

INFOGEM AG

Informatiker Gemeinschaft für Unternehmensberatung

*Rütistrasse 9, Postfach
5401 Baden, Switzerland
Phone: +41 56 222 65 32
Internet: www.infogem.ch*

Tonleiter oder Akkord: Wie spielt die Musik im Test

Thomas Haas
Senior Consultant

Tonleiter – Akkord

- Eine **Tonleiter** oder Skala ist in der Musik eine definierte auf- oder absteigende Folge von Tönen, die in einem musikalischen Zusammenhang stehen.



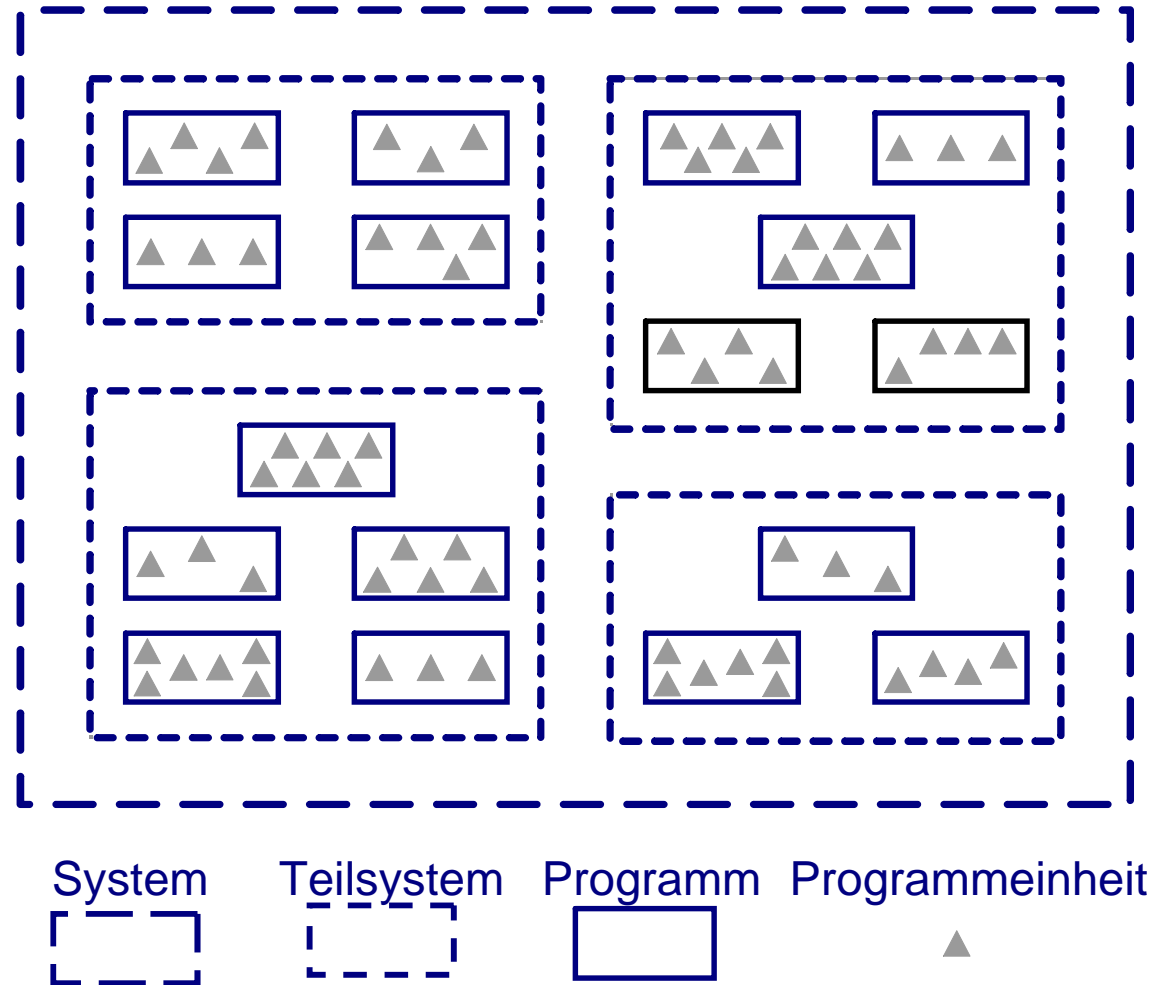
- Ein **Akkord** ist in der Musik das gleichzeitige Erklingen mehrerer unterschiedlicher Töne, die sich harmonisch deuten lassen.



