

eXtreme Programming in der Praxis

Dipl. Informationswirt **André Janus**

André Janus - IT-Consulting, Karlsruhe

Java User Group Karlsruhe

20. Oktober 2010



JUG Karlsruhe

jug-ka.de | twitter.com/@jugka



Inhalt

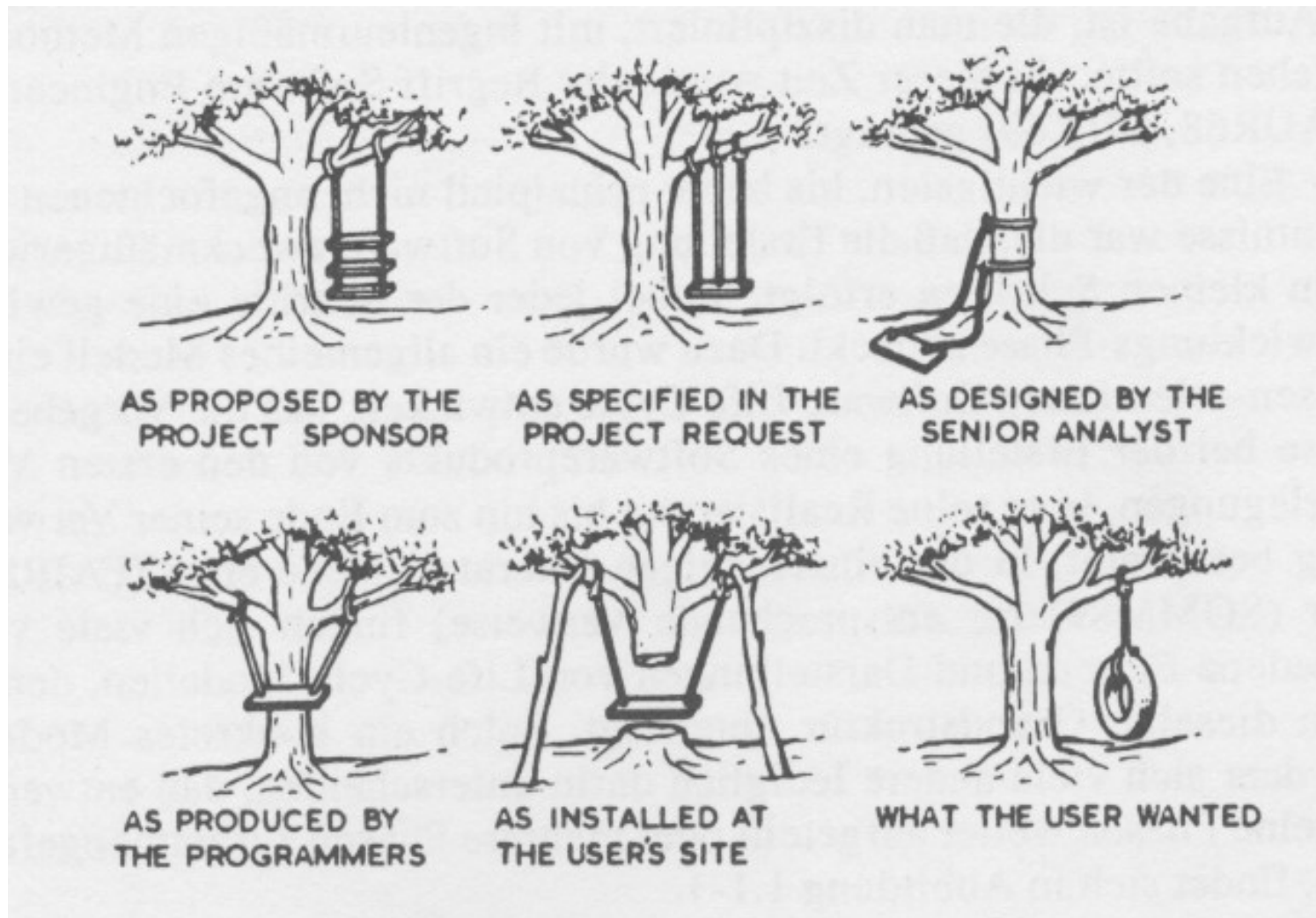
- Traditionelle Softwareentwicklungsprozesse
- Agile Softwareentwicklungsprozesse
- eXtreme Programming
- Praxisbeispiel *I2-Projekt, T-Systems*



Warum Softwareentwicklungsprozesse?

