

Live-Code-Umgebungen: Werkzeuge mit Agilität im Blut

Agile/Leane Softwareentwicklung

- wertorientiert
- inkrementell
- feedbackgetrieben
- experimentell

Softwareentwickler/in

- Geschäftsmann/-frau
- Ingenieur/in
- Künstler/in
- Forscher/in

Das Prinzip der direkten Verbindung

- Bret Victor, CUSEC 2012: „Inventing on Principle“
 - <http://vimeo.com/36579366>
 - „Creators need an immediate connection with what they create“
 - „Effects of decisions need to be seen immediately, there can't be things hidden“
- Werkzeuge müssen diese direkte Verbindung mit sofortiger Rückmeldung zur Verfügung stellen, um Ideen entwickeln zu können

Jackson Pollock

- <http://www.sfmoma.org/explore/multimedia/videos/250>
 - © Hans Namuth, Ltd.

A stylized, handwritten signature in black ink, likely reading 'Pollock', with thick, expressive strokes and a dynamic, calligraphic feel.

Programmierung

- Chris Lattner, Leiter Developer Tools bei Apple, verantwortlich für XCode und Swift: „A personal passion of mine, to make programming more interactive and approachable“
- Chris Granger, Entwickler von Light Table: „Light Table is an IDE that lets you modify running programs. It provides the real time feedback we need to not only answer questions about our code, but to understand how our programs really work.“
- „All ideas are old ideas“

Die Riesen,

- auf deren Schultern wir stehen:
 - Doug Englebart (The Mother of all Presentations)
 - Larry Tesler („Don't mode me in“)
 - Alan Kay (Dynabook, „The best way to predict the future is to invent it“)
 - Adele Goldberg (Learning Research Group, Story Telling)
- Alle hatten eins gemeinsam: die Vision, den Computer dem einzelnen Menschen zu nutze zu machen, eben das „Personal Computing“

Erkenntnisse

- Software dient einem Zweck und muß Nutzen bringen
- Softwareentwicklung ist Modellierung von Wirklichkeit
- Die Wirklichkeit und unser Wissen darüber ändern sich mit der Zeit

Folgerungen

- Reflexivität, Offenheit, Änderbarkeit des gesamten Systems
- Entwicklungswerkzeuge sind wesentlicher Teil des Systems
 - Sie müssen einfach, intuitiv und dynamisch sein
 - Programmiersprache und -paradigma sind Teil der UI!
- => Objektorientierung als Methode, Smalltalk als Plattform und Referenzimplementierung

Künstler oder Ingenieur?

- Was ist wichtiger: das „wie“? Oder das „was“?
- Jackson Pollock: „Technique is just a means of arriving at a statement“
- Für den Entwickler ist der Sourcecode die Materie, mit der er arbeitet und sich ausdrückt
- Sprache ist die mächtigste Ausdrucksmöglichkeit des Menschen
 - Domain Driven Design: ubiquitous language

Dead or Alive?!

- Dead-Code-Umgebung:
 - Code als Text in separaten Dateien
 - Ausführbarmachen durch Kompilieren und ggfs. Linken
 - Werkzeuge und Plattform sind nicht integriert
- Live-Code-Umgebung:
 - Während der Laufzeit änderbare Anwendung
 - Vollständig integrierte Plattform mit Entwicklerwerkzeugen
 - Code besteht aus „ganz normalen“ Objekten

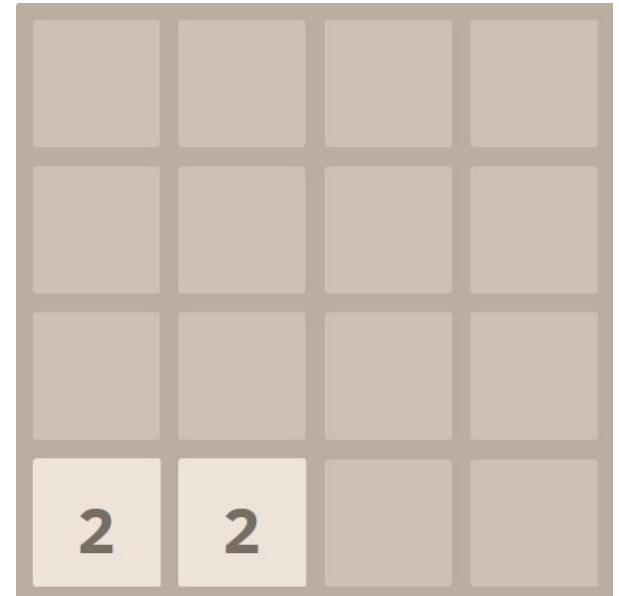
Smalltalk

- Vorbild für viele
- Innovationswerkzeug
- Reif – wie guter Port
- Aktuell – wie immer