Quelle: Wikipedia: <http://de.wikipedia.org/wiki/Tonleiter>, <http://de.wikipedia.org/wiki/Akkord>

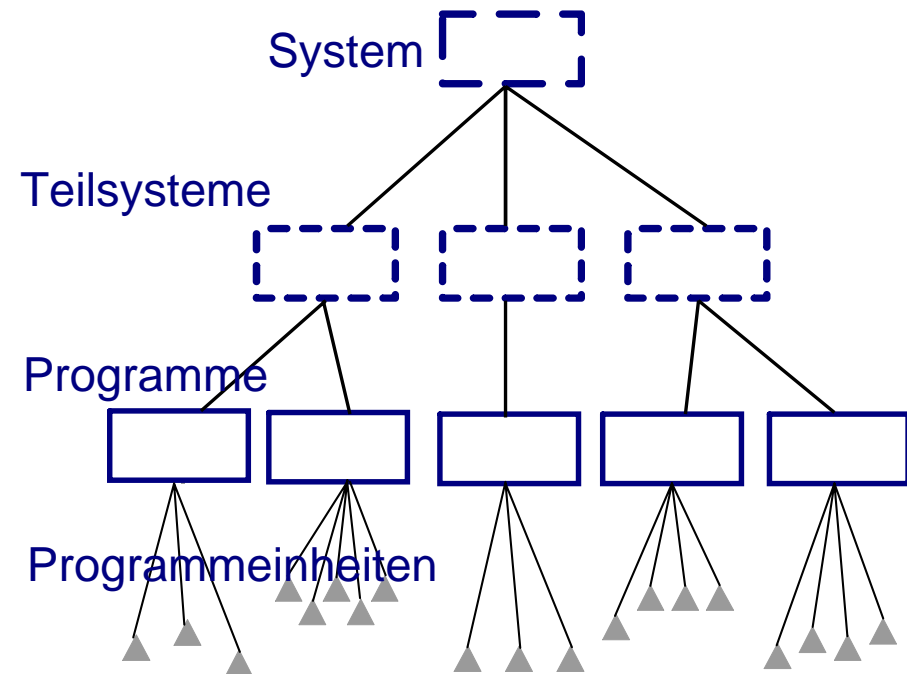
Systemaufbau als Baumstruktur

- Ein (Gesamt-) System besteht typischerweise aus einzelnen Komponenten.
- Die einzelnen Komponenten bilden eine Baumstruktur: System, Teilsysteme, Programme (und Bibliotheken), Programmeinheiten.



Strategien für den Gesamt-Test (1/2)

- Gemäss dem System-Aufbau:
 - Bottom-Up: Beginnend mit den Programmeinheiten über die Programme, Teilsysteme zum Gesamtsystem – benötigt Treiber.
 - Top-Down: Beginnend mit dem System über die Teilsysteme zu den Programmeinheiten -- benötigt Stubs (Stellvertreter)
- Entlang des (Haupt-) Ausführungsstrangs
- In Stücken:
 - Vollständig inkrementell: Stück um Stück zusammenbauen und testen.
 - Am Stück: Zusammenbauen und testen.
 - Teilweise inkrementell: eine (gute) Mischung.



Strategien für den Gesamt-Test (2/2)

- Die Wahl der Strategie zu Integrationstests wird durch verschiedene Rahmenbedingungen zusätzlich beeinflusst:
 - Verantwortung über die Komponenten bei unterschiedlichen Organisationen: Lieferanten, Unterlieferanten, Integrator
 - Unterschiedliche Risiken in den einzelnen Komponenten: Neuentwicklung, Anpassung, Integration von neu mit alt, eingesetzte Technologien, Reife des Lieferanten
 - Verfügbarkeit von Ressourcen und Hilfsmitteln: Testumgebung, Test-Werkzeuge, Testdaten, Testfälle, Tester, Know-How
 - Zeitaspekt: Aufwand und Dauer der verschiedenen Tests.

