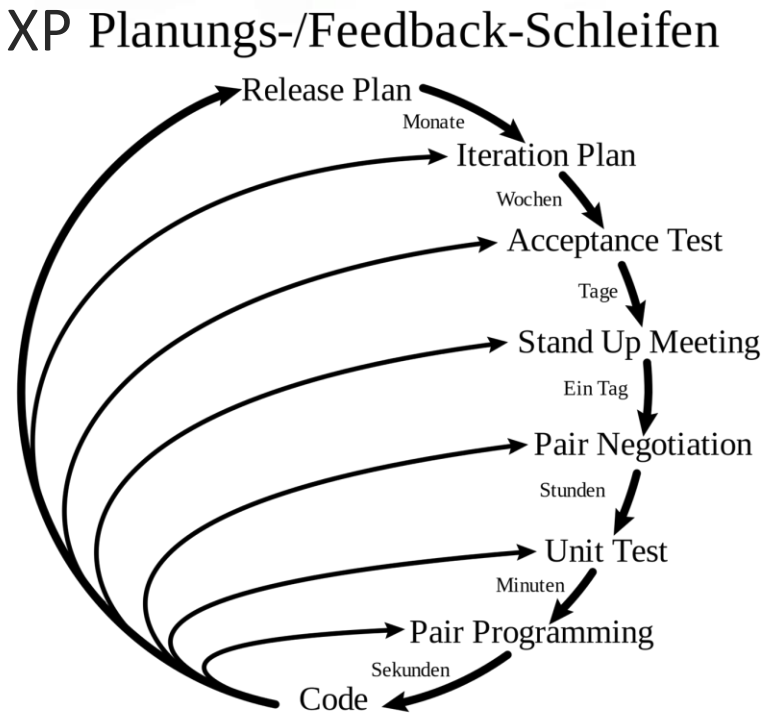
Emergente Testarchitektur

Moritz Tiedje

moritz.tiedje@andrena.de

Warum schreiben wir Tests?



Warum?
=> Feedback

© 2021 andrena objects ag

Warum schreiben wir Tests?

- Freiheit von hohem manuellen Testaufwand
- Lebendige Dokumentation
- Leichte Wartbarkeit
- Anforderungsdefinition (TDD)
- Grundbaustein im Entwicklerworkflow:
  - Write Failing Test -> Make it pass -> Refactor (TDD)
  - Get it working –> Test it –> Refactor

© 2021 andrena objects ag

Wie erkenne ich einen guten Test?

Ein guter Test gibt **schnelles**, **wertvolles** und **kosteneffizientes** Feedback zum Verhalten des Programms.

Schnell:

- Die Ausführungszeit behindert nicht meinen Entwicklerworkflow

Wertvoll:

- Keine False Positives / Negatives
- Es ist leicht nachvollziehbar, welches Szenario getestet wird
- Wenn der Test fehlschlägt ist leicht erkennbar, wie das Problem angegangen werden kann.

Kosteneffizient:

- Der Aufwand den Test zu schreiben ist verhältnismäßig zur Entwicklung
- Der Aufwand den Test zu warten ist angemessen
- Die benötigte Infrastruktur ist angemessen

© 2021 andrena objects ag

Arten von Tests…

- Funktionales Verhalten
  - Unit Test, Integrationstest, Akzeptanztest, UI-Test, Screenshot Test, End-to-End Test, Contract Test…
- Interne Qualität
  - Code Smells, Linting, Test Coverage, Mutation Coverage…
- Performance
- Security
  - Vulnerability Scan, Pen-Test…

© 2021 andrena objects ag

Arten von Tests - Architektur

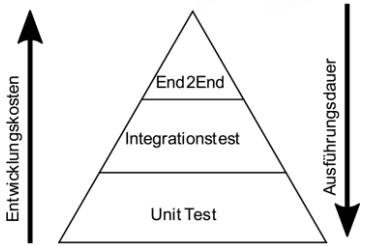
- Automatisierte Tests gibt es in vielen Arten.
- Für jede mögliche Art gibt es viele mögliche Technologien & Frameworks & 3rd-party-libraries.

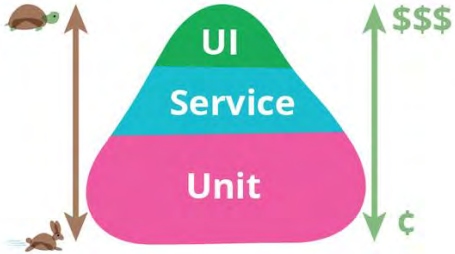
=> Wir müssen Entscheidungen treffen.

