

Agile Prozesse



Vortrag: Fredrik Gundelsweiler & Thomas Memmel

Universität Konstanz

FB Informatik und Informationswissenschaft

1. Entstehung der agile SE

- a. Einführung
- b. Gründung der Agile Alliance
- c. Was bedeutet „Agil“? Prinzipien und Denkweisen
- d. Wann ist eine Technik/Methode „agil“?
- e. Wie wird ein Projekt „agiler“?

2. Ausgewählte Agile Methoden

- a. XP – Extreme Programming
- b. Agile Modeling

3. Beispiel: AUCSL Projekt



1 Einführung

- Softwareentwicklung kennt viele Methoden vom Wasserfallmodell bis hin zu aufwendigen UE-Lifecycles
- Probleme bei Softwareprojekten stehen Qualität und Wartbarkeit gegenüber:
 - **Änderungen** durch Kunden oftmals **spät** im Software Lifecycle
 - schnellere **Releasezyklen**
 - immer kürzere **time-to-market**
- deshalb gilt für immer mehr Projekte:
 - Großer **Zeitdruck**
 - aber **Qualität** der Ergebnisse wichtig
 - oft frühe **Zwischenergebnisse** notwendig
 - Anforderungen sind aber meist **vage** und oder **unklar**
 - Auf Änderungen (Anford., Rahmenb.) muss **flexibel** reagiert werden

Gründung der Agile Alliance

- Entstehung der „Agile Alliance“
 - <http://www.agilemanifesto.org/>
 - <http://www.agilealliance.com/home>
 - Werte/Normen der Agile Alliance:
 - Individuals and interactions *over* processes and tools
 - Working software *over* comprehensive documentation
 - Customer collaboration *over* contract negotiation
 - Responding to change *over* following a plan
 - Unterstützung der Einführung dieser Methoden
 - „Non-profit Organisation“

[Webseite: http://www.agilemanifesto.org/principles.html](http://www.agilemanifesto.org/principles.html)

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions** over processes and tools
- Working software** over comprehensive documentation
- Customer collaboration** over contract negotiation
- Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

- | | | |
|-------------------|----------------|------------------|
| Kent Beck | James Grenning | Robert C. Martin |
| Mike Beedle | Jim Highsmith | Steve Mellor |
| Arie van Bennekum | Andrew Hunt | Ken Schwaber |
| Alistair Cockburn | Ron Jeffries | Jeff Sutherland |
| Ward Cunningham | Jon Kern | Dave Thomas |
| Martin Fowler | Brian Marick | |

Was bedeutet „Agil“? - Prinzipien und Denkweisen

- Grundwerte, Prinzipien und Praktiken* sind:
 - höchste Priorität: **Kundenzufriedenheit**
 - früh und häufig laufende Software ausliefern
 - man weiss nicht alles → Lernen im Prozess
 - Einfachheit
 - unnötige Arbeit vermeiden
 - auch späte Änderungen willkommen
 - face-to-face **Kommunikation**
 - regelmäßige Überprüfung und Verbesserung der eigenen Effektivität
 - Feedback auf mehreren Ebenen

*: aus <http://www.agilemanifesto.org/principles.html>

Was bedeutet „Agil“? - Prinzipien und Denkweisen II

*nach Frühauf,
Conquest 2001*

	Bisheriger Ansatz	Agiler Ansatz
Mitwirkung des Kunden	unwahrscheinlich	kritischer Erfolgsfaktor
Etwas Nützliches wird geliefert	erst nach einiger (längerer) Zeit	mindestens alle sechs Wochen
Das Richtige entwickeln durch	langes Spezifizieren, Vorausdenken	Kern entwickeln, zeigen, verbessern
Nötige Disziplin	formal, wenig	informell, viel
Änderungen	erzeugen Widerstand	werden erwartet und toleriert
Kommunikation	über Dokumente	zwischen Menschen
Vorsorge für Änderungen	durch Versuch der Vorausplanung	durch „flexibel Bleiben“



Wann ist eine Technik/Methode „agil“?

- Die wichtigen Dinge zuerst machen!
 - Die Features mit dem höchsten Mehrwert zuerst realisieren.
 - Geschäftswert = $f(\text{Kosten, Zeit, Funktionalität, Qualität})$
- Zyklischer und inkrementeller Entwicklungsprozess:
 - Früh und häufig ausliefern
 - Entwicklungsprozess möglichst leichtgewichtig gestalten.
 - Passe den Prozess bei Bedarf an eine veränderte Umgebung an.
- Eine Technik/Methode kann als agil angesehen werden, wenn sie:
 - ihre Aufgabe erfüllt
 - **verständlich** ist
 - **ausreichend** genau, konsistent und detailliert ist
 - einen positiven Wert darstellt (**Nutzen** > Kosten)
 - so **einfach** wie möglich ist



Wie wird ein Projekt „agiler“?

