Extrem Programming

Gründe, Methoden und Prozesse

Nick Zbinden

25. Januar 2011

Extrem Programming?


Antec

Extrem Programming ist der Einzug von Wahrheit und Vernunft in die Software Entwicklung.

Pavel Maier

Probleme


I'LL GO UP AND FIND OUT WHAT THEY NEED AND THE REST OF YOU START CODING !

Probleme

- Software wird nicht rechtzeitig fertig.
- Entwicklung wird fürhzeitig beendet.

Probleme

- Software wird nicht rechtzeitig fertig.
- Entwicklung wird fürhzeitig beendet.
- Änderungsaufwand wird zu gross.

Probleme

- Software wird nicht rechtzeitig fertig.
- Entwicklung wird fürhzeitig beendet.
- Änderungsaufwand wird zu gross.
- Software wird wegen Unzuverlässigkeit nicht genutzt

Probleme

- Software wird nicht rechtzeitig fertig.
- Entwicklung wird fürhzeitig beendet.
- Änderungsaufwand wird zu gross.
- Software wird wegen Unzuverlässigkeit nicht genutzt
- Kunde weiss nicht was er will!

Probleme

- Software wird nicht rechtzeitig fertig.
- Entwicklung wird fürhzeitig beendet.
- Änderungsaufwand wird zu gross.
- Software wird wegen Unzuverlässigkeit nicht genutzt
- Kunde weiss nicht was er will!
- Kunde ändert seine Meinung.

Probleme

- Software wird nicht rechtzeitig fertig.
- Entwicklung wird fürhzeitig beendet.
- Änderungsaufwand wird zu gross.
- Software wird wegen Unzuverlässigkeit nicht genutzt
- Kunde weiss nicht was er will!
- Kunde ändert seine Meinung.

Ängste

- Kunde
  - Kriegt nicht was er will

Ängste

- Kunde
  - Kriegt nicht was er will
  - Zahlt zu viel

Ängste

- Kunde
  - Kriegt nicht was er will
  - Zahlt zu viel
  - Pläne sind Märchen.

Ängste

- Kunde
  - Kriegt nicht was er will
  - Zahlt zu viel
  - Pläne sind Märchen.
  - Kann Anforderungen nicht ändern.

Ängste

- Kunde
  - Kriegt nicht was er will
  - Zahlt zu viel
  - Pläne sind Märchen.
  - Kann Anforderungen nicht ändern.

Ängste

- Entwickler
  - Kunde verlangt zu viel.

Ängste

- Entwickler
  - Kunde verlangt zu viel.
  - Verantwortung aber keine Autorität.

Ängste