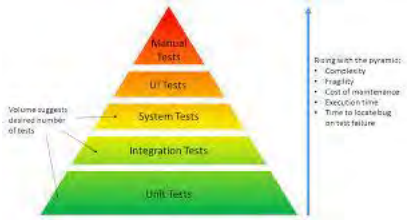
- Diese Entscheidungen sind nicht in Stein gemeißelt, aber
- Oft kostenaufwändig zu ändern / zu korrigieren
- Werden berücksichtigt wenn weitere Entscheidungen anstehen

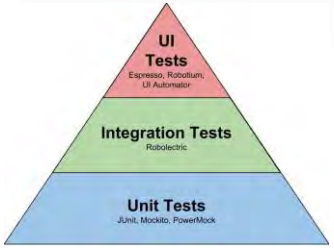
© 2021 andrena objects ag

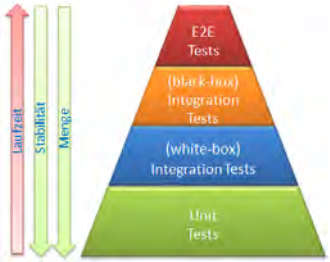
Die ideale Testarchitektur

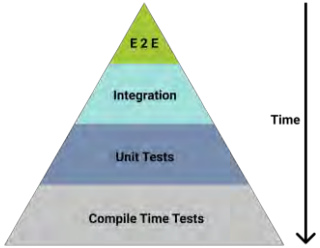


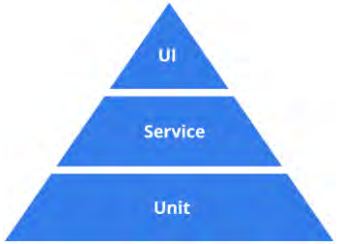


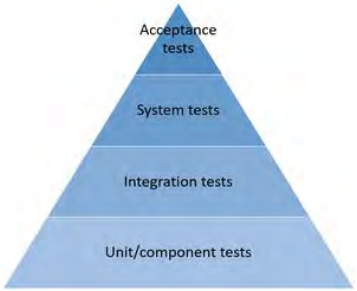






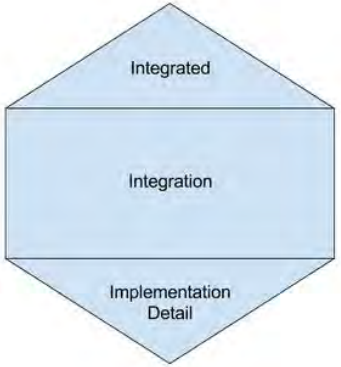






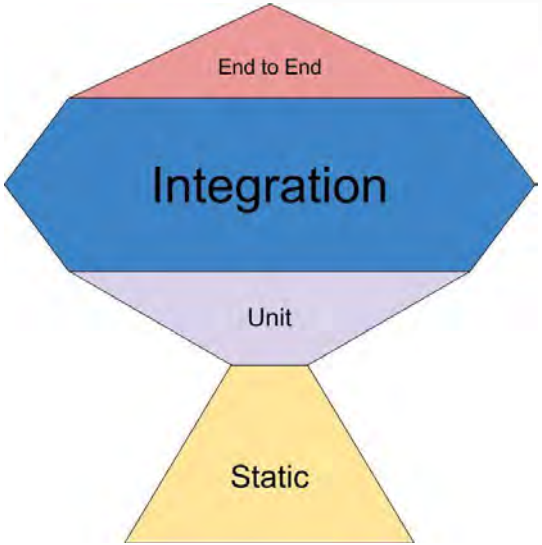
© 2021 andrena objects ag

Die ideale Testarchitektur



Testing Honeycomb for Microservices - Spotify

Testing Honeycomb



Test Trophy

© 2021 andrena objects ag

Die ideale Testarchitektur

Wenn ich etwas baue, dass wie ein Flugzeug aussieht, dann wird es nicht automatisch fliegen können.

Wenn ich etwas baue, dass fliegen kann, dann *könnte* es wie ein Flugzeug aussehen.

© 2021 andrena objects ag

Die ideale Testarchitektur

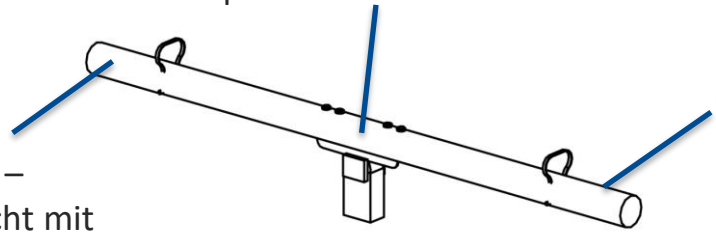
Wenn ich eine Suite baue, die wie eine Testpyramide aussieht, dann wird sie nicht automatisch schnelles, wertvolles und kosteneffizientes Feedback geben.

Wenn ich etwas baue, dass schnelles, wertvolles und kosteneffizientes Feedback gibt, dann *könnte* es wie eine Testpyramide aussehen.

© 2021 andrena objects ag

Emergente Testarchitektur – Die Balance

Emergente Testarchitektur – Vorhandene Vision mit Inspect & Adapt. Die Möglichkeiten und auch die gängigen Architekturen sind bekannt. Experimente finden statt.



Accidental Architecture – Man beschäftigt sich nicht mit dem Thema Testarchitektur. Leicht sichtbare Probleme werden über die Jahre immer schlimmer. Es fehlt die Konsequenz oder das Know-How für Inspect & Adapt.