- Druck reduzieren durch leicht-gewichtige Ansätze
 - Unnötige Dokumente einsparen
 - Vorgaben und Templates “verschlanken”
- vage Anforderungen und Änderungen einkalkulieren
 - schnell zum Kernsystem, Weiterentwicklung inkrementell
- besseres Feedback
 - konkrete organisatorische und technische Maßnahmen
 - engere Kundeneinbindung
- Einsatz agiler Methoden und Techniken
 - Pair Programming, Unit Tests, inkrementelles Vorgehen

2 Ausgewählte Agile Methoden

- **XP** – „Extreme Programming“ (Kent Beck)
- **Agile Modeling** (Scott Ambler)

Hier nicht behandelt:

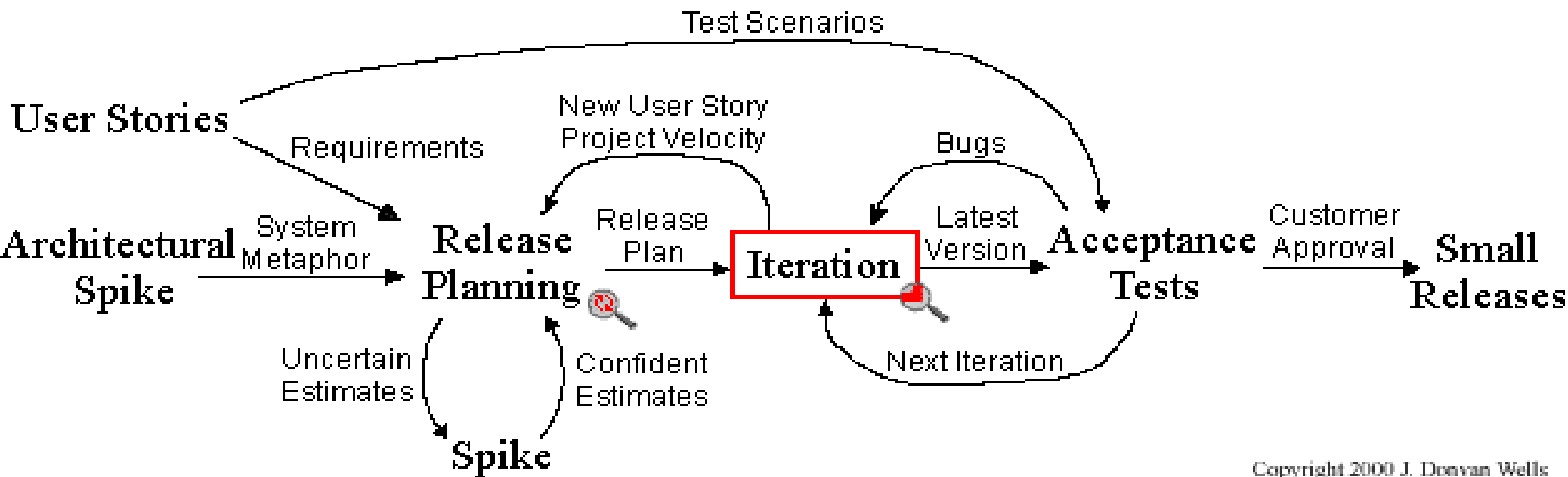
- Crystal Familie (Alistair Cockburn)
- Feature Driven Development (Stephen R. Palmer)
- SCRUM (Ken Schwaber, Mike Beedle)
- ... viele andere ...
- Agile Usage Centered Software Lifecycle (Reiterer, Gundelsweiler, Memmel)