Elemente des Gesamt-Tests

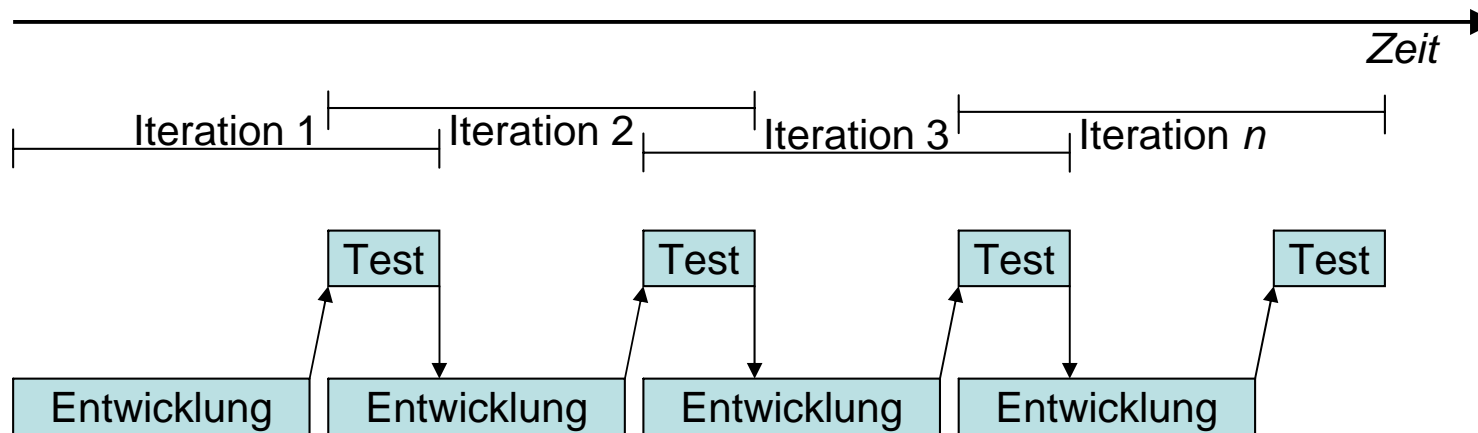
- Für die komplette Prüfung (der Software) werden in der Regel verschiedene Prüfmethode angewendet: Reviews, Inspektionen, **Tests**, etc.
- Die **Gesamtheit aller Tests** können weiter nach **Test-Stufen** unterteilt werden (Bottom-Up): Programmeinheiten, Programme und Bibliotheken, Teilsysteme, System, etc.
- In jeder **Test-Stufe** werden wiederum unterschiedliche **Test-Arten** eingesetzt: Whitebox, Blackbox, Positive, Negative, Stress-Tests, etc.
- Zu Jeder **Test-Art** schliesslich gib es einen oder mehrere **Test-Fälle**.

