

Themenccluster: Agile und schlanke Methoden

Thema: Extreme Programming

Dr. Walter Rafeiner-Magor

09.09.2013

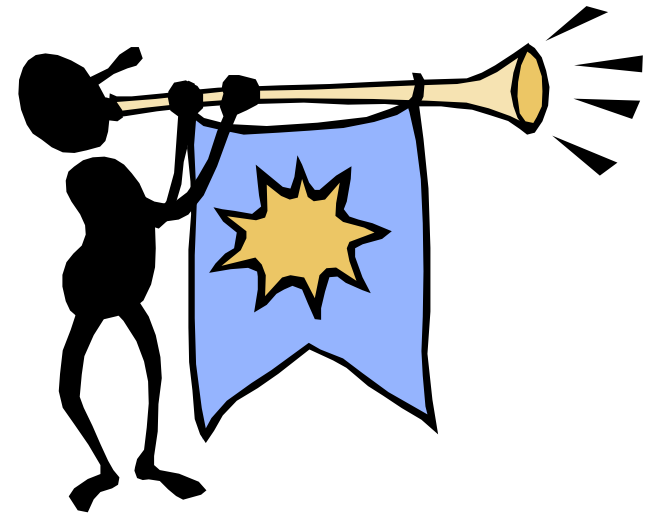
Extreme Programming (XP)

- **Ausgangspunkt:**

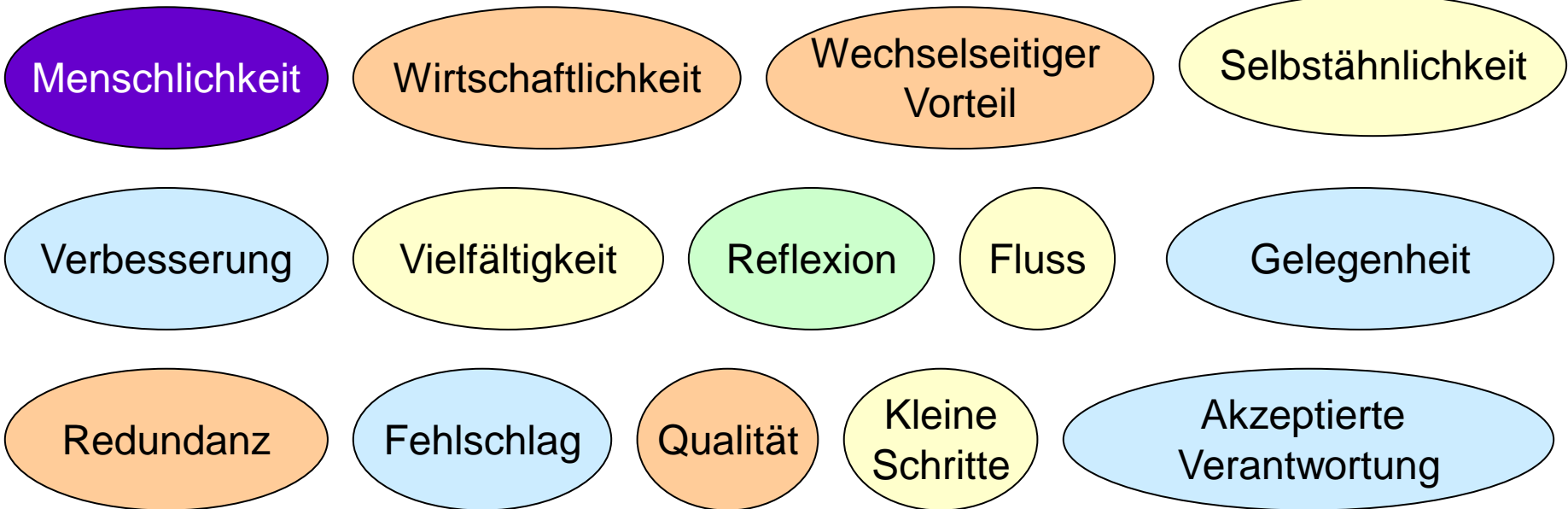
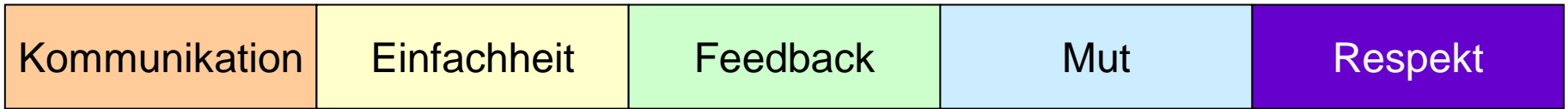
- Ein (beinahe) gescheitertes Projekt bei Chrysler (C3, 1996)
- Aus einer informellen agilen Praxis ein neues Prozessmodell

