

Extreme Programming - XP

Vortrag mit anschließendem Gedankenaustausch

Carsten Siedentop
GFU Cyrus + Rölke mbH

Diese Folien sind begleitendes Material für einen ca 45-minütigen Vortrag im Rahmen der Vortragsreihe "Semicolon" der GFU Cyrus + Rölke mbH, Köln. Für ein Selbststudium sind sie nicht konzipiert und daher aus sich selbst heraus auch nicht aussagekräftig genug.

Geschützte Namen und Begriffe werden anerkannt aber nicht gesondert mit einem Copyright oder Trademark versehen, um den Lesefluss nicht zu stören.

Die Inhalte beziehen sich im Wesentlichen auf die Erstveröffentlichung von Kent Beck "eXtreme Programming explained" von 1999 (ISBN 0-201-61641-6, bzw. 3-8273-1709-6).

Inhalt

- Ausgangslage, allgemeine Probleme
- Prinzipien von XP
- XP in der Praxis
- Planung von XP-Projekten
- Erfahrungsberichte
- Schwierigkeiten in der Praxis
- Gegenseitiger Gedankenaustausch

Seit Jahren die gleichen Projektziele

- Kosten klein halten
- Entwicklungszeit kurz halten
- Hohe Qualität erreichen
- Hohen Funktionsumfang erzielen

- Jedes der 4 Ziele kann nur in Abhängigkeit der anderen optimiert werden.

Seit Jahren die gleichen Probleme

- Terminverzögerungen, Budget-Überschreitung
- Hohe Fehlerrate
- Unrentables System
- Geforderte Funktionalität fehlt
- Nicht geforderte Funktionalität wurde implementiert
- Das Geschäftsziel ändert sich
- Personalwechsel

Im Jahre 1999 veröffentlichte **Kent Beck** Erkenntnisse aus einem "anders" geführten Projekt (C3 bei Chrysler). Beteiligt waren

**Ward Cunningham,
Martin Fowler,
Erich Gamma,
und Ron Jeffries.**

Das Buch hieß "eXtreme Programming explained" mit dem Untertitel "embrace change".

Prinzipien von XP

- Was allgemein als "gut" betrachtet wird, wird extrem gelebt. Schnelles Feedback
- Testen, Automatisierung von Tests
- 4-Augen-Prinzip
- Inkrementelle Implementierung
- Inkrementelle Planung
- Evolutionäres Design