XP – Extreme Programming – Gesamtprozeß



Extreme Programming Project



Copyright 2000 J. Donovan Wells

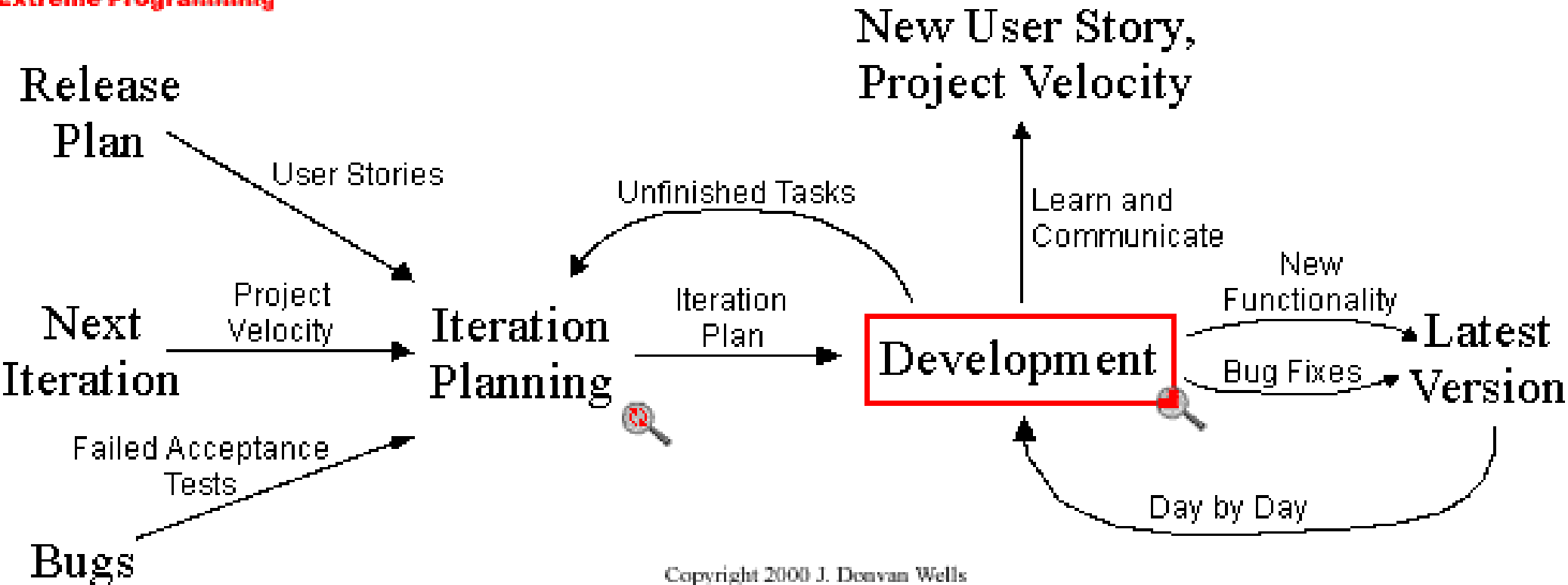
Grafik aus <http://www.extremeprogramming.org/> (Don Wells)

XP – Extreme Programming – Iteration Planning



Iteration

Zoom Out



Copyright 2000 J. Donovan Wells

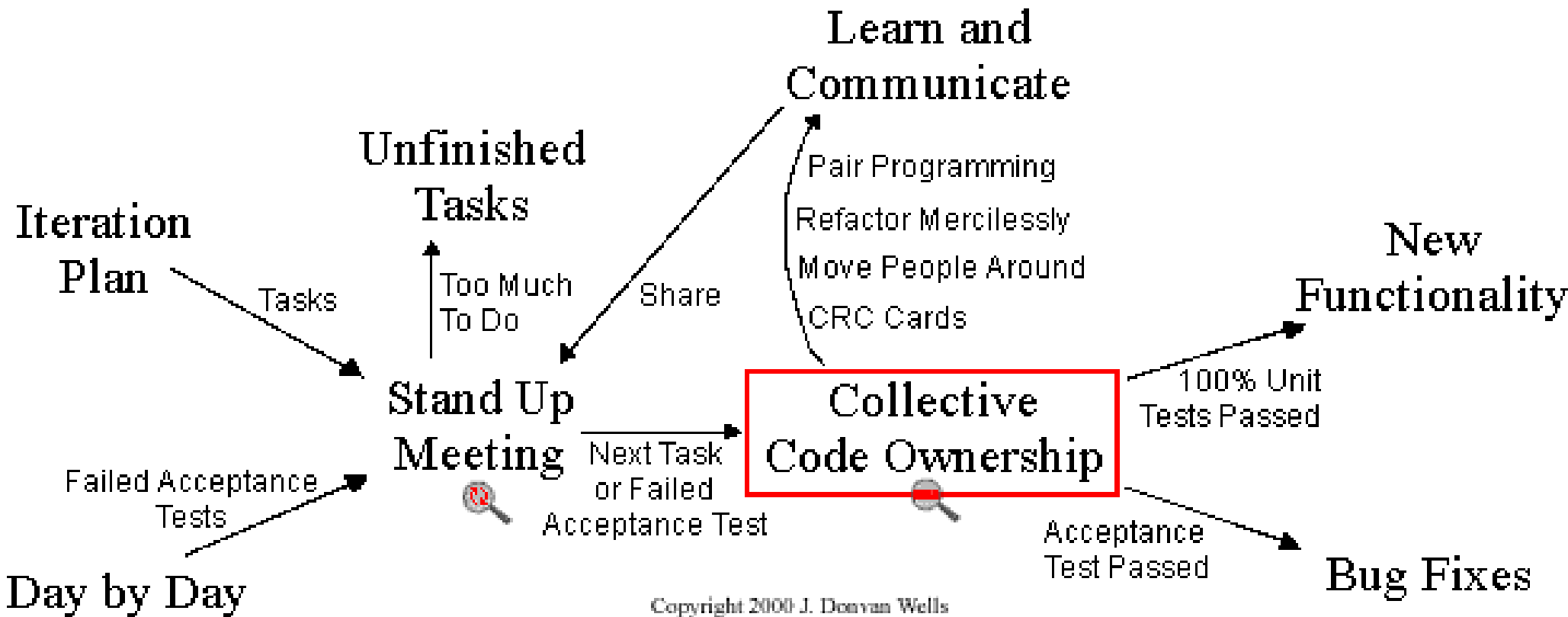
Grafik aus <http://www.extremeprogramming.org/> (Don Wells)