→ Gesamt-Test → Test-Stufen → Test-Arten → Test-Fälle

Teststufen im Gesamt-Test

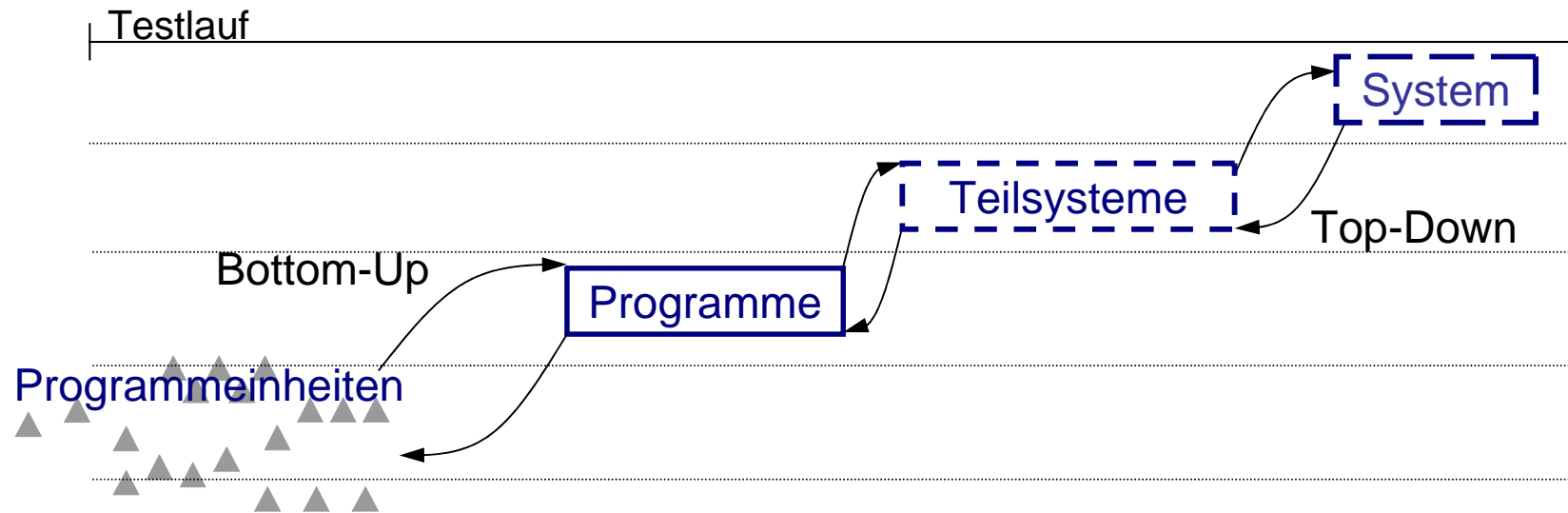
Teststufe	Zweck	Beschreibung
Programmeinheiten	Finden von Verarbeitungsfehlern; prüfen interner Schnittstellen.	Test einzelner, isolierter Programmeinheiten in geeigneter Umgebung.
Programme und Bibliotheken	Finden von Fehlern im Zusammenspiel der Programmeinheiten bezüglich den funktionalen und nicht funktionalen Anforderungen.	Testen des gesamten Programms in einer geeigneten Umgebung.
Teilsysteme, System	Finden von Fehlern in Teilsystemen respektive im Gesamtsystem bezüglich den Anforderungen – sowohl aus fachlicher als auch aus technischer Sicht.	Test des Systems respektive der Teilsysteme in einer Testumgebung.

Gesamt-Test und Entwicklungs-Iterationen (im Einklang)



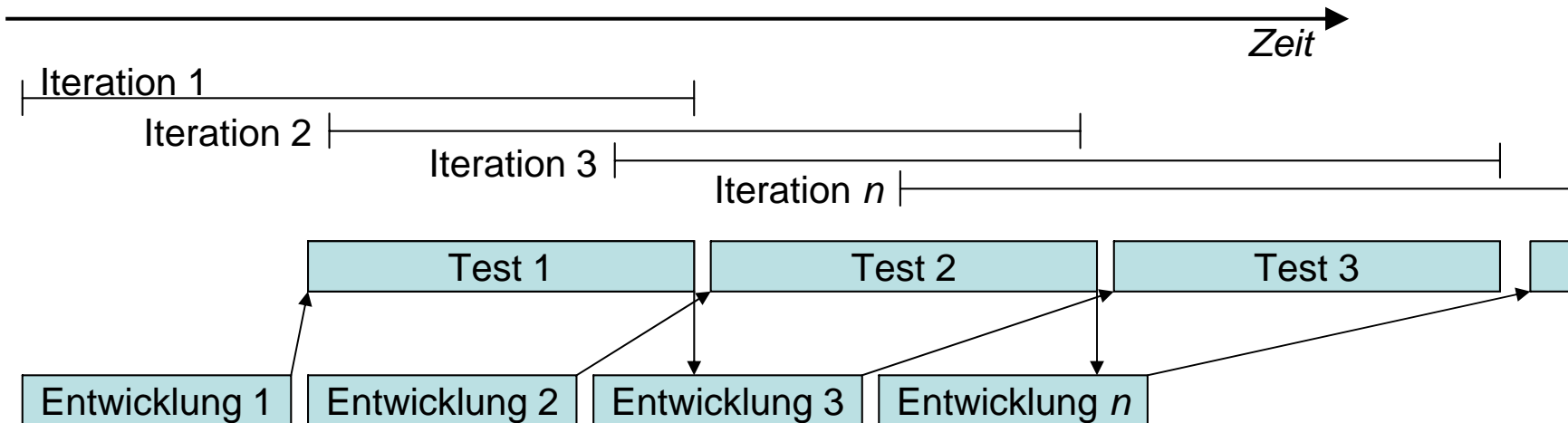
- Die Software-Entwicklung erfolgt in Iterationen. Jede Iteration schliesst mit einer Lieferung ab. Eine Lieferung beinhaltet neue Funktionalität und Fehlerkorrekturen.
- Jede Lieferung einer Iteration wird geprüft: Prüfung der neuen Funktionalität und Verifikation korrigierter Fehler.
- Damit Fehler einer Iteration in der nächsten korrigiert werden können, müssen die Tests rechtzeitig (Größenordnung Wochen) vor Ende der Entwicklung der nachfolgenden Iteration abgeschlossen werden.

