



„10 things I wished they'd told me!“

aktuell. innovativ. praxisbezogen.





Reinhard Salomon
SEQIS Geschäftsleitung



SEQIS „10 things“ – Programm 2015



- 19.03.2015 Mobile Testing:
APPlaus, APPlaus, Qualität für unterwegs
- 18.06.2015 Agile Steuerung:
Trotz Vernunft bei der Auslastung zum Erfolg
- 17.09.2015 Pair Analyse:
Qualität von Anfang an
- 19.11.2015 Continuous Integration:
Mehr als nur ein Schlagwort**



Continuous Integration

Mehr als nur ein Schlagwort

Markus Schwabeneder

SEQIS Consultant



Ziele von Continuous Integration

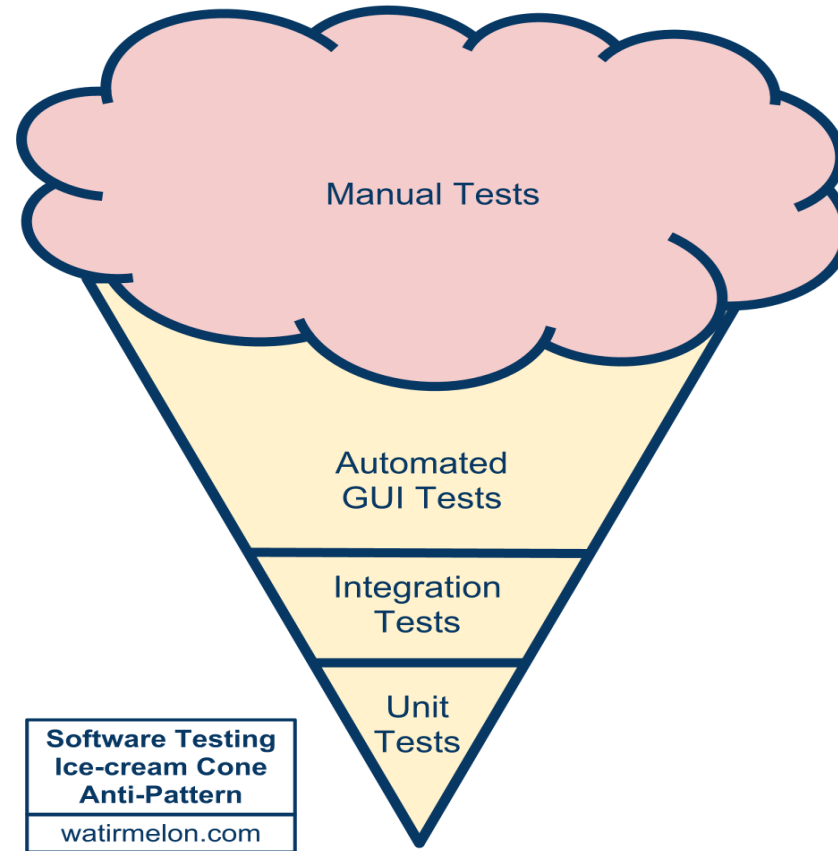


- Integration und Qualitätssicherung des Codes
- Fast Feedback über den Zustand des Codes und des Builds