- Lehre aus der Softwarekrise
- Entwicklung in größeren Teams

Warum Softwareentwicklungsprozesse?






Traditionelle (schwergewichtige) Entwicklungsprozesse

- **Schwergeichtig** = viel Detailplanung und Dokumentation
- **Beispiele**
 - Wasserfallmodell
 - V-Modell
- **Grundmodell:** festgelegte Phasen folgen aufeinander
- **Grundannahme:** Anforderungen stehen von Beginn an fest



Traditionelle (schwergewichtige) Entwicklungsprozesse

- Probleme
 - Fehler in einer Phase erfordern Neubeginn
 - Vorhandene Arbeit nicht mehr verwendbar
- Beispiel: Projektende wg. Budgetüberschreitung:
 - Software nicht lauffähig
 - Projekt komplett gescheitert
- Praxiserfahrung zeigt: Anforderungen ändern sich




Agile (leichtgewichtige) Entwicklungsprozesse

- Leichtgewichtig = ad hoc Planung, min. Dokumentation
- Beispiele
 - ☐ eXtreme Programming
 - ☐ Scrum
 - ☐ Feature Driven Development
- agil ≠
 - ☐ Planlos
 - ☐ Chaotisch
 - ☐ „Quick-and-Dirty-Hacking“

Agile (leichtgewichtige) Entwicklungsprozesse


- Leichtgewichtig = ad hoc Planung, min. Dokumentation
- Beispiele
 - eXtreme Programming






Agile (leichtgewichtige) Entwicklungsprozesse

- Grundmodell:
 - Planung in kleineren Zeiteinheiten
 - Planung in kleineren Realisierungseinheiten (inkrementell)
 - Wiederholung einzelner Phasen (iterativ)
- Grundannahme: Anforderungen ändern sich (=> aktives Risikomanagement)



Agile (leichtgewichtige) Entwicklungsprozesse

- Beispiel: Projektende wg. Budgetüberschreitung:
 - ☐ lauffähige Software
 - ☐ verminderter Funktionsumfang
 - ☐ Projekt ist nicht komplett gescheitert



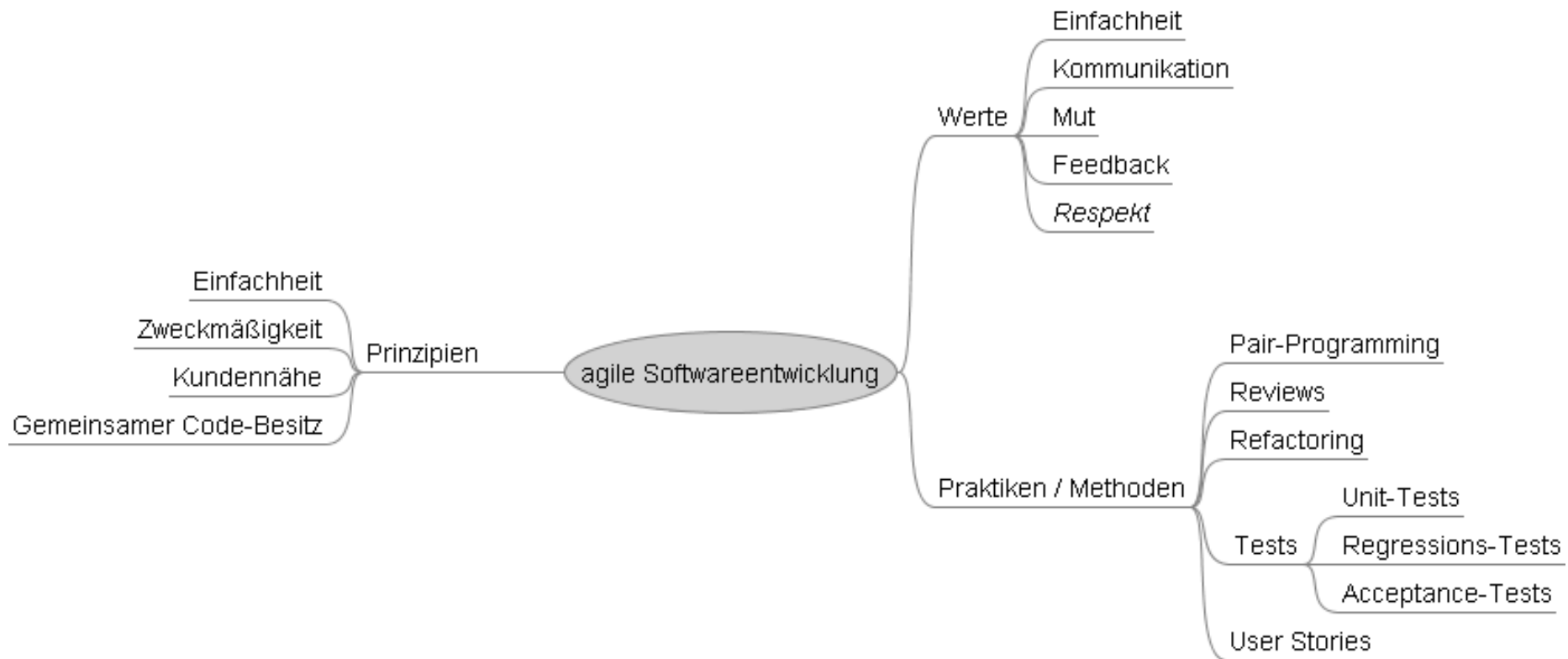
Agile (leichtgewichtige) Entwicklungsprozesse