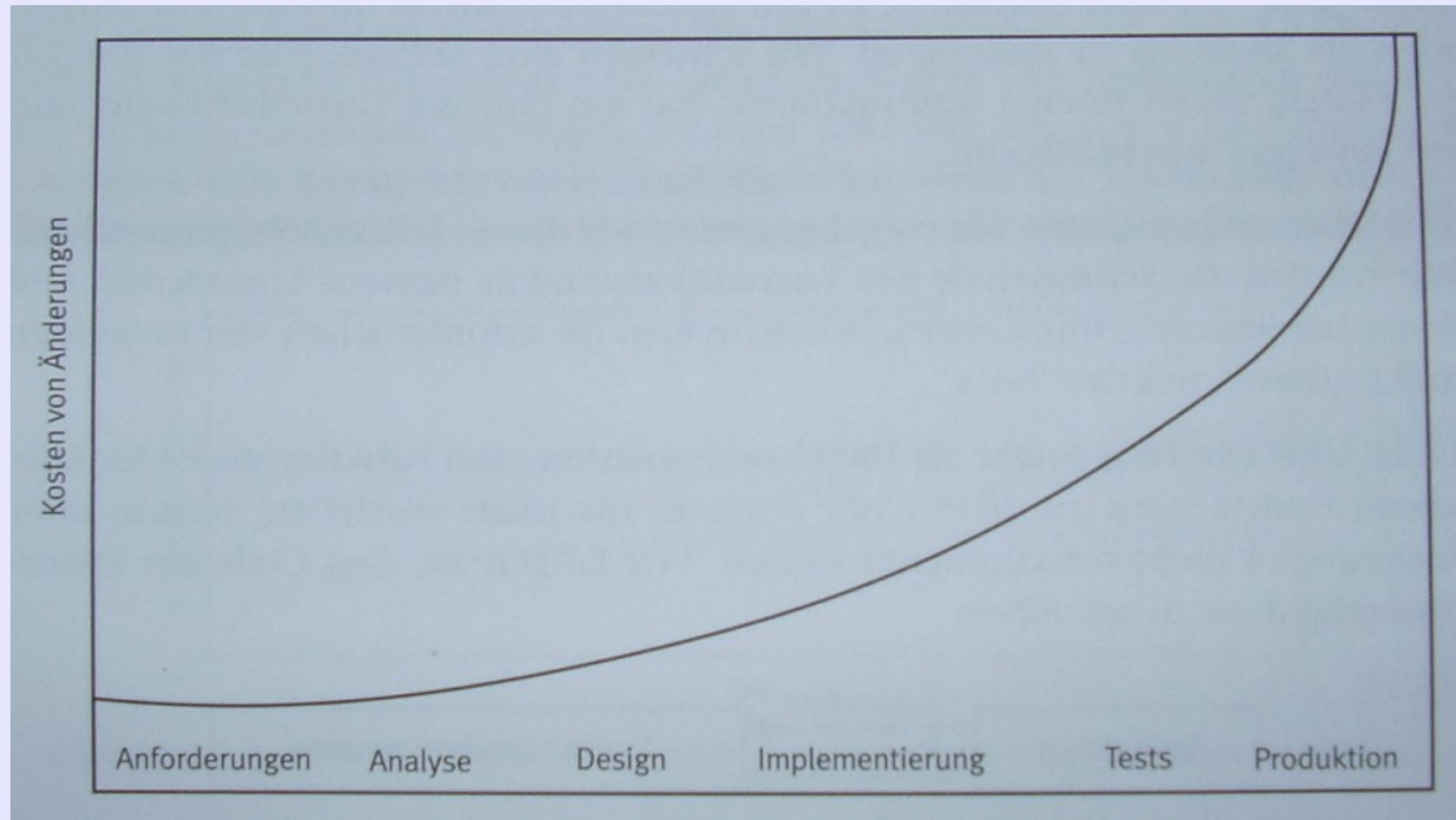
Zusammenhang von Prinzipien + Zielen

- Automatisierung von Tests
→ steigert Qualität, beschleunigt Fehlererkennung
- 4-Augen-Prinzip
→ verteilt Verantwortung auf viele Schultern
- Schnelles Feedback
→ steigert Qualität, vermeidet Fehlentwicklung
- Inkrementelle Implementierung, evolutionäres Design
→ beschleunigt das Erkennen von Fehlentwicklung
→ verringert Aufwände für Integrationstests
→ bringt Nutzen schneller in Produktion

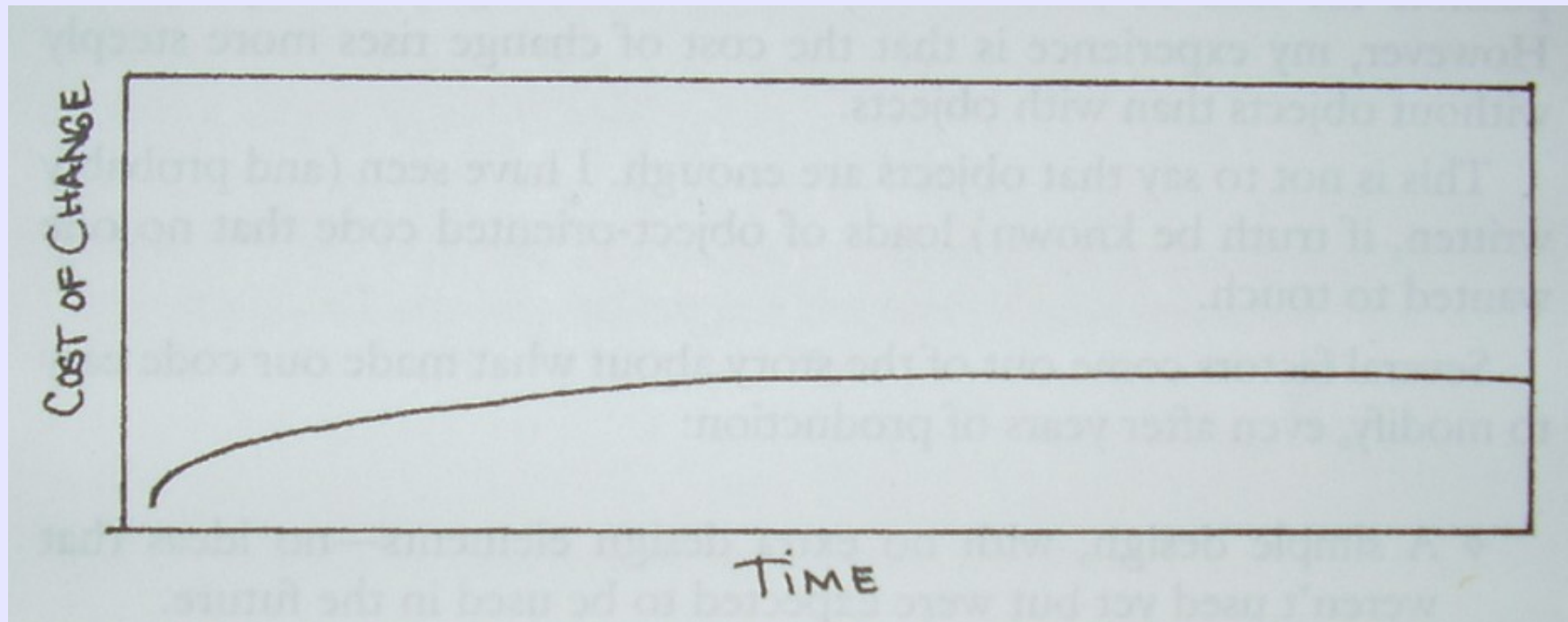
Das Extreme an XP

- Automatisierung von Tests
→ Continuous Integration, Nightly Build
- 4-Augen-Prinzip
→ Entwickler arbeiten nur noch zu zweit
- Schnelles Feedback
→ tägliche Kommunikation zum Fachbereich
- Inkrementelle Implementierung, evolutionäres Design
→ Kurze Iterationszyklen (6 Monate für Release 1.0)
(1 Monat nachfolgende Release)
→ Kleine Tasks (Entwicklungszeit: 1 Tag)