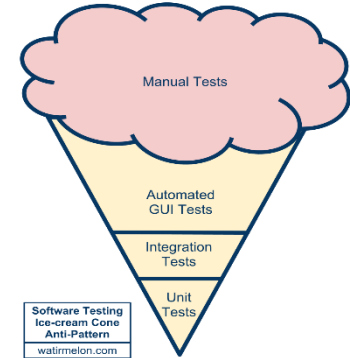
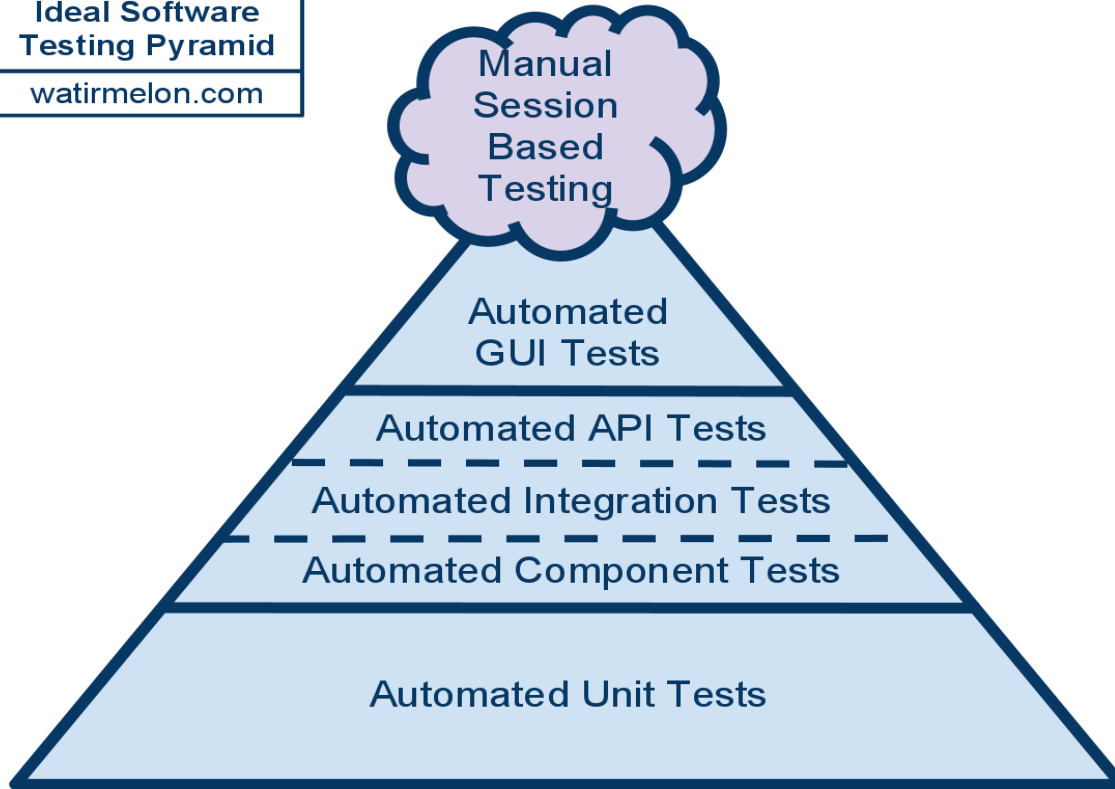


1. Erweitern Sie Ihre Lösung kontinuierlich

- Cost of Delay
- schnelles Feedback



Ideal Software Testing Pyramid
watirmelon.com





2. Unit-Tests, aber richtig!

- Frühe Fehlererkennung -> geringe Kosten
- Unit-Tests ohne kompliziertes Mocking
- 100% Abdeckung der neuralgischen Stellen, nicht des Codes



3. Nützen Sie Style Checks und automatische Code Reviews

- Einheitlicher Style erhöht die Lesbarkeit
- Bessere Wartbarkeit durch schlankere Codebasis



"Works on my machine"?!

This is Phyllis and she doesn't care if the build works on your machine. As you can see, she's a busy woman with a jam-packed social calendar. She doesn't have time for brittle builds from the likes of you; she needs a build that just plain works. You hear?!

A photograph of an elderly woman with white hair and glasses, wearing a white chef's hat and a white apron. She is holding a black frying pan in her right hand, a bottle of wine in her left hand, and a lit cigarette in her mouth. She has a serious expression.

If you start using continuous integration, Phyllis can get back to gettin' her groove-thing on.
www.BuildsOnMyMachine.com



4. Automatisieren Sie den Buildvorgang

- Nachvollziehbar und reproduzierbar
- Arbeitersparnis
- Jeder hat den gleichen Build



Quelle:
<http://www.flickr.com/photos/hanuska/5931613961/in/photostream/> licensed under
CC BY-SA 2.0 via Wikimedia Commons -
[https://commons.wikimedia.org/wiki/
File:Series_of_build_lights.jpg](https://commons.wikimedia.org/wiki/File:Series_of_build_lights.jpg)

SEQIS Plug-In für Jenkins

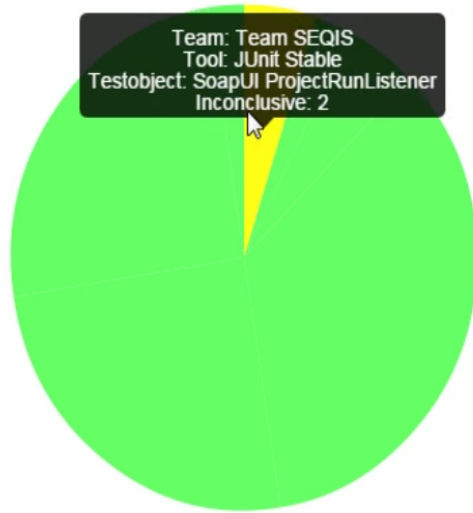


Übersicht über durchgeführte Tests und deren Status.