Agiles Manifest 2001, Kent Beck et al.

Agile Werte

- Individuen und Interaktionen gelten mehr als Prozesse und Tools.
- Funktionierende Programme gelten mehr als ausführliche Dokumentation.
- Die stetige Zusammenarbeit mit dem Kunden steht über Verträgen.
- Der Mut und die Offenheit für Änderungen steht über dem Befolgen eines festgelegten Plans.

Agile (leichtgewichtige) Entwicklungsprozesse





eXtreme Programming

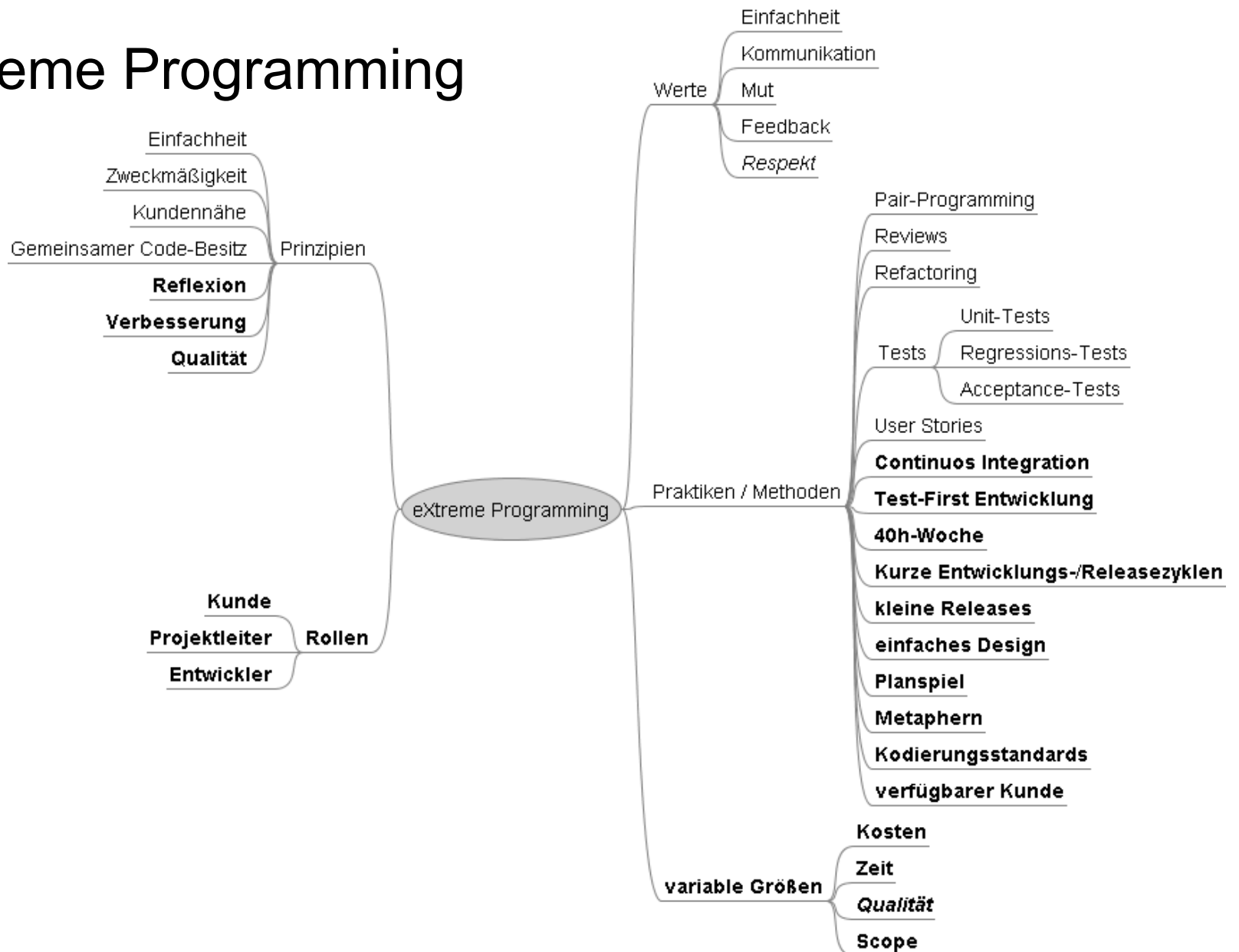
- Best Practices „ins extrem getrieben“
- zusätzliche Prinzipien
- zusätzliche Methoden
- Rollen im Projekt
- variable Größen