XP – Extreme Programming – Iteration Planning



Development

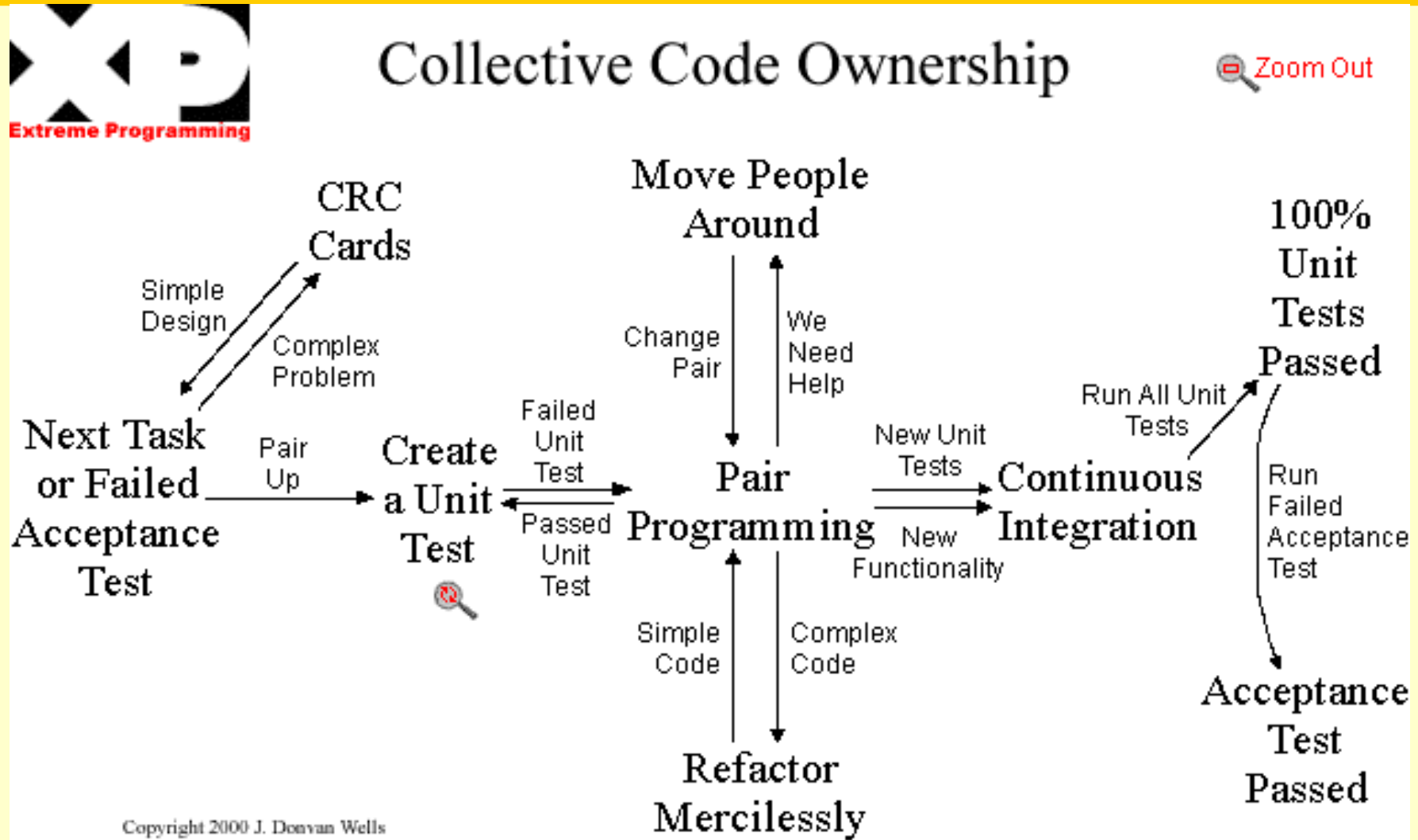
Zoom Out



Copyright 2000 J. Doornik Wells

Grafik aus <http://www.extremeprogramming.org/> (Don Wells)

XP – Extreme Programming – Test/Programmierung



Grafik aus <http://www.extremeprogramming.org/> (Don Wells)

Agile Modeling (Scott W. Ambler)

Definition

- „Techniken-basierter Prozess, wie man wirksam modelliert.“

Das heißt ...

- Auf verschiedene Entwicklungsprozesse anwendbar
- Modellieren ist Voraussetzung für Anwendung

Interessanter, komplementärer Ansatz

- Modelle kommen in fast allen (auch agilen) Ansätzen vor
- Modellierung ist nur ein Teil der Aufgabe

Wieso ist es dann eine „agile Methode“?

- Basiert auf den selben Werten (wir wissen nicht alles)