© SEQIS Software Testing GmbH



Testobject Version: trunk



Team/Tool/Testobject Filter

Apply Filter

Reload Data

Team, Tool and Testobject	Last runs	Date of last run	Result of last run
Team: Blue TestCase Tool: JUnit Stable Testobject: Stocks TestCase		23.02.2015	1/1
Team: Blue TestSuite Tool: JUnit Stable Testobject: Stocks TestSuite		23.02.2015	2/2
Team: Team on Package lvl Tool: JUnit Stable Testobject: Testobject on Package lvl		01.03.2015	1/1

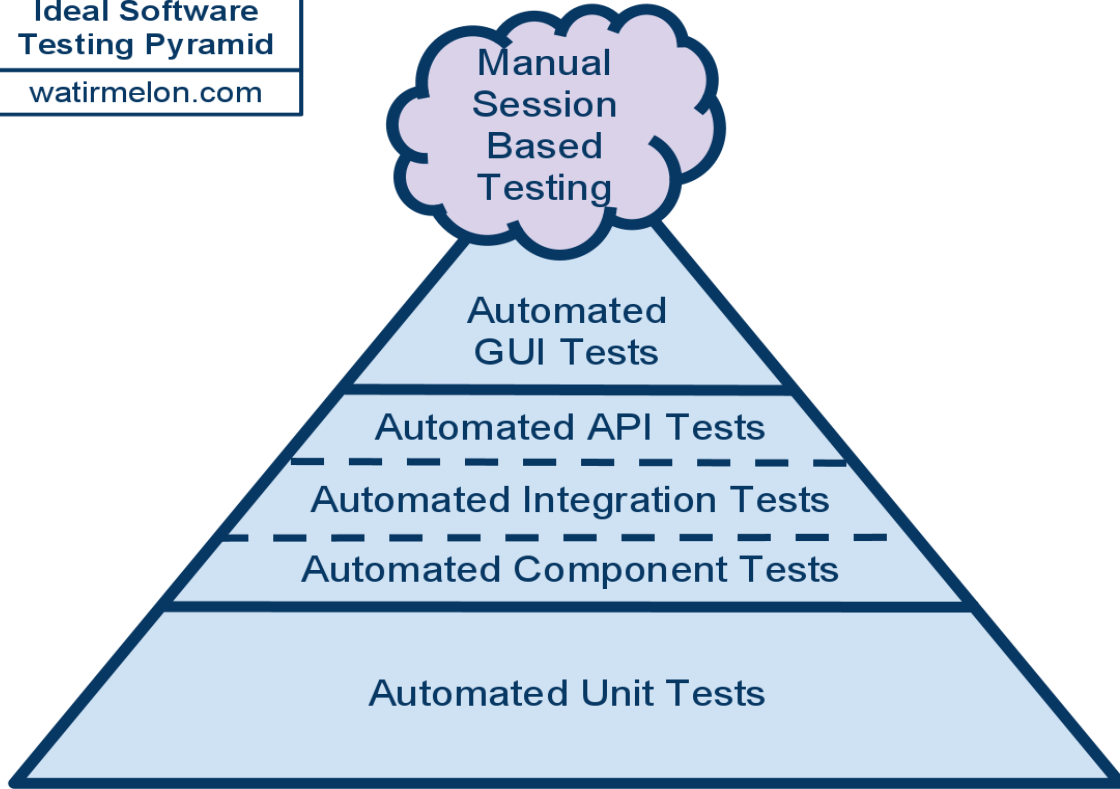
Aufteilung nach Teams, Testtools, Testobjekten, Versionen, etc.



5. Zeigen Sie den Status des Builds

- Fast Feedback
- Problem erkannt, Problem gebannt

Ideal Software
Testing Pyramid
watirmelon.com





6. Downstream Builds für länger laufende automatisierte Tests

- „Nightly Build“
- Weitere Umsetzung der Testpyramide
- Faster Feedback durch Parallelisierung

YOU BROKE THE BUILD!