eXtreme Programming

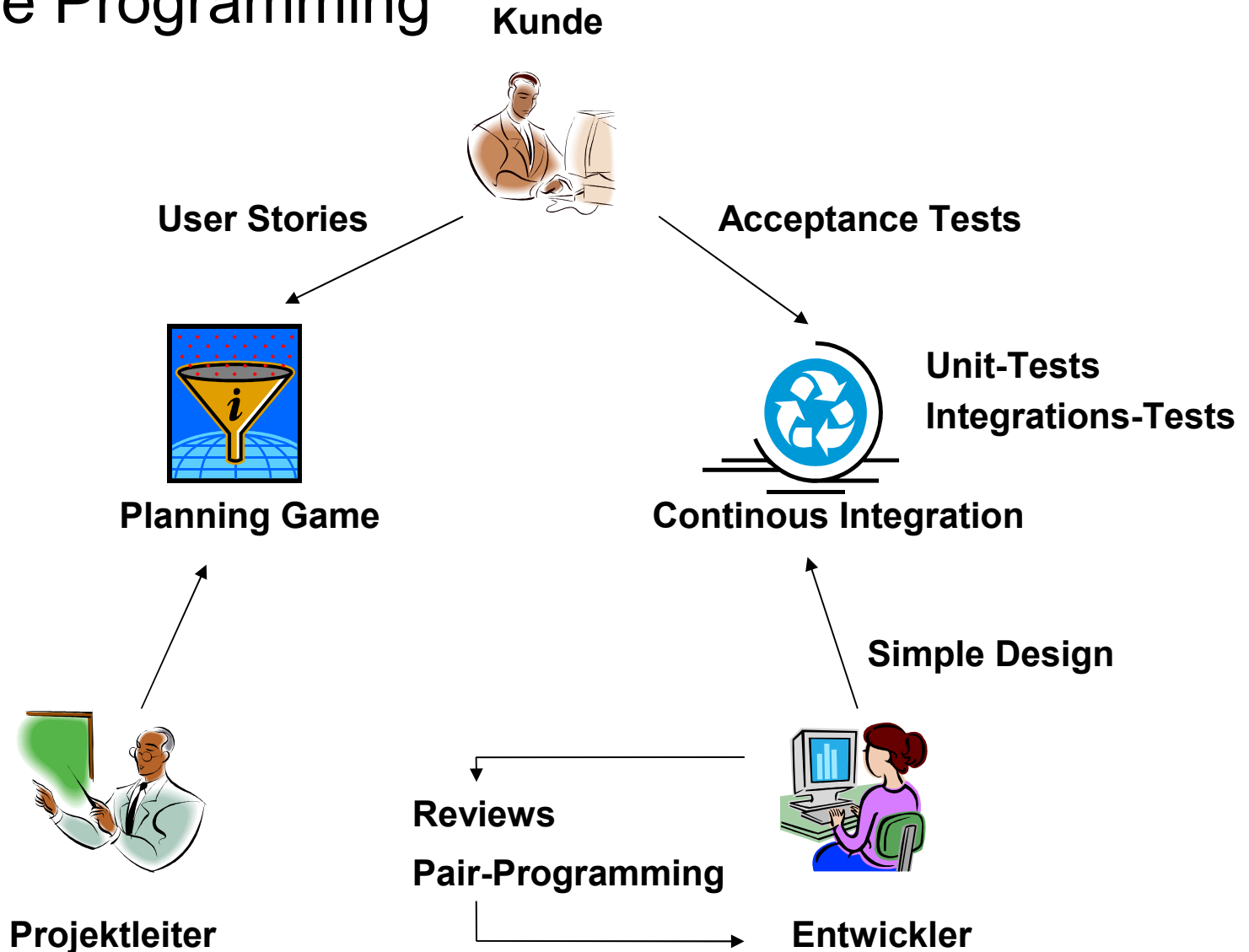
- Best Practices „ins extrem getrieben“



eXtreme Programming



eXtreme Programming



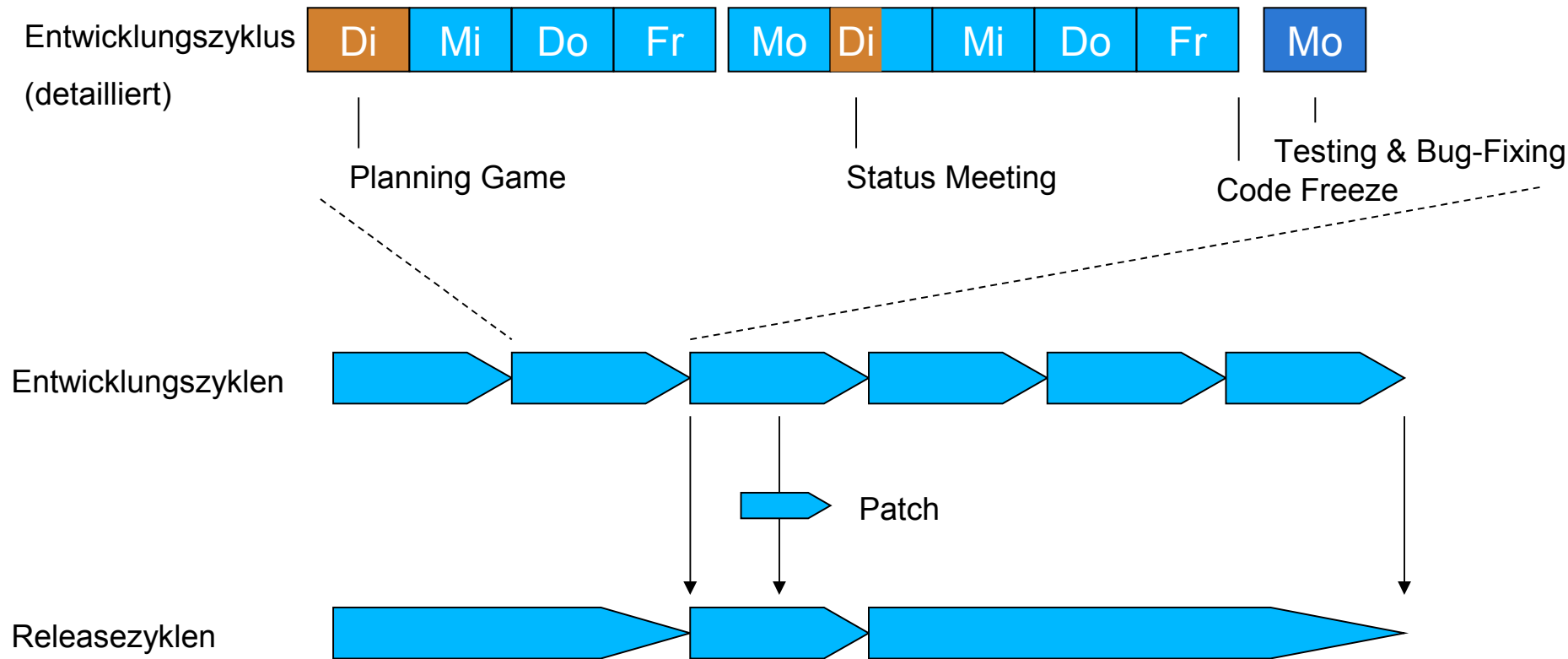
Praxisbeispiel *I2-Projekt, T-Systems*

„Variante“ von eXtreme Programming

- zweiwöchiger Entwicklungs-Zyklus
- Planning Game mit Planning Poker (Aufwandschätzung)
- Grobschätzung Hauptfunktionen vor Planning Game (~> Festpreis)
- Feste Puffer für unerwartete Aktivitäten (Bugfixing)
- User Stories = MiniSpecs (Mini-Spezifikationen)
