

Continuous Integration on a Dollar a Day

frei nach James Shore (<http://www.jamesshore.com/Blog/Continuous-Integration-on-a-Dollar-a-Day.html>)



Continuous Integration kann sehr einfach sein. Man benötigt dafür keine Software wie CruiseControl oder Jenkins. Es ist so einfach, dass du sofort damit starten kannst. Nimm einen alten ausgedienten Entwicklungsrechner, schließ ihn an einen Monitor an und finde eine Ecke in eurem Teamraum. Richte dort eine Arbeitsumgebung ein und checke den letzten Stand aus der Quellcodeverwaltung aus.

Suche dir ein Build-Maskottchen. Ein mittelgroßes Plüschtier reicht vollkommen aus. Automatisiere deinen Build. Jedes Teammitglied muss in der Lage sein, alle für den Build notwendigen Schritte durch einen einzigen Tastendruck (oder einen einzigen Befehl auf der Kommandozeile) in der richtigen Reihenfolge auszuführen. Ein Tool dafür ist zum Beispiel Gradle. Folge den sieben Schritten.

1 Lokal bauen bis grün



Prüfe deine Änderungen am Code lokal. Ist der Build erfolgreich? Laufen alle Tests noch? Committe deine Änderungen (*git commit*).

2 Integrationstoken reservieren



Nimm dir das Build-Maskottchen. Integrieren darf nur, wer dieses Token besitzt. Damit verhinderst du, dass sich Integrationen gegenseitig behindern und du unnötig mergen musst.

3 Lokal integrieren bis grün



Hole die letzten Updates aus der Quellcodeverwaltung (*git update*). Damit stellst du sicher, dass du keine Änderungen deiner Teammitglieder verpasst. Prüfe, ob der Build lokal durchläuft. Sind deine Änderungen mit den Änderungen der Teammitglieder kompatibel? Laufen noch immer alle Tests?

4 Einchecken



Bringe deine Änderungen in die Quellcodeverwaltung (*git push*).



Sven Günther arbeitet bei der it-agile GmbH als Entwickler und Trainer für agile Entwicklungspraktiken. Er studierte an der Fachhochschule Brandenburg Informatik mit den

Schwerpunkten Softwareentwicklung und Mikroprozessortechnik. Er interessiert sich besonders für agile Prozesse und für testgetriebene Softwareentwicklung. Zur Zeit entwickelt er mit Kollegen die nächste Top-100-iOS-App.

7 Ausliefern (optional)



Alle Änderungen sind integriert, der Build ist erfolgreich und alle Tests laufen. Warum nicht gleich ausliefern? Frühes Feedback ist wichtig und zeigt uns, ob wir auf dem richtigen Weg sind.

6 Bescheid geben



Gib das Integrationstoken frei und informiere deine Teammitglieder über deine Änderungen. Diese können jetzt ihre Arbeitsumgebungen aktualisieren, um auf dem neuesten Stand zu arbeiten und Merges zu verringern.

5 Auf Integrationsserver integrieren bis grün



Aktualisiere die Workarea auf dem Integrationsrechner. Starte dort den Build. Ist alles noch grün, laufen alle Tests? Hast du alles eingchecked?