This is Agnes and she's none too pleased with you, Bub. You broke the build. You should have known that you were making breaking changes, but you checked them in anyway.



If you start using continuous integration, Agnes won't have to come back.

www.YouBrokeTheBuild.com





7. Entwickler ist erste Qualitätssicherung

- **Moderne IDEs unterstützen**
 - Unit-Tests
 - Style Check
 - Code Review
- **Buildprozess wie am CI System**
- **Broken Builds beeinflussen das ganze Team**



8. Schwierigkeiten werden gemeinsam gemeistert

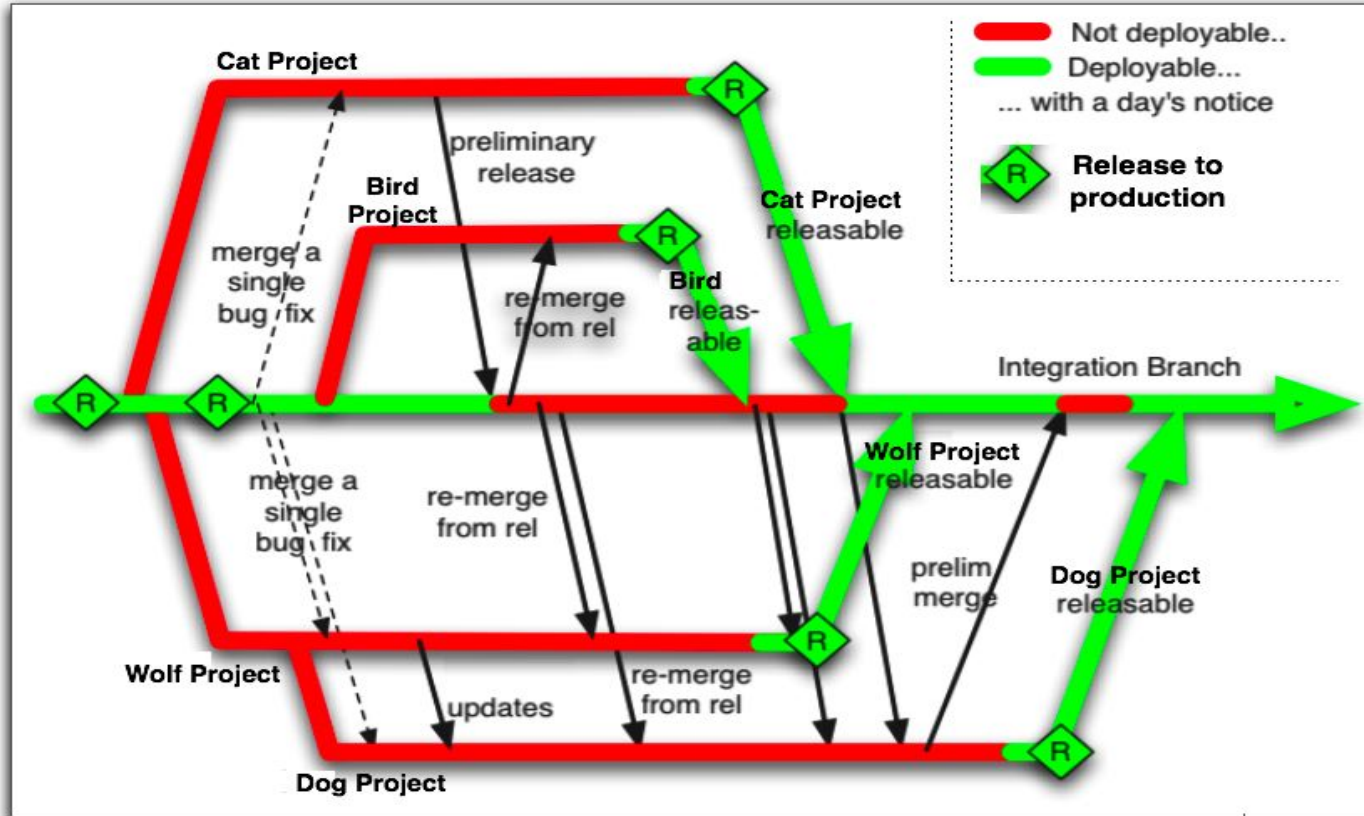


Bild aus http://paulhammant.com/blog/branch_by_abstraction.html

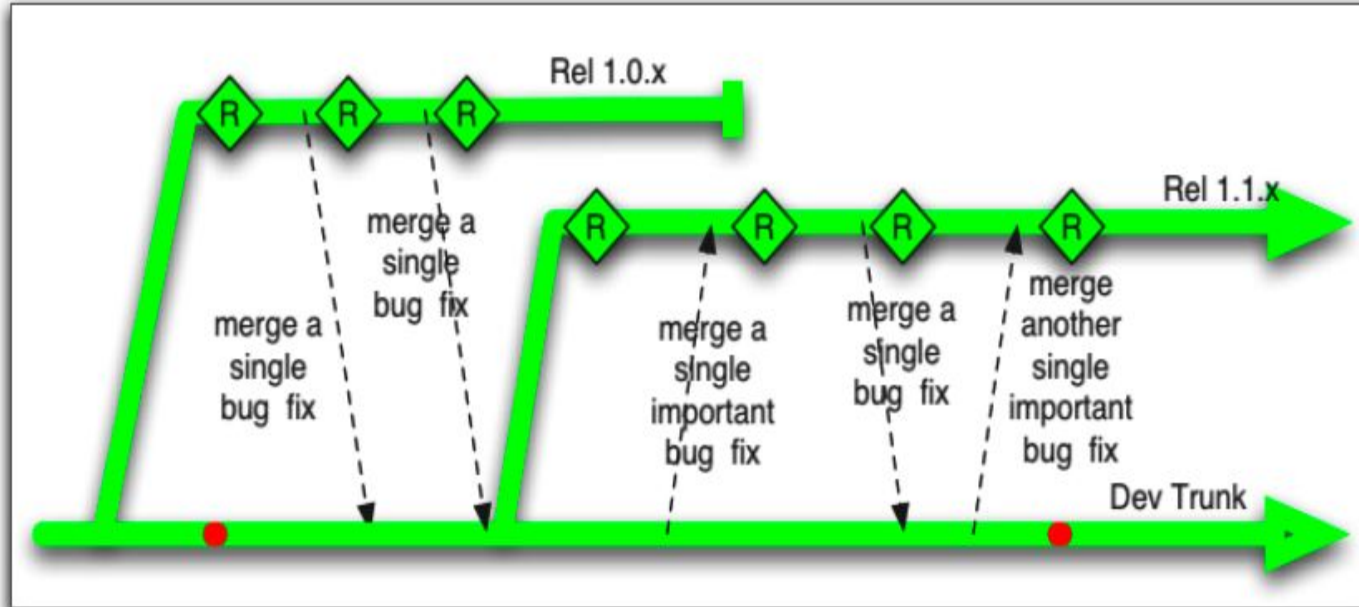


Bild aus http://paulhammant.com/blog/branch_by_abstraction.html



9. Arbeiten Sie an einer gemeinsamen Codebasis („Trunk“)

- „Branching“ nur wenn wirklich notwendig
- „Feature Toggles“ können helfen



10. Vermeiden Sie „Integration Hell“: Sofort einchecken

- Sobald Änderungen abgeschlossen und lokale Tests erfolgreich
- Am Ende des Arbeitstages alle erfolgreich getesteten Codeteile
- kleine Codeeinheiten garantieren einfaches Zusammenführen



Zusammenfassung

1. Erweitern Sie Ihre Lösung kontinuierlich
2. Unit-Tests, aber richtig
3. Nützen Sie Stylechecks und automatisierte Code Reviews
4. Automatisieren Sie den Buildvorgang
5. Zeigen Sie den Status des Builds
6. Downstream Builds für intensive automatisierte Tests
7. Entwickler ist erste Qualitätssicherung
8. Schwierigkeiten werden gemeinsam gemeistert
9. Arbeiten Sie an einer gemeinsamen Codebasis („Trunk“)
10. Vermeiden Sie „Integration Hell“: Sofort einchecken



Continuous Integration

Mehr als nur ein Schlagwort

Markus Schwabeneder

SEQIS Consultant



SEQIS „10 things“ – Fortsetzung 2016



- 4 Termine (2x Frühjahr, 2x Herbst)
- Themenvorschläge in Kürze unter www.SEQIS.com/unternehmen/events

Ihre Meinung ist uns wichtig!