3/3)

```
<publishers>
  <currentbuildstatuspublisher
    file="C:\builds/cc-logs/currentbuildstatus.txt">
  </currentbuildstatuspublisher>
  <htmlemail logdir="C:\builds/cc-logs/testing-triangle"
    mailhost="localhost"
    css="C:\cruisecontrol-2.1.6/reporting/jsp/css/cruisecontrol.css"
    subjectprefix="[BUILD]" returnaddress="christoph_steindl@at.ibm.com"
    defaultsuffix="@localhost"
    xsldir="C:\cruisecontrol-2.1.6/reporting/jsp/xsl">
    <map address="christoph_Steindl@at.ibm.com" alias="steindl">
    </map>
    <failure address="christoph_steindl@at.ibm.com">
    </failure>
  </htmlemail>
  <XSLTLogPublisher directory="c:\builds\buildstatus„
    outfilename="trianglebuildstatus.rss" xsltfile="buildstatus.xsl" />
</publishers>
```

- Publishers
- Email, RSS
- Mapping for names

# Maven / CruiseControl files

- Maven‘s „project.xml“ (under version control)
  - Deploy the build result with the target „site:fsdeploy“ (file system deployment)
    <siteDirectory>c:\builds\buildresults</siteDirectory>
  - You can also deploy the results to a web server (with „site:deploy“)
- Maven‘s „build.properties“ (specific for each developer / machine)
  - maven.cruisecontrol.checkout.dir=C:\\builds
  - maven.cruisecontrol.logs.dir=c:\\builds\\cc-logs
- CruiseControl‘s „cruisecontrol.xml“
  - Generated by Maven (maven cruisecontrol)
  - Added RSS publisher (under <publishers>):
    <XSLTLogPublisher directory="c:\builds\buildstatus" outfilename="trianglebuildstatus.rss" xsltfile="buildstatus.xsl" />

```
maven.username=Administrator

#Location of cruise control installation - for finding the reporting directory
maven.cruisecontrol.home=C:\\cruisecontrol-2.1.6

#Where to checkout/update code temporarily. Default value is ${basedir}/checkout.
maven.cruisecontrol.checkout.dir=C:\\builds

#Seconds in between builds for this project. Default value is 300.
maven.cruisecontrol.schedule.interval=60

#Default value is scm:update-project|clean test|site:deploy.
maven.cruisecontrol.goals=scm:update-project|clean test|site site:fsdeploy

#Default value is the domain name of the first email in ${pom.build.nagEmailAddress}.
maven.cruisecontrol.mail.defaultsuffix=@localhost

#Default value is ${maven.build.dir}/cc-logs.
maven.cruisecontrol.logs.dir=c:\\builds\\cc-logs

#Default value is localhost.
#maven.cruisecontrol.mail.host=
#Default value is [BUILD].
#maven.cruisecontrol.mail.subjectprefix=
#Config file to update/create. Default value is ${basedir}/cruisecontrol.xml.
#maven.cruisecontrol.config=
#Template file to use in generating the cruisecontrol.xml file.
#maven.cruisecontrol.template=
```

- User name
- Home of CruiseContro
- Schedule interval
- Goals for build
- Mail suffix

# eXtreme Feedback Devices (XFDs)

- XFDs are inexpensive to build and operate, they add fun and color to the workspace, and most important they are effective in providing the team with feedback on key items and getting them to act upon it.
- XFDs help you to achieve and maintain focus on what is most important to the organization at any given time.
- By making the feedback mechanism fun you help to draw attention to it.
- By making the feedback broadly available you send out a signal that the objectives are important and that many people care about, and depend upon, their achievement.



# Additional XFDs


Photo 3: Lava Bubbles = Build Troubles

# Quote of the day

„By the time it was released, Microsoft Windows NT 3.0 consisted of 5.6 million lines of code spread across 40,000 source files. A complete build took as many as 19 hours on several machines, but the NT development team still managed to build every day (Zachary, 1994). Far from being a nuisance, the NT team attributed much of its success on that huge project to their daily builds.“

Steve McConnell

http://www.stevemcconnell.com/ieeesoftware/bp04.htm

# References


- Mike Clark: Pragmatic Project Automation, Pragmatic Bookshelf, 2004.
- Martin Fowler: Continuous Integration
  http://www.martinfowler.com/articles/continuousIntegration.html
- Continuous Integration with CruiseControl.NET and Draco.NET
  http://www.theserverside.net/articles/showarticle.tss?id=ContinuousIntegration

# Online References

- Continuous Build Environment:
  - CruiseControl: http://cruisecontrol.sourceforge.net/
  - CruiseControl.NET: http://ccnet.thoughtworks.com/
  - DamageControl: http://damagecontrol.codehaus.org/
  - Draco.NET: http://draconet.sourceforge.net/
  - Anthill: http://www.urbancode.com/projects/anthill/default.jsp
  - IntegrationGuard: http://iguard.sourceforge.net/
  - Gump: see Google
- Various Resources:
  - http://c2.com/cgi/wiki?ContinuousIntegration
  - Daily Build and Smoke Test: http://www.stevemcconnell.com/ieeesoftware/bp04.htm
  - XFDs: http://www.developertesting.com/managed_developer_testing/000036.html
  - Continuous Integration WebLog: http://weblogs.asp.net/mnissen/category/5211.aspx
  - Automated Continuous Integration and the Ambient Orb http://blogs.msdn.com/mswanson/articles/169058.aspx
  - Mathmos SoftLight http://www.thinkgeek.com/cubegoodies/lights/6331/