Continuous Integration

01. 11. 2016

Geschrieben von

Sascha Becker
Schwarze-Dorn Str.6
30974 Wennigsen
s.becker@wertarbyte.com

Inhaltsverzeichnis

1 Was ist CI?

Continuous Integration, kurz CI, ist ein Entwicklungsverfahren, welches Entwickler dazu drängt mehrmals am Tag bei einem gemeinsamen Repository beizutragen. Jeder Beitrag wird automatisch verifiziert. Dieser Vorgang erlaubt es dem Team Probleme früh zu erkennen. Durch einen regelmäßigen Beitrag der Entwickler können Fehler schnell erkannt und lokalisiert werden.

1.1 Spät Änderungen integrieren

Ein spät erkannter Fehler führt zu einer schwereren Eingrenzung des Codebereiches. Durch einen regelmäßigen Beitrag zum Code und den damit verbunden automatischen Überprüfungen sind Fehler direkt mit zeitlich kleineren Arbeitseinheiten verknüpft. Dadurch können sie schneller gefunden werden.

> "Continuous Integration doesn't get rid of bugs, but it does make them dramatically easier to find and remove." - Martin Fowler, Chief Scientist, ThoughtWorks [1]

1.2 Kosten

Eine ständige Entwicklung mit verbundener Integration und Testphase ist teuer. Denken Sie hier langfristiger. Der angesprochene Zeitfaktor zum Finden ist kritisch. Früh erkannte Fehler sind einfach zu beheben und verbrauchen somit weniger Ressourcen. Wird nicht nach dem Entwicklungsverfahren gelebt entstehen längere Zeitperioden in dem eine Integration statt findet. Eine solche Zeitspanne bedeutet eine exponentielle Steigerung der nötigen Ressourcen zum Finden und Beheben der Fehler. Sie fahren demnach deutlich besser, wenn Fehler möglichst früh erkannt und behoben werden.

1.3 Projektstabilität

Spät erkannte Fehler rauben Ressourcen, welche an anderer Stelle wichtig sind. Werden Mitarbeiter von der Entwicklung zur Fehlerbehebung geschoben ist der Zeitplan in Gefahr. Im schlimmsten Fall können Kritische Fehler das komplette Projekt zum Fall bringen. Mit CI sollten diese Gefahren in akzeptablen Größen auftauchen. Fehlerbehebungen sind gut zu planen und bringen nicht überraschend das Projekt in Gefahr.

1.4 Vorteile

Jedes Entwicklungsunternehmen profitiert von vielen Vorteilen gegenüber einer späten Integration dank CI.

- Die langen intensiven Integrationen entfallen

- Ein häufiger Beitrag erhöht die Sichtbarkeit der Codeänderungen für andere Mitarbeiter und fördert somit die Kommunikation
- Probleme werden schnell erfasst und behoben
- Es kann mehr Zeit in die eigentliche Entwicklung anstatt in die Fehlerbehebung investiert werden
- Aufbauende Änderungen basieren auf einem festen Fundament
- Direktes Feedback über Funktionsfähigkeit des neuen Codes
- Eine Reduzierung von Integrationsproblemen erlaubt eine häufigere Auslieferung der Software

2 Anforderungen

Continuous Integration muss, wie jedes Entwicklungsverfahren, gelebt werden. Dazu sind einige Anforderungen an das System und Team zu stellen.

2.1 System

Es gibt ein gemeinsames Quellverzeichnis in das alle Mitarbeiter ihre Änderungen hinterlegen. Dieses wird regelmäßig gebaut und mit hinterlegten Tests selbständig überprüft. Jede neue Änderung muss auf einem Integrationssystem einen erfolgreichen Schritt des Kompilers mit sich ziehen. Jeder Kompiliervorgang sollte kurz gehalten werden um ein direktes Feedback über die Qualität der Änderung zu erhalten. Es ist immer ratsam die Tests in einem direkten Klon der späteren Produktionsumgebung durchzuführen. Dies verhindert spätere Integrationsprobleme. Die getestete Software sollte für alle zur Verfügung stehen um einen Austausch über den aktuellen Stand zu gewährleisten.

Einen Schritt weiter sollte das Continuous Deployment angesetzt werden um es direkt ausliefern zu können.

2.1.1 Umsetzung

Soll CI intern genutzt werden sind einige Schritte nötig um die gewünschten Ergebnisse zu erzielen.

Es wird ein globales Quellverzeichnis angelegte, welches jeder Entwickler nutzen wird. Dazu dienen Git Dienst wie zum Beispiel Github oder Bitbucket. Lokale Änderungen werden getestet und nach Erfolg in das Verzeichnis übertragen. Der CI Server hört auf Änderungen in dem Quellverzeichnis und holt sich den aktuellen Stand. Dieser wird gebaut und mit Unit- und Integrationstests überprüft.

Nach Erfolg wird es dem Team mit einem Namen der Version zur Verfügung gestellt. Eine Benachrichtigung über den Zustand wird nachfolgend an das Team gesendet.

Ist jedoch ein Fehler während des Bauvorgangs aufgetreten wird das Entwicklerteam alarmiert diesen schnellst möglich zu beheben.

Dieser Prozess wird durch das komplette Projekt zyklisch gezogen.

2.2 Team

Auf menschlicher Ebene müssen natürlich auch einige Regeln eingehalten werden. Essentiell ist der regelmäßige Änderungsbeitrag jedes Entwicklers in das Quellverzeichnis. Änderungen mit Fehlern im Code gehören nicht den gemeinsamen Pool der Software, sondern sollten vorher lokal syntaktisch korrigiert werden. Code muss lokal immer lauffähig sein bevor darüber nachgedacht wird die Änderung zu veröffentlichen. In vielen Unternehmen wird deshalb die Arbeitszeit nach hinten gezogen bis dies geschehen ist.

Viele Teams entwickeln aus diesen Regeln ein tägliches Ritual. Dadurch korrigieren und kontrollieren sie sich selbst. Es ist nicht nötig weitere Regeln aus höherer Hierarchieebene geben zu müssen.

3 Software

Unterstützen gibt es natürlich für alle Prozessschritte bereits Software. Nachfolgende gehen wir auf drei Softwaregestützte Aspekte ein.

3.1 Templates

Jedes Projekt muss irgendwo anfangen

Literaturverzeichnis

[1] ThoughtWorks, Inc.: *Continuous integration eliminate blind spots so you can build and deliver software more rapidly.*, November 2016. https://www.thoughtworks.com/de/continuous-integration.