Klassisch: Der Gesamt-Test als Tonleiter



- Test einzelner Komponenten über verschiedene Stufen von Programmeinheiten zum System (Bottom-Up) oder vom System zu Programmeinheiten (Top-Down).
- Nach erfolgreichem Test auf einer Stufe erfolgt der Wechsel auf die nächst höhere (Bottom-Up) oder tiefere (Top-Down) Stufe.
- Zeigt eine Test-Stufe zu viele Fehler auf, kann der laufende Gesamttest jederzeit abgebrochen werden.

Gesamt-Test und Entwicklungs-Iterationen (ausser Takt)

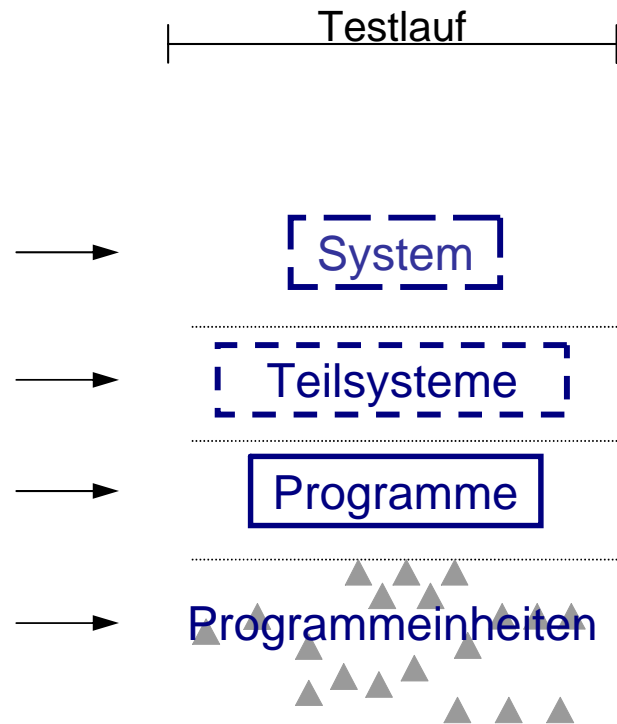


- Die Iterationen müssen in der Dauer aufeinander abgestimmt sein.
- Kurze Entwicklungs- und lange Testiteration führen zu spätem Feedback und zu Mehraufwendungen in der Entwicklung:
 - Die gefundenen Fehler müssen in der aktuellen Entwicklungsversion nachvollzogen werden.
 - Kritische Fehler können nicht auf den nächsten Test-Lauf korrigiert werden.

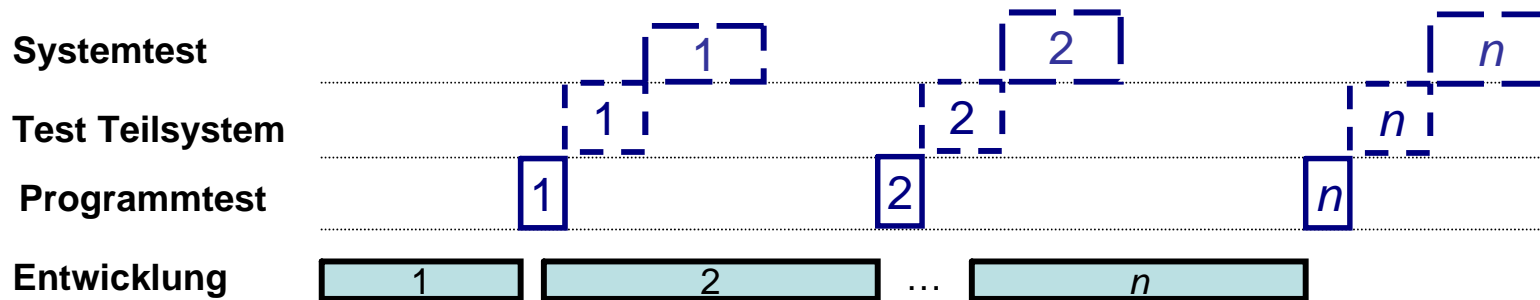
Massnahmen:

- Iteration in der Entwicklung verlängern – oder Pausen einlegen.
- Dauer der Tests verkürzen.