- Ist agil aufgebaut: Werte-Prinzipien-Praktiken
- Verwendet viele Prinzipien der Agile Alliance
- Erweitert die Praktiken auf beliebige Modelle
- handelt von Modellen, nicht von Programmierung



Agile Modeling (Scott W. Ambler)

„To change your fate, you must first change your attitude“

- Beruft sich auf die Werte des Agile Manifesto
- Kein Soll-Prozess, der detaillierte Schritte vorschreibt

Werte:

- Communication, Simplicity, Feedback, Courage – und Humility
- Humility führt zu Respekt und anderem Verhalten gegenüber dem Team

Drei Ziele:

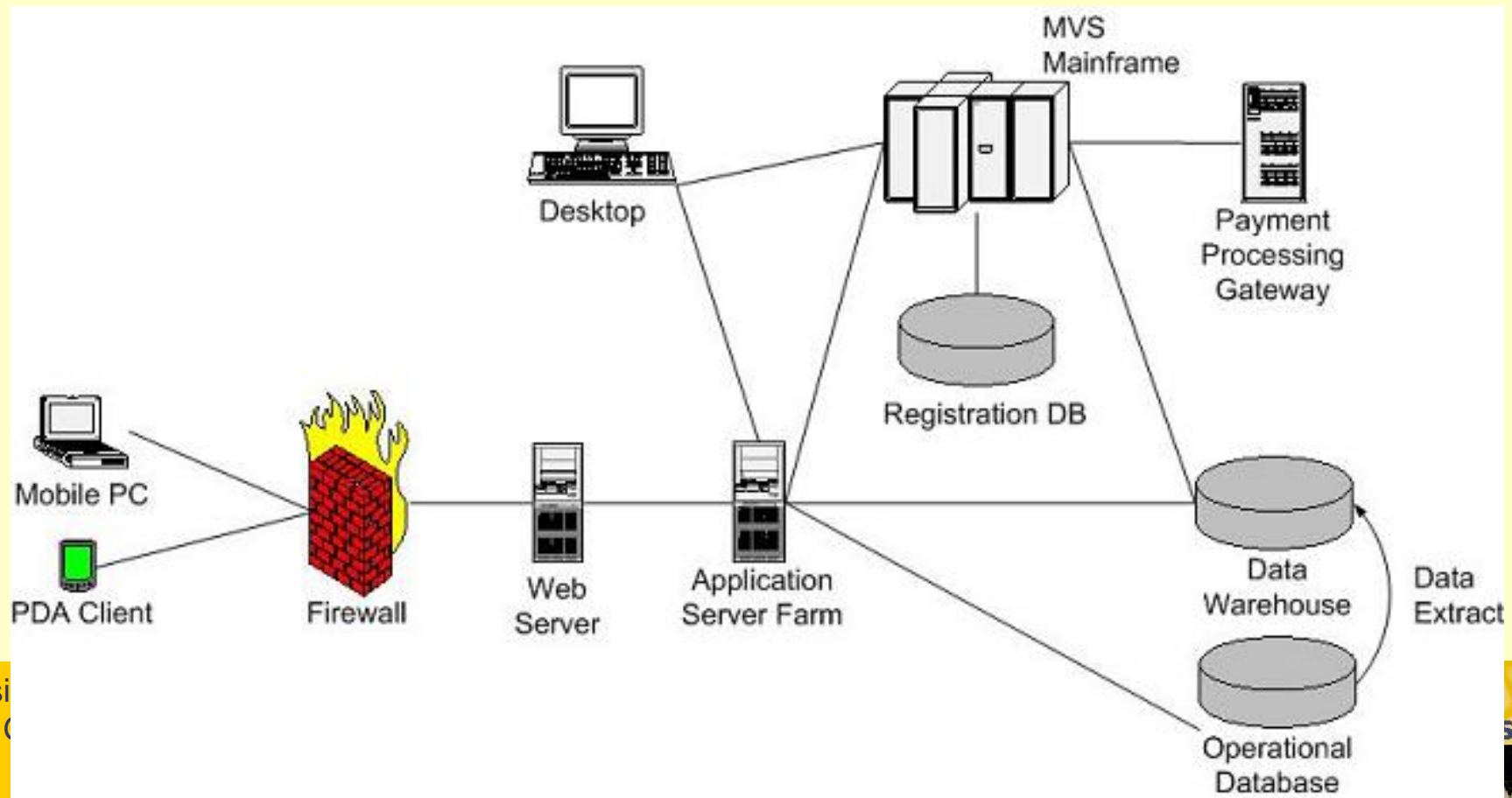
- Es geht nicht um die Modellierungsarten, sondern um deren Einsatz
- Zeigen, wie man agil und wirksam mit Modellen arbeitet
- Zeigen, wie man erfolgreich „fast-agil“ vorgehen kann