- Verfügbarer Kunde = Fachteam (Anforderungsverdichtung)
- Feedback = Statusmeeting, Umplanungen
- Entwicklungszyklus != Releasezyklus
- Continuous Integration = Continuous Build
- Tool-Unterstützung für
 - Requirements
 - Entwickler-Aktivitäten



Praxisbeispiel *I2-Projekt, T-Systems*





eXtreme Programming

- Herausforderungen
 - ☐ Kunde ist nicht agil
 - ☐ Festpreisprojekte (keine scheinbare Sicherheit)
 - ☐ Team-Größe

eXtreme Programming

- Herausforderungen
 - Kunde ist nicht agil
 - Festpreisprojekte (keine scheinbare Sicherheit)
 - Team-Größe





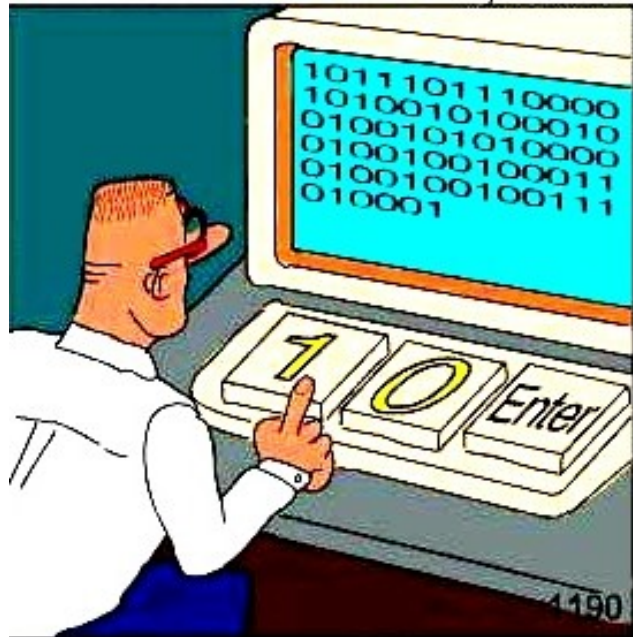
eXtreme Programming

- Erfolgsfaktoren
 - Prozess muss gelebt werden
 - Vertrauen
 - Individuelle Variante erforderlich
 - Anpassungen des Prozesses an Projekt
 - Anpassungen des Prozesses an Kunde
 - Kontinuierliche Verbesserung des „eigenen“ Prozesses

eXtreme Programming

■ eXtreme Programming =>

- „Das beste, was man für den Preis bekommen kann.“
- „Am meisten Software fürs Geld“



REAL Programmers code in BINARY.