

Gemischte Teams

Entwickler checken Code zeitnah und in kleinen, sinnvollen Changesets ins Repository ein.

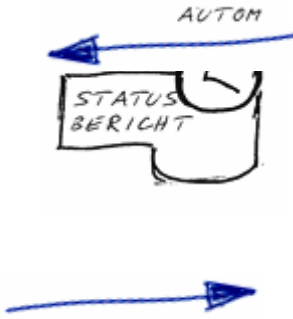
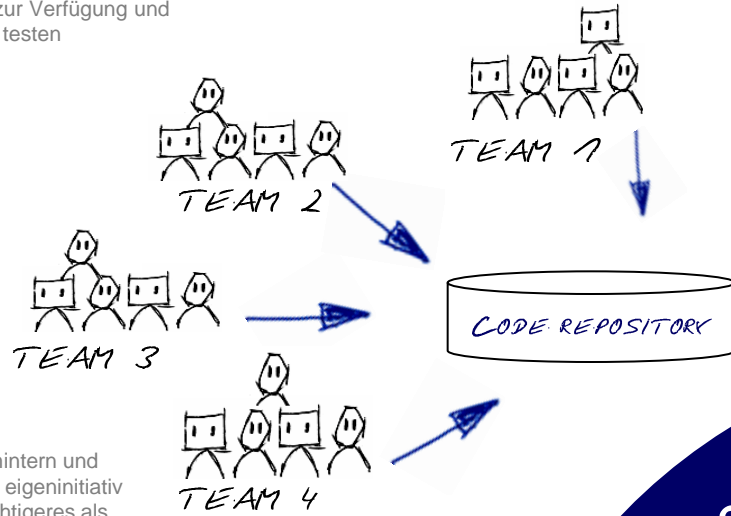
Tester und Entwickler haben einfachen Zugriff auf einen guten Softwarestand. Sie haben diesen ständig lokal zur Verfügung und können so unmittelbar dagegen testen und entwickeln.

Alle Projektbeteiligten haben Verantwortung für eine gute Qualität und bekommen auch die (technischen) Voraussetzungen um sie wahrzunehmen.

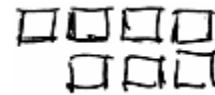
Vor dem Einchecken werden betroffene Komponenten lokal getestet.

Gefundene Fehler werden teamintern und übergreifend sofort verteilt bzw. eigeninitiativ angegangen, es gibt nichts Wichtigeres als funktionierende Software.

Alle Projektbeteiligten können zusätzlich jederzeit eine beliebige Auswahl automatisierter Akzeptanztests lokal und gegen die Testumgebung(en) ausführen.



GATE 1



UEBERWACHT
MODULWEISE
UND ZEITNAH

GATE 2

FUEHRT UNIT
UND DB TESTS
AUS

Build Server

Verschiedene Arten von „Build“-Tests liefern schnelles Feedback für den Entwickler.

Top-Down-Builds bauen alle zusammengehörenden Module und hinterlassen einen „SCM Changelog“ sowie einen Tag / Label im Repository zwecks Reproduzierbarkeit, sowie ggf. die Release Notes.

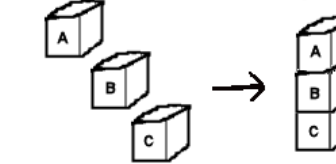
Das Autodeployment erfolgt auf Basis der zusammenpassenden Artefakte aus den Top-Down-Builds.

Die Regressionstests können ebenfalls versioniert und getagged werden, idealerweise gemeinsam mit der Software.

Diese Tests werden zeit- oder ereignisgesteuert ausgeführt. Die Testberichte können ggf. zentral historisiert werden.

Beim Aufbau der Testumgebung sind Informationen über die Produktionsumgebung mit eingeflossen.

GATE 3



FUEHRT REGELMAESSIG
„TOP DOWN BUILDS“ AUS

AUTO DEPLOY
BINARIES
UND MOCKS

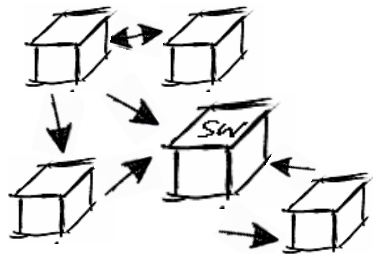


Lieferbare Software

Weil immer der „last known good build“ released wird, ist für den Regelprozess kein Code-Freeze nötig. Kurze Roundtrips gewährleisten schnelle Fehlerbehebung.

Die Endanwender bzw. Anforderer haben Einblick in den real existierenden Softwarestand und können ihre Wünsche fortlaufend einbringen. Art und Umfang aller durchgeführten Tests fließen mit in die kontinuierliche Abnahme ein.

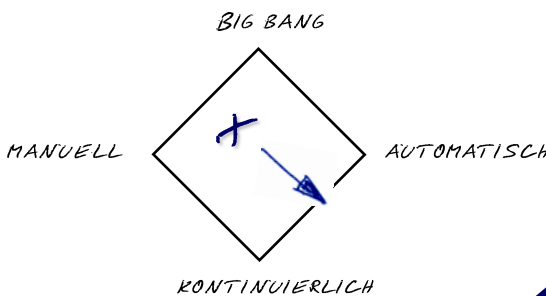
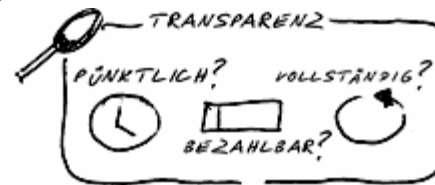
Der Power-On-Self-Test kann mit zur Verifikation der korrekten Installation und Vernetzung der Software-Landschaft in der Produktivumgebung dienen.



Die Tests können gemeinsam mit der Software gebranchet und gemerged werden, so dass auch der Release-Branch ständig überwacht wird.

Die Tests bzw. Bausteine daraus können teilweise für die Überwachung der Funktionalität in Produktion wieder verwendet werden (Monitoring).

Kontinuierliche Integration



Kontinuierliche Verbesserungen

Testumgebung

Die Testumgebung ist komplett deterministisch: Es kann ständig ein definierter Ausgangszustand erzeugt werden, auf dem die Tests aufsetzen können.

Mocks simulieren dabei abhängige Schnittstellen von Drittsystemen für den jeweiligen Anwendungs- bzw. Testfall.

Ein Power-On-Self-Test stellt die korrekte Installation der Software sicher: Externe Schnittstellen, Datenbanken, Dateisystem etc.

Testberichte gehen automatisch an alle Projektmitarbeiter um daraus Aktivitäten abzuleiten.

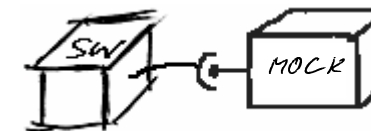
Die Testreports werden durch QA-Verantwortliche bewertet. Neben dem persönlichen Gespräch mit anderen Projektmitgliedern werden hier auch die Changelogs und Release-Notes aus GATE 3 verwendet.

Auch Last- und Performance-Tests werden regelmäßig durchgeführt. Sie können und sollten auf die funktionalen Tests aufsetzen.

GATE 4

AUSFUEHREN
POWER ON
SELF TEST

GATE 5



AUTOMATISCHE
AKZEPTANZTESTS

GATE 7

LAST UND
PERFORMANCE
TESTS
AUSWERTEN

GATE 6

REPORTS
AUSWERTEN