„Modeling is similar to planning – most of the value is in the act of modeling, not the model itself.“



Agile Modeling (Scott W. Ambler) – Agile Methoden

- Übersicht über die verschiedenen Modellierungstechniken unter: <http://www.agilemodeling.com/artifacts/index.htm>
- Viele UML Techniken (z.B. auch UML State Machine Diagram)
- Beispiel Network Diagram:



Workshops

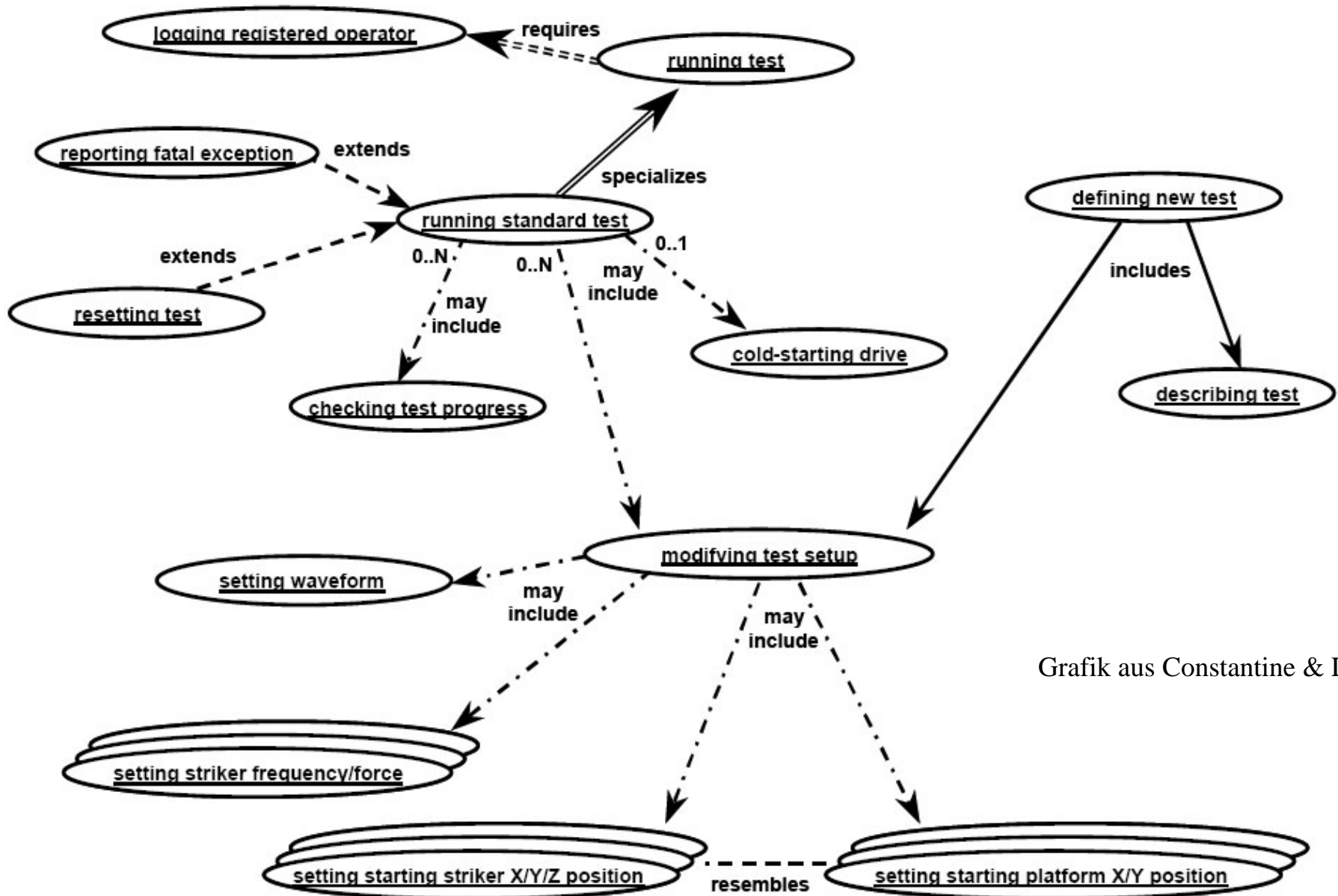


Use Cases

Identification	
ID	Name
Contextual Purpose	Supported Roles