- **Herausforderungen:**

- Häufige Änderungen in den Spezifikationen
- Funktionsumfang nicht endgültig geklärt
- Hohe Produktivität der Programmierer
- Hohes Projektrisiko
- Automatisiertes Testen



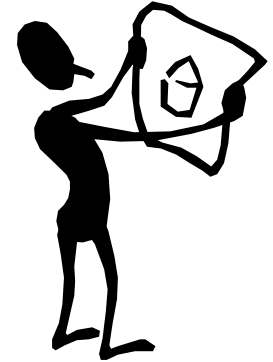
XP: Werte und Prinzipien



XP: Rollen

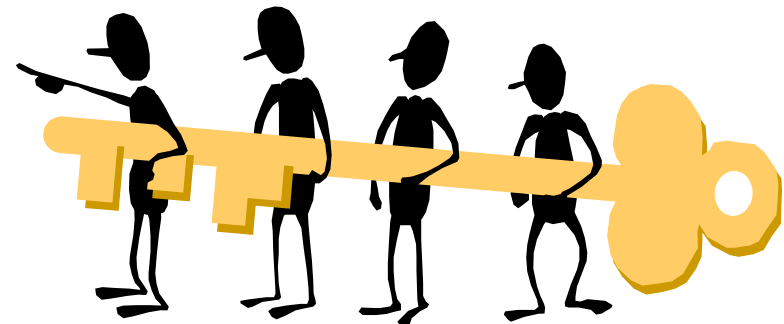
● Ausgangspunkt:

- XP-Teams sind selbstorganisiert
- XP-Teams sind funktionsübergreifend
- XP-Teams bekommen alles Expertenwissen, um die Anordnungen umsetzen zu können