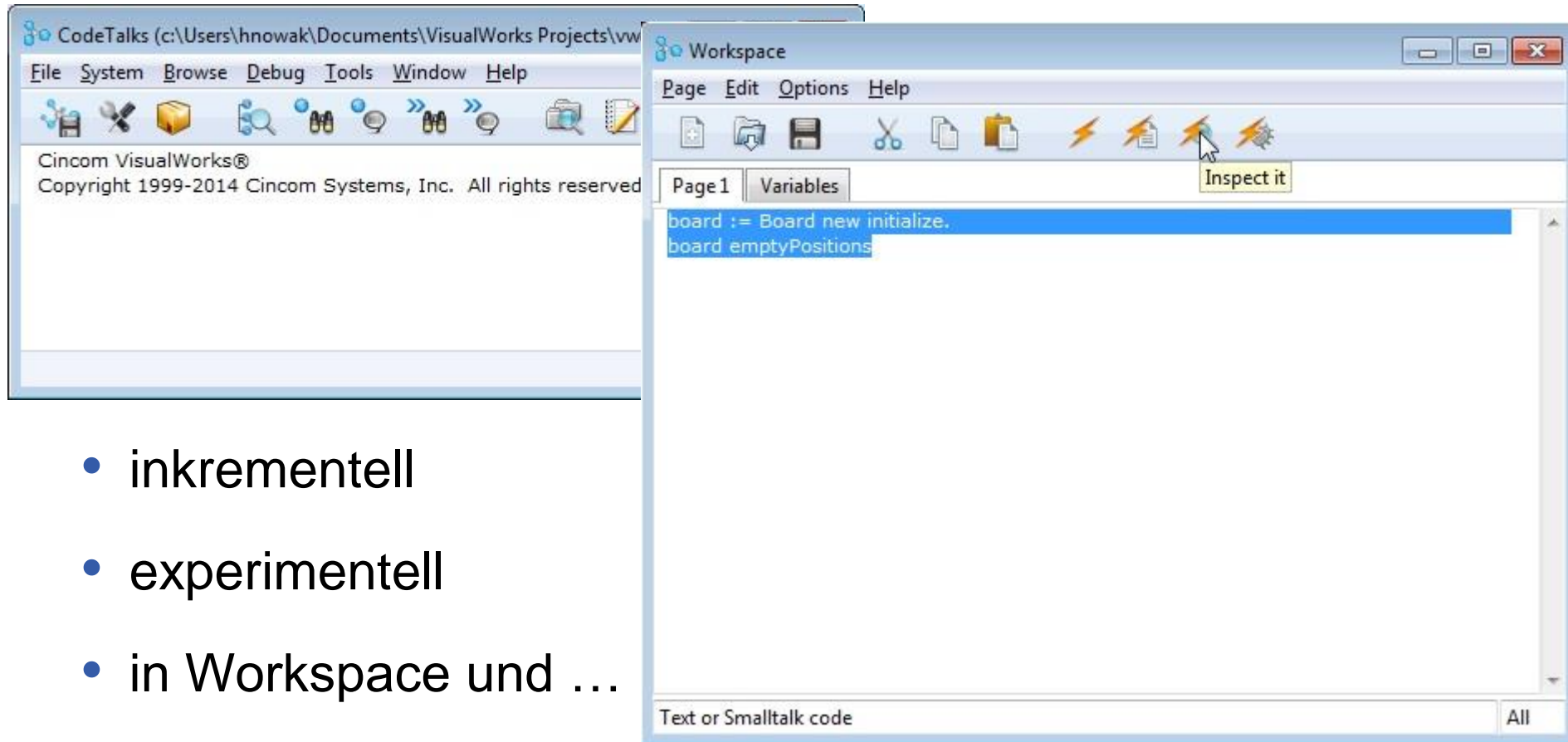


Demo

- 2048
- <http://gabrielecirulli.github.io/2048/>



Explorative Entwicklung ...