Relationships	
Specializes	Extends
Resembles	Equivalentents
Process	
Preconditions	
User Intentions	System Responsibilities
Asynchronous Extensions	Asynchronous Extensions
(steps)	(steps)
Post-conditions	
Business Rules	

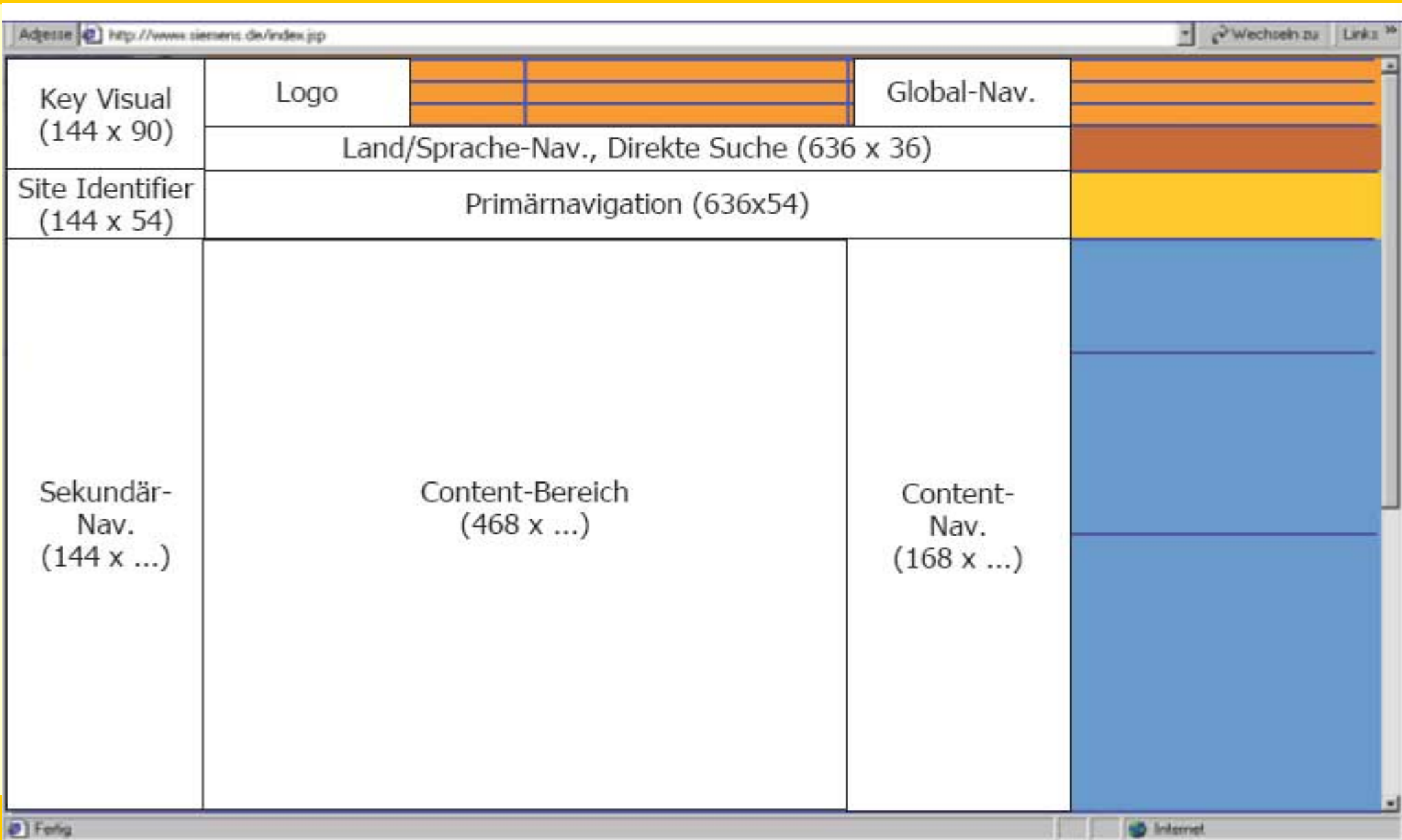
<u>getting cash</u>	
USER INTENTIONS	SYSTEM RESPONSIBILITIES
	1. Request identity.
2. Identify myself.	3. Verify identity.
	4. Offer choices.
5. Make choice.	7. Give requested cash.
7. Take cash.	