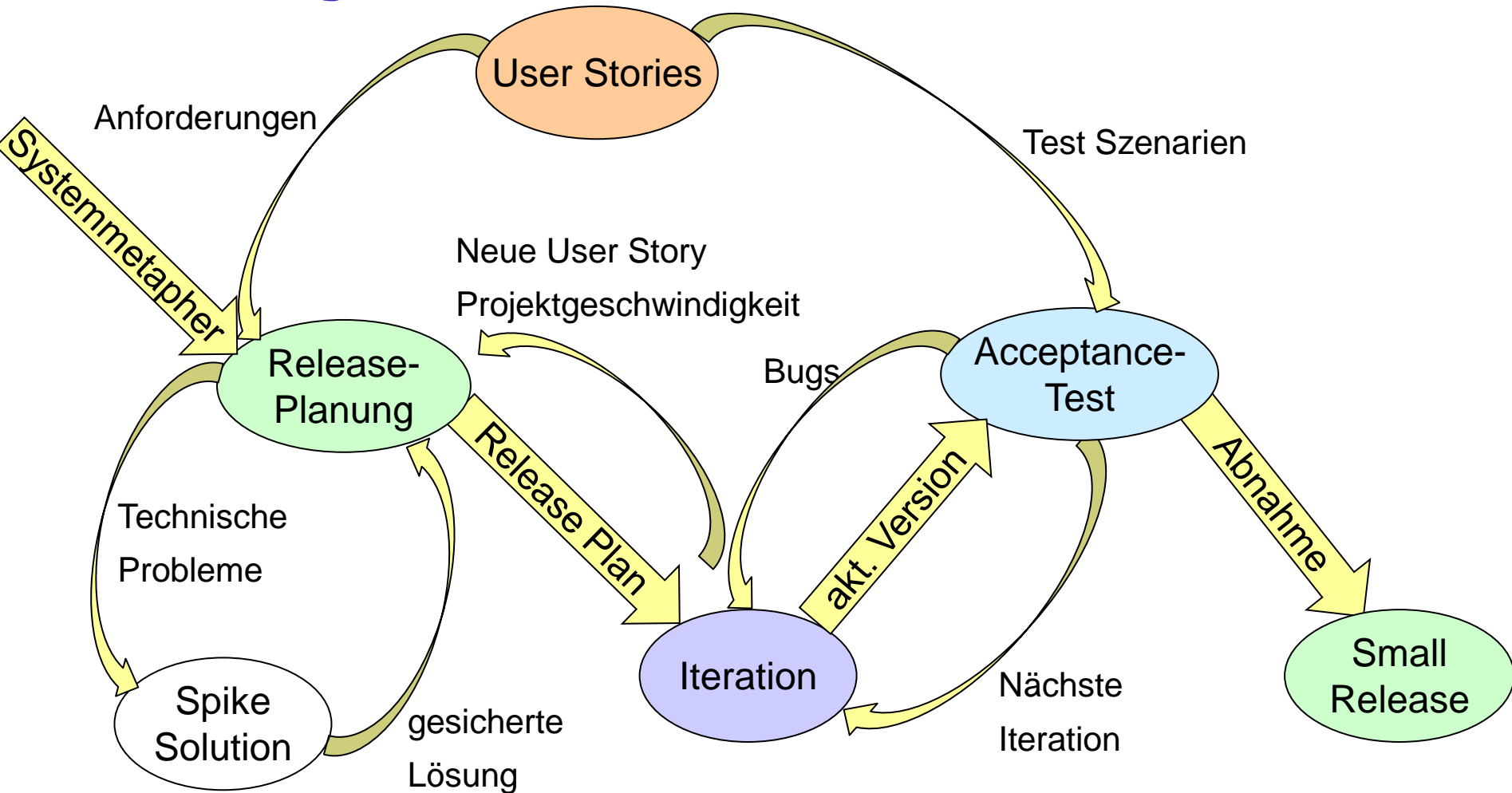


● Rollen:

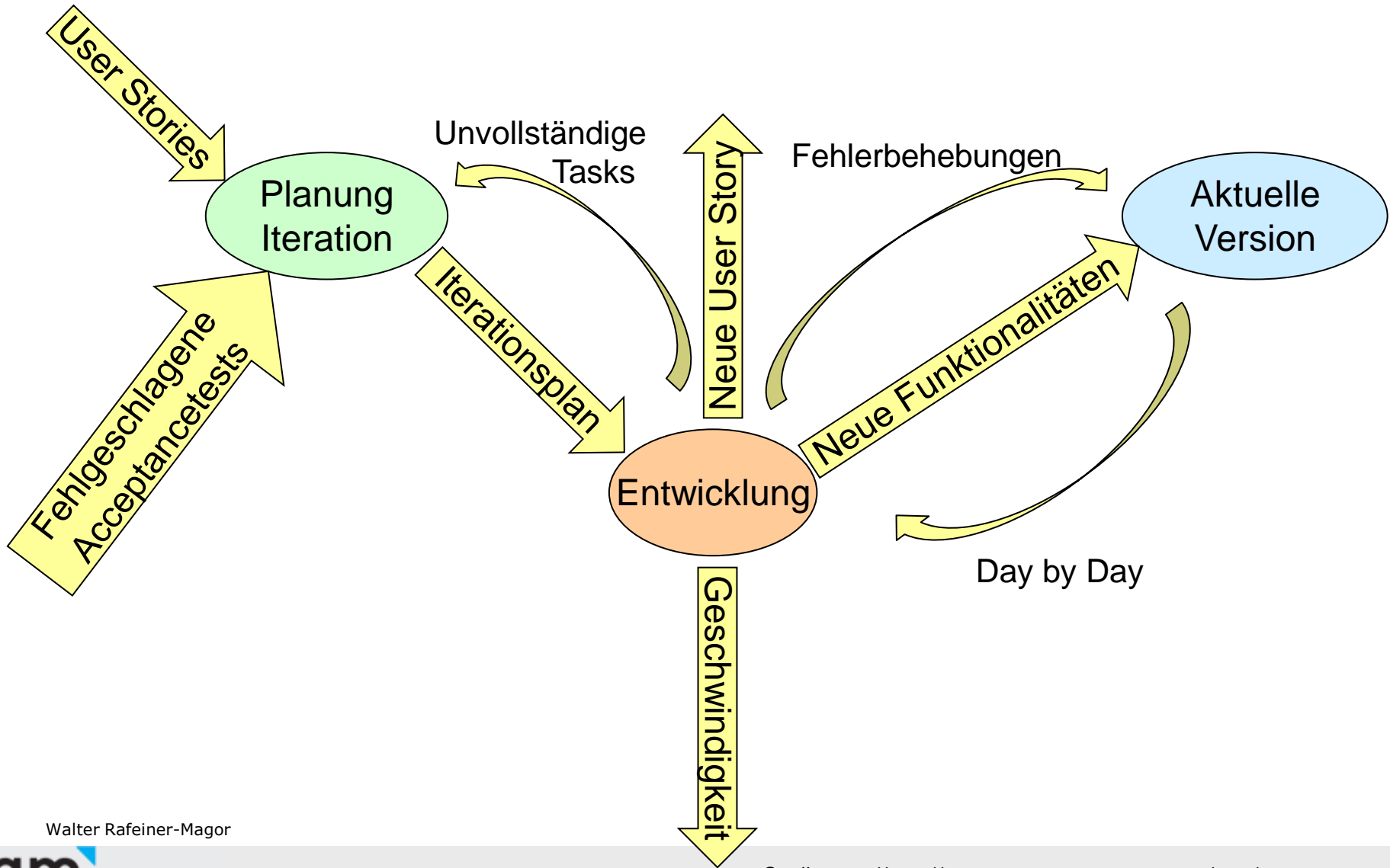
- Business-Analysten (Kunde)
- Implementierungsexperten (Architekten, Programmierer)
- Qualitätsexperten (Tester)
- Je nach Anforderung:
 - Projektmanager
 - Coach
 - Fachexperten