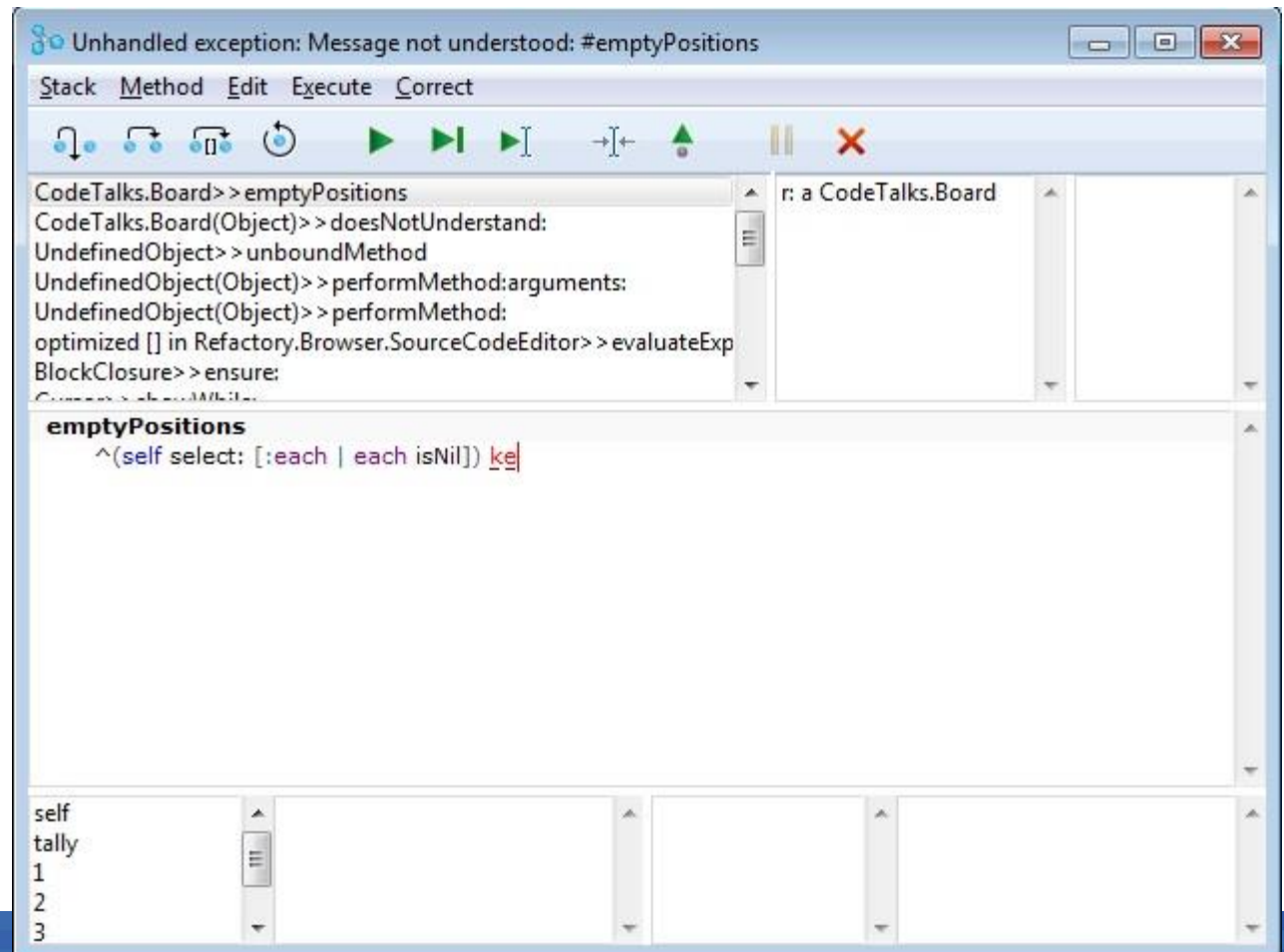


The screenshot shows two windows from the Cincom VisualWorks IDE. The left window is titled 'CodeTalks' and shows a menu bar with 'File', 'System', 'Browse', 'Debug', 'Tools', 'Window', and 'Help'. Below the menu is a toolbar with various icons. The main area contains the text: 'Cincom VisualWorks® Copyright 1999-2014 Cincom Systems, Inc. All rights reserved'. The right window is titled 'Workspace' and has a menu bar with 'Page', 'Edit', 'Options', and 'Help'. It features a toolbar with icons for file operations and a lightning bolt icon. Below the toolbar are tabs for 'Page1' and 'Variables'. The main area contains the code: 'board := Board new initialize.' and 'board emptyPositions'. A yellow tooltip labeled 'Inspect it' is positioned over the lightning bolt icon. At the bottom of the workspace window, it says 'Text or Smalltalk code' and 'All'.