Die Kostenkurve ohne XP



Die Kostenkurve mit XP



Zitate zur Kostenkurve mit XP

- "Es wird allerdings nicht ganz einfach sein, einen solchen Ansatz zu entwickeln."
- "...einfaches Design ohne zusätzliche Designelemente..."
- "...eine Menge Übung im Ändern des Designs... "
- "Automatisierte Tests, die uns das Vertrauen geben, wir würden es sofort bemerken, wenn wir unbeabsichtigt das Verhalten des vorhandenen Systems ändern würden."
- "In einem System ohne Werte tendieren Menschen dazu, sich nur um ihre eigenen, kurzfristigen Interessen zu kümmern."

Die Werte von XP

- Kommunikation
 - Mangelhafte Kommunikation ist kein Zufall
- Einfachheit
 - Wie sieht die einfachste Lösung aus?
- Feedback
 - Optimismus ist eine Berufskrankheit, die nur durch Feedback zu behandeln ist.
- Mut
 - Fehlerhafter Code soll weggeworfen werden
 - Codieren sie eine kleine Zeit parallele Varianten

XP in der Praxis

- Code gehört allen
- Tägliches Integrieren funktioniert gut!
- Pair-Programming ist effizient
- Pair-Programming führt zu Wissens-Transfer
- Die Kosten für den Aufbau einer Continuous-Integration Umgebung sind gering
- Release 1 eines neuen Projektes sollte ca. 6 Monate dauern (9 Monate)
- Folge-Releases sollten 1 Monat dauern. (3 Monate)

Code gehört allen

- Schlechter Code hält sich nicht lange.
- Gemeinsame Verantwortlichkeit stärkt das Gefühl persönlicher Einflussnahme auf das Projekt.
- Wissen wird über das Team verteilt

Pair-Programming

- Es entsteht von Anfang an besserer Code
- Einer tippt – zwei Denken.
- Die Kommunikation während des Programmierens bringt den Entwicklungsprozess voran.
- Ein Rückfall in schlechte Gewohnheiten unter Stress findet seltener statt.
- Es findet ein fachlicher und technischer Wissenstransfer statt.

XP aus der Planungssicht (I)

Die Planung dient den Zwecken,

- ein Team zusammenzustellen
- über Umfang und Prioritäten zu entscheiden
- einen Fixpunkt für ein regelmäßiges Feedback zur Verfügung zu stellen
- das Vertrauen der Projektmitarbeiter zu stärken.
(Jeder Mitarbeiter muss Verantwortung übernehmen.)

XP aus der Planungssicht (II)

- Zu Beginn einer Iteration (1 – 4 Wochen) wählt die Fachseite die wichtigsten Feature aus.
- Die Entwickler schätzen den Aufwand
- Die Entwickler können Aufwandsaussagen korrigieren.
- Nach vollendeter Implementierung gleicht der Entwickler seine Aufwandsschätzung mit der Realität ab.
- Der Planungsaufwand sollte ca. 1 Tag auf 30 Arbeitstage ausmachen. Nicht mehr.
- Die Fachseite kann jederzeit neue Funktionen definieren.

Berichte von XP-Projekten

- 1995 – 1997, Softwarearchitektur für Client/Server mit Windows-Clients und IBM-Großrechner als Server, COBOL etc., Versicherungsbranche
- 2001 – 2002, Basel-II Implementierung, Java, Oracle-DB, Bankenbranche
- 2005 – 2006, Cash-Management System, Java, Web-Frontend, Oracle-DB, Großhandel

Probleme von XP in der Praxis

- Schnelles Feedback muss zu Konsequenzen führen
- "XP ist nicht steuerbar"
- XP widerspricht den Instinkten vieler Programmierer.
 - Probleme sollen vermieden werden.
 - Wenn schon Probleme, dann nicht jetzt.
- Nicht jeder kann mit jedem zusammenarbeiten
- Zeitschätzung/Zeitmessung findet selten statt
- Unit-Tests werden unter Zeitdruck vernachlässigt
- Die Räumlichkeiten passen nicht

Call to Action

Überlegen Sie sich, ob das nächste Projekt mit "Extreme Programming" schlechter laufen würde, als mit Ihrem bisherigen Vorgehen.

Wenn nicht, dann sollten Sie den Versuch wagen.

Herzlichen Dank

Fragen, Kritik, Anregungen?