Architekturplan Cargo Cult – Kaum Inspect & Adapt Möglichkeiten ohne viel Diskussion und Konflikt. Es gibt keine Experimente. Der Plan wird eingehalten, die Tests könnten aber schneller, wertvoller und kosteneffizienter sein.

© 2021 andrena objects ag

Emergente Testarchitektur - Inspect

- **Schnell**
  - **Repeatability** – Wie oft führen wir diese Suite aus?
  - **Speed** – Wie lange warten wir auf Feedback?
- **Wertvoll**
  - **Stability** – Laufen die Tests zuverlässig?
  - **Coverage** – Was decken wir mit dieser Suite ab?
  - **Useful Failures** – Schlagen die Tests aus hilfreichen Gründen fehl?
  - **Debugability** – Wie leicht fixen wir Fehler wenn ein Test failt?
  - **Readability** – Wie leicht kann das Verhalten des Programms im Test nachvollzogen werden?
- **Kosteneffizient**
  - **Development** – Wieviel Implementieraufwand wird uns diese Testschicht kosten?
  - **Cognitive Load** – Wieviel komplizierter wird das Produkt mit dieser Testschicht?
  - **Infrastructure** – Welche Infrastruktur wird benötigt & wie teuer ist die?
  - **Maintenance** – Wie oft & wie teuer müssen wir die Testsuite warten?

© 2021 andrena objects ag

Emergente Testarchitektur - Inspect

Myitems - Testschichten	Mehrwert	Kosten	Debugability	Cognitive Load	Stability	Useful Failure
Unit Tests	😍		😍😍😍		😍	😍
Integration Test	😍		😍😍	☹	😍	😍
Contract Tests	😍😍		😍😍😍	☹	😍	
End-to-End Tests	😍	☹	☹☹☹	☹	☹☹	☹☹☹
Performance Tests	😍😍😍	☹☹		☹☹☹	☹	😍

© 2021 andrena objects ag

Emergente Testarchitektur

Wieso?

- Schnelles, wertvolles & kosteneffizientes Feedback

Welche Schichten?

- It depends

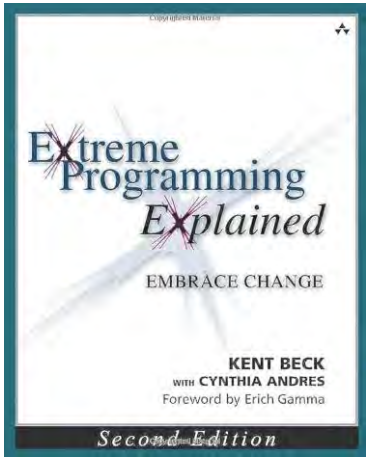
Wie?

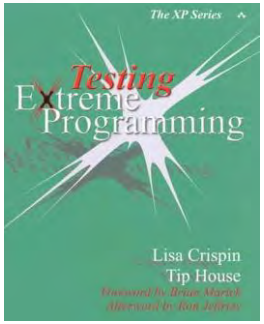
- Bewusst Zeit für Inspect nehmen. Visualisierung hilft.
-

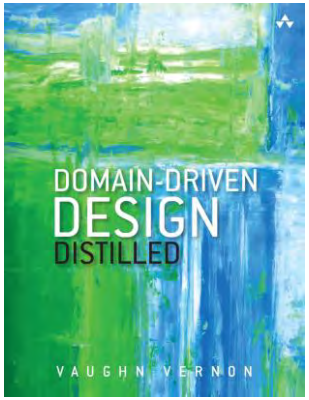
© 2021 andrena objects ag

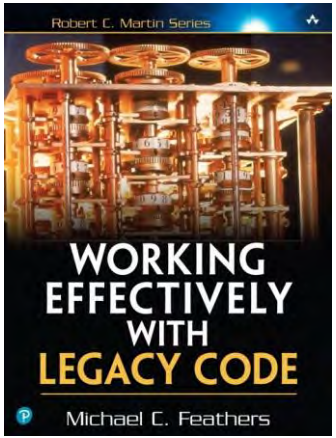
Emergente Testarchitektur

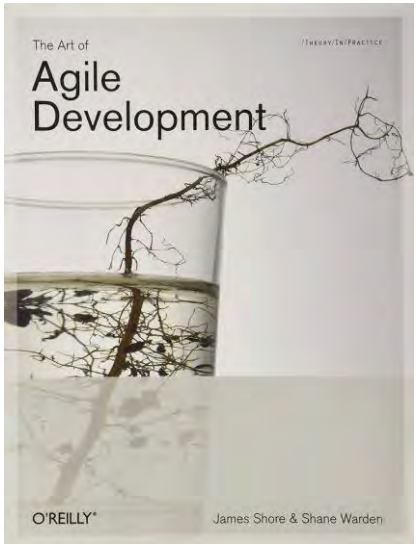
Wie?











© 2021 andrena objects ag