XP: Projektablauf



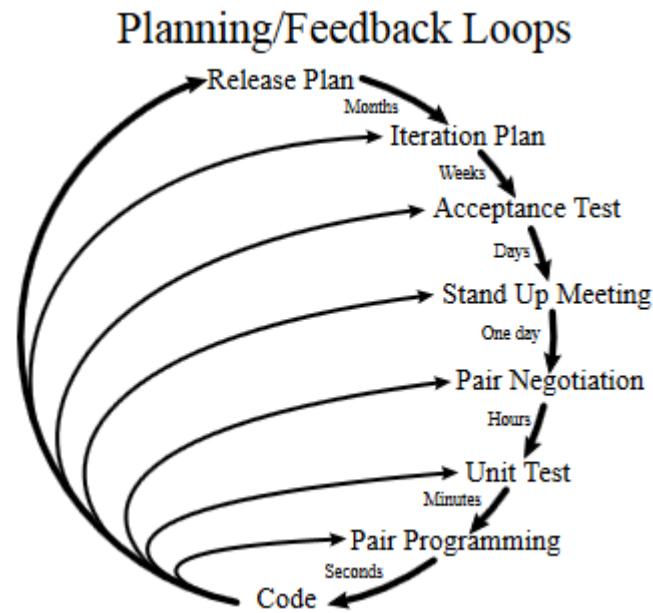
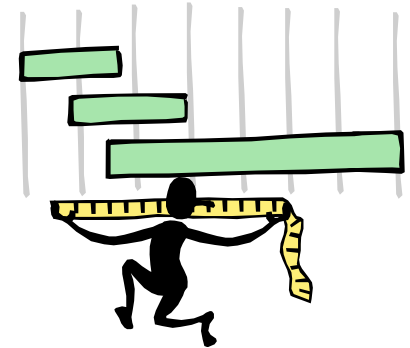
XP: Iteration



XP: Regeln

● Planung:

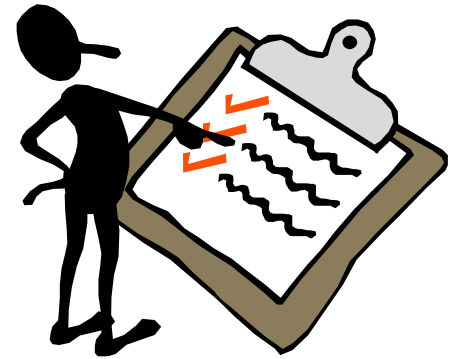
- User Stories werden vom Kunden geschrieben
- Release Planung: 80 +/- 20 User Stories
- Projekt ist in Iterationen geteilt
- Am Start jeder Iteration gibt es eine Iterationsplanung



XP: Regeln

● Coding:

- Kodierung erfolgt nur auf Basis gemeinsam festgelegter Standards
- Pair Programming: Immer 2 Programmierer an einem PC
- Collective Ownership: Jedes Team-Mitglied soll Verantwortung für den gesamten Code tragen.
- Zuerst wird der Unit-Test erstellt!
- Der Kunde ist für die Entwickler verfügbar: schnelle Reaktion auf sich ändernde Kundenwünsche bzw. rasche Antworten auf Entwicklerfragen möglich.