- inkrementell
- experimentell
- in Workspace und ...

... Explorative Entwicklung

- ... direkt im Debugger:
 - Code Ändern
 - Akzeptieren
 - Weiterlaufen lassen



Spaß, aber nachhaltig!

- Befriedigung durch den schöpferischen Akt
 - Live-Code: viele kleine sofort erfahrbare Erfolgserlebnisse
 - Führt in den „Flow“
- Spaß und Zufriedenheit erzeugen intrinsische Motivation
- Motivation bringt gute Ergebnisse hervor
- was Spaß macht und Zufriedenheit schafft ...

Spaß, Lernen und Arbeiten

- Chris Lattner: „I hope that by making programming more approachable and fun, we'll appeal to the next generation of programmers and help redefine how Computer Science is taught“
- Scratch, Etoys (Squeak)
- NORDAKADEMIE – Hochschule der Wirtschaft, Elmshorn, Prof. Johannes Brauer
 - Didaktischer „Objects-first“-Ansatz: Einführung in die Programmierung von Anfang an mit der OO-Methodik, ohne vorher andere Paradigmen zu lehren

Qualität

- Blue Screen, Null-Pointer-Exception, Internal Server Error 500
- Schlechte UI
- Falsche Funktionalität
- Möglicherweise katastrophale Folgen

Fehler

- Testen
- Testen
- Testen
- Automatisierung
 - SUnit, Ursprung aller XUnit-Frameworks
 - Fit, FitNesse
- Mächtige Debugging-Umgebung erforderlich

Richtige Funktionalität

- Kann nur der Fachexperte bestimmen und verifizieren
- eXploratory Modelling
- Heinz Ulrich Roggenkemper, SAP: „Result: surprising progress within three weeks“
 - „Much better understanding of the problem and the data“
 - „Clarification of core concepts“
 - „Flexible implementation of algorithms“
 - „Support for case building strategies“

Gute UI

- Kann nur mit den Anwendern zusammen entwickelt werden
- RAD
- On-Site Customer und Whole Team Approach

Entwicklung ist Teamarbeit

- Kommunikation über den Code
 - Lesbarkeit, selbsterklärender Code
- Prozesse und Werkzeuge

Teamarbeit in Echtzeit

- Multi-User-Live-Code => Real-Time Collaboration
 - Synchronous Groupware (TUKAN, VITAL usw. basierend auf COAST, am GMD-IPSI Darmstadt)
 - Croquet (collaborative, live-programming, audio-visual, 3D environment, Squeak)
 - Wolf Pack Programming
 - Swarming, Mob Programming

Prozeßintegration

- Continuous Integration, Continuous Delivery
- Integration aller Teilprozesse der Softwareentwicklung
- Live-Code: Integration von Integrationstest und Entwicklung, schnelle Fehlerbehebung durch
 - Sofortige Nachvollziehbarkeit und Fehlerbehebung direkt im Debugger, ohne das Problem aufwendig nachstellen zu müssen
 - Automatische Zuweisung an einen Entwickler oder Collective Code Ownership

Aus Geschäftssicht

- Höhere Produktivität
- Bessere Lösungen
- Nachhaltig anpaßbar an sich ändernde Anforderungen
- Neue Marktchancen
- Schnelle Marktverfügbarkeit
- Hoch profitabel

Praxisbeispiel: ChannelStream

- Multi-Kanal-Druckmanagement
- Einfach zu bedienende Web-Oberfläche
- Integration in unterschiedlichste Systemlandschaften
- *Ein* Produkt mit sehr kundenindividueller Ausprägung
- Keine Programmierung, kein Scripting auf Kundenseite

ChannelStream Entwicklung

- Gemeinsame Basis, individuelle Ausprogrammierung
- Parallele Entwicklung und Continuous Delivery
- Technologie und Werkzeuge:
 - Technologie und IDE: Cincom Smalltalk VisualWorks
 - Repository: Store
 - Webframework und Webapplikationsserver: Seaside
 - Web-Tests: Selenium mit Beach Parasol
 - Continuous Delivery: Jenkins, gesteuert durch Smalltalk

on: Debugger do:

- SMTPClient defaultOutgoingMailClient sendMessage: alertMailMessage.
- ObjectMemory snapshotThenQuit.

Helge Nowak

hnowak@cincom.com

Twitter: @nowagil

CINCOM and the Quadrant Logo are registered trademarks of Cincom Systems, Inc.
All other trademarks belong to their respective companies.

© 2015 Cincom Systems, Inc.
All rights reserved