Gesamt-Test als Akkord?

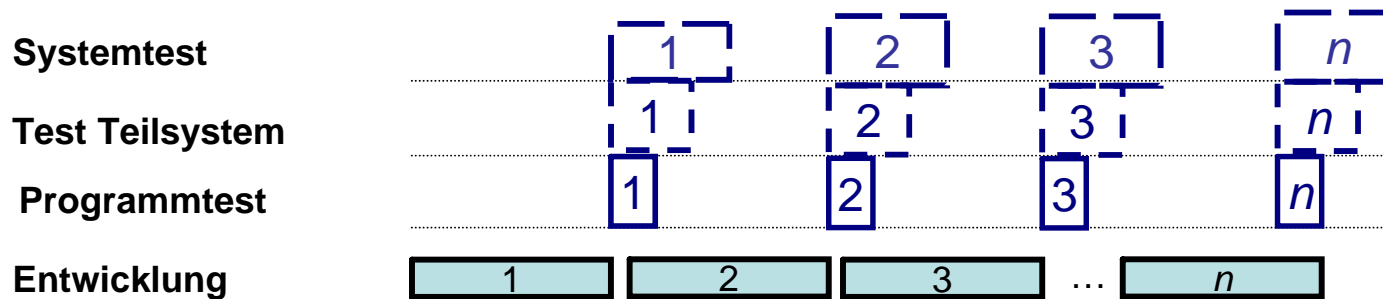


Eigenschaften der Tonleiter



- Schrittweise Testen einzelner Stufen (Bottom-Up oder Top-Down).
- Unnötige Test-Aufwendungen können durch Abbruch des Gesamt-Tests in jeder Stufe vermieden werden.
- Mit geringem organisatorischem und planerischem Aufwand realisierbar.
- Erste Wahl – entsprechende Rahmenbedingungen vorausgesetzt.

Eigenschaften des Akkords



- Sämtliche Teststufen werden gleichzeitig getestet.
- Der Gesamt-Test kann in kurzer Zeit durchgeführt werden.
- Erkenntnisse aus einer Teststufe können nicht in die Tests der nächsten Stufe einfließen (auslassen/forcieren von Tests bestimmter Aspekte auf Grund von Test-Resultaten).
- Hoher Bedarf an Test-Hilfsmittel und -Ressourcen bedingt durch paralleles Arbeiten.
- Die zu suchenden Fehlern in den einzelnen Stufen müssen gut aufeinander abgestimmt sein.

Kanon

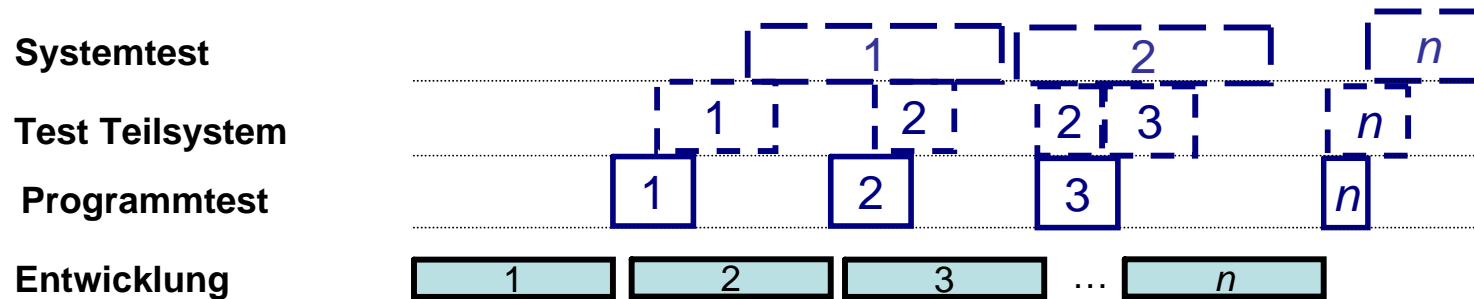
- Unter **Kanon** versteht man in der mehrstimmigen Musik kontrapunktische Kompositionen, in denen sich eine oder mehrere gleichzeitig erklingende Stimmen streng imitieren. In den meisten Fällen leiten sich aus einer notierten Stimme eine oder mehrere simultan erklingende Stimmen ab, welche nacheinander einsetzen. Diese Art des Kanons ist eng mit der sich durch freiere Imitation auszeichnenden Fuge verwandt und kann auch als Sonderfall der Fuge angesehen werden. Die Konstruktion seiner Melodie bedarf der besonderen Sorgfalt des Komponisten.