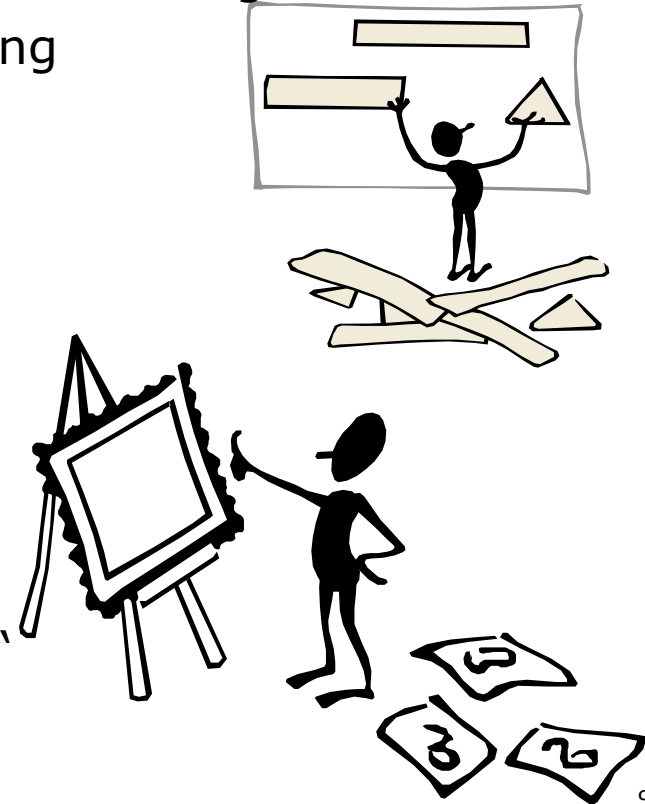
XP: Regeln

● Management:

- Messung der Projektgeschwindigkeit (User Stories, Iterationen)
- Team benötigt dedizierte Arbeitsbereiche
- Geschwindigkeit: nachhaltig, messbar und vorhersagbar sein
- Der Tag beginnt mit einem Standup-Meeting
- Jeder sollte ausreichend know-how über alle Teile des Systems besitzen
- Fix XP When It Breaks

● Design der Software:

- Einfachheit: KISS
- Systemmetapher
- CRC-Cards: Class, Responsibilities and Collaboration
- Risikominimierung durch „Spike Solutions“
- Umstrukturierung des Designs



Walter Rafeiner-Magor

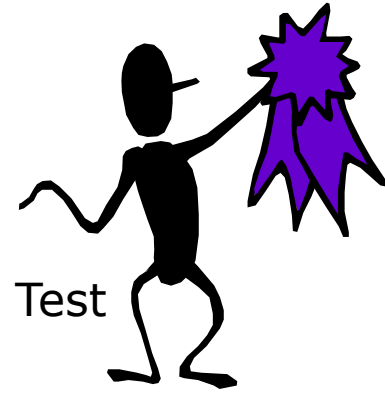
XP: Zusammenfassung

● Vorteile:

- Collective Ownership
- The Planning Game: sehr kurze Iterationen
- TDD: Codierung nur bei vorhandenen (fehlgeschlagenen) Test
- Refactoring: it must also be clean!
- Einfachheit: KISS
- Coding Standards: Alle Entwickler halten sich an gemeinsame Standards
- Whole Team: Entwickler, Business-Analysten, Projektmanager,... Team arbeitet in gemeinsamen Raum zusammen

● Agiles Manifest:

- Menschen wichtiger als Prozesse: ja
- Laufende Software wichtiger als Doku: ja
- Zusammenarbeit mit dem Kunden wichtiger als Vertragsverhandlungen: ja
- Veränderungen begrüßen statt Planverfolgung: ja!



Vielen Dank!