Use Case Map



Grafik aus Constantine & Lockwood

Fibonacci UI Pattern



Ende – Vielen Dank!



Quellen

- Agile Modeling / eXtreme Programming
 - **Agile Modeling** – Scott W. Ambler. John Wiley & Sons.
 - **Extreme Programming Explained** – Kent Beck. Addison-Wesley, 1999
 - **Agile Software Development** – Alistair Cockburn. Addison-Wesley, 2001
 - **Integrating User Centered Design in Agile Development processes** – 2003, EC Wise, Inc.
 - **Process Agility and Software Usability: Toward Lightweight Usage-Centered Design** – Larry L. Constantine, Constantine & Lockwood, Ltd., University of Technology Sydney, 2002 *Information Age*
 - **Usage-Centered Engineering for Web Applications** - Larry L. Constantine, Constantine & Lockwood, Ltd.
 - **Adopting User-Centered Design within an Agile Process: A Conversation** – William Hudson, Syntagm Ltd., Design for Usability, Abingdon, UK, whudson@syntagm.co.uk
 - **Is Design Dead?** – Fowler, Martin. Thought Works. Februar 2001.
 - **Agile under Constraints** – Schneider, Kurt et. Al; DaimlerChrysler AG Research Center Ulm
 - **The New Methodology** – Martin Fowler, April 2003, www.martinfowler.com/articles/newMethodology.html

Quellen II

- Customer-Rolle
 - Thorup, Lars and Jepsen, Ole: Report on XP Workshop at JA00 2003 - Improving Customer Developer Collaboration.
 - van Deursen, Arie: Customer Involvement in Extreme Programming, 2001
 - Rising, Linda: The Product Champion, 2003
 - Farell, C., Narang, R., Kapitan, S. and Webber, H. Towards an Effective Onsite Customer Practice, 2002
 - Wallace, N., Bailey, P. and Ashworth, N. Managing XP with Multiple or Remote Customers, 2002
 - Lippert, M., Roock, S., Henning, W., Zullighoven, H. XP in Complex Project Settings: Some extensions, 2001
 - L.A. Griffin. A Customer Experience: Implementing XP, 2001
 - Shukla, A. and Williams, L. Adapting Extreme Programming in for a Core Software Engineering Course, 2002
 - Schuh, P. Recovery, Redemption and Extreme Programming, 2001

Quellen III

- Usability Engineering
 - **Software for Use: A Practical Guide to the Methods of Usage-Centered Design (ACM Press)** von Larry L. Constantine, Lucy A. D. Lockwood, 1. April 1999
Addison-Wesley
 - **Lightweights, Heavyweights and Usable Processes for Usable Software**, Larry Constantine Folien 2001
 - **The Usability Engineering Lifecycle: A Practitioner's Handbook for User Interface Design** von Deborah J. Mayhew, März 1999 Morgan Kaufmann Publishers
 - **Interaction Design** von Jennifer Preece, Yvonne Rogers, Helen Sharp, 31. Jan. 2002 John Wiley and Sons Ltd
 - **Scenario-Based Usability Engineering** von Mary B. Rosson, John M. Carroll, 1. Oktober 2001 Morgan Kaufmann Publishers
 - **Einführung Usability Engineering** von Daimler Chrysler, 20.06.2001