Frè - re Jac - ques, Dor - mez - vous ? Son - nez les ma - ti - nes, Ding, daing, dong ! *etc.*

Frè - re Jac - ques, Dor - mez - vous ? Son - nez les ma - ti - nes, *etc.*

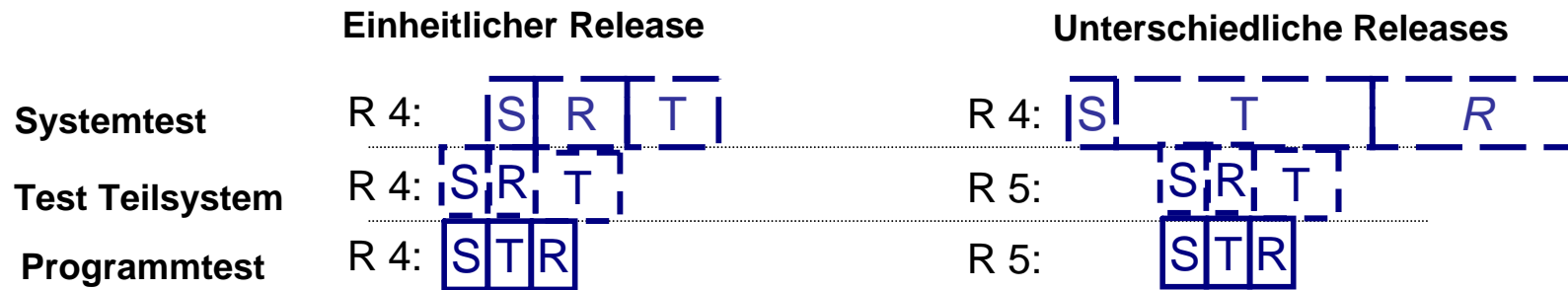
Frè - re Jac - ques, Dor - mez - vous ? *etc.*

Eigenschaften des Kanons



- Der Kompromiss: Gleichzeitiges Testen verschiedener Stufen verschiedener Versionen, schrittweise Testen einzelner Stufen – mit Überspringen.
- Hohe Ansprüche an Organisation, Planung und Koordination.
- Hohe Anforderungen an Änderungs- und Konfigurationsmanagement.
 - Fehlermeldungen zu „alten“ Releases (und zu bereits reparierten Fehlern!)

Der Kanon im Testen (1/2)



Ziel: Test-Mehraufwendungen vermeiden

1. Mittels (gestaffeltem) Smoke-Test die Testwürdigkeit ermitteln.
2. Nach Smoke-Test in unteren Test-Stufen testen, um Abbruch-Kriterien für höhere Test-Stufen zu provozieren. In oberen Test-Stufen die Zeit mit Regressions-Tests nutzen.
3. In höheren Test-Stufen den Test-Lauf mit den primären Tests abschliessen – in den unteren Test-Stufen umgekehrt.

Ziel: Schnelles Feedback in Entwicklung

- In den unteren Test-Stufen wurde der alte Release (R4) für testwürdig befunden. Für rasches Feedback Systemtest mit (neuen) Tests beginnen.
- Die unteren Test-Stufen für den neuen Release (R5) werden unabhängig gestaltet.
- *Variante:* Zeit gewinnen und Regressions-Tests im alten Release (R4) auslassen.

S: Smoke Test; R: Regression Test; T: Test

Der Kanon im Testen (2/2)

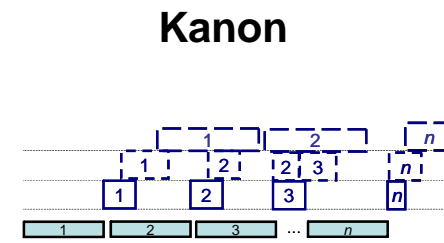
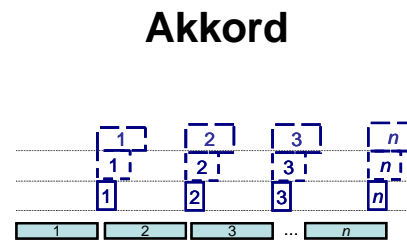
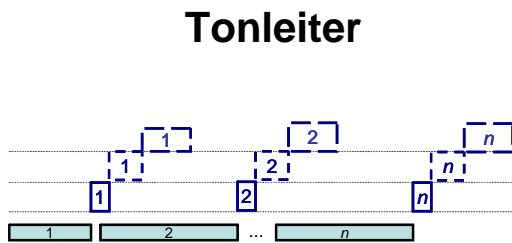
Kenne Deine Fehler

- Test-Arten und -Fälle so ausrichten, dass erwartete und vermutete Fehler und Fehler-Arten erkannt werden.
- Fehler-Arten:
 - Verarbeitung von gültigen Eingaben
 - Verarbeitung von ungültigen Eingaben und unerwarteten Situationen
 - Zusammenwirken von Teilen der Lösung
 - Verarbeitung von gültigen Eingaben in Abhängigkeit von der Systemkonfiguration
 - Antwortzeitverhalten, Durchsatz
 - Zusammenwirken mit andern Komponenten oder Systemen

Kenne Deine Abhängigkeiten

- Test-Fälle in höheren Test-Stufen sind teilweise redundant zu Test-Fällen in unteren Test-Stufen.
(Beispiel End-To-End System-Test prüft implizit Aspekte der Schnittstellen.)
- Redundanzen zwischen Test-Fällen in verschiedenen Test-Stufen in der Testplanung berücksichtigen.
- Mit „schnellen“, „billigen“ Programmtests, teure System-Tests vermeiden

Vergleich von Tonleiter, Akkord und Kanon



Laufzeit des Projekts	Hoch	Niedrig	Mittel
Mehraufwand Entw.	Hoch	Niedrig	Mittel
Mehraufwand Test	Niedrig	Hoch	Mittel
Organisationsgrad (Change/Konfig Mgmt.)	Niedrig	Hoch	Mittel
Planungs- und Koordinationsaufwand	Niedrig	Hoch	Mittel
Bedarf Test- Ressourcen	Niedrig	Hoch	Hoch
Wissenstand Fortschritt	Hoch	Mittel	Tief

Gesamt-Test als Tonleiter, Akkord oder Kanon? Entscheidungshilfen

Kriterium	Tonleiter	Kanon	Akkord
Art des Projektes:			
Eigenentwicklung, Neu-Entwicklung, umfangreiche Weiterentwicklung (Major Release)	X		
Wartung, (Hot-)Fix, Patch, kleine Weiterentwicklung (Minor Release)			X
(Grosses) Integrationsprojekt (externe Entwicklung)		X	
Organisationsgrad, Prozesssicherheit	Niedrig	Hoch	Mittel
Kosten, Dauer, Qualität:			
Kurze Dauer – kritische Termine (Geld spielt keine Rolle)		X	X
Stabilität, Qualität, Kosten (Dauer spielt keine Rolle)	X		
Zeitabschnitt:			
Am Anfang im Projekt	X		
Mitten im Projekt		X	
Am Schluss des Projektes, vor/während des Rollouts			X

Wie spielt die Musik im Test?

... ganz verschieden – entscheidend ist der richtige Komposition.

- Unterschiedliche Arten von Fehlern in Testfällen je Test-Stufe spezifizieren.
- Hohe Test-Abdeckung in unteren Stufen erreichen (wenn möglich automatisiert). Augenmerk auf Verarbeitungsfehler und Fehlverhalten.
- Unnötige Tests vermeiden, frühe Abbruch-Kriterien im Test-Lauf erzwingen.
- Mit der einfachen **Tonleiter** beginnen: Produkt kennenlernen, Risiko-Bereiche identifizieren, kostengünstige „Killer“-Tests identifizieren
- Mit zunehmendem Test-Umfang in den **Kanon** wechseln: Testen verschiedener Aspekte auf allen Stufen an unterschiedlichen Konfigurationen in verschiedenen Umgebungen.
- Mit abnehmender Änderungs- und Fehler-Rate in den Akkord wechseln: kurze, gezielte Testläufe auf allen Ebenen an einer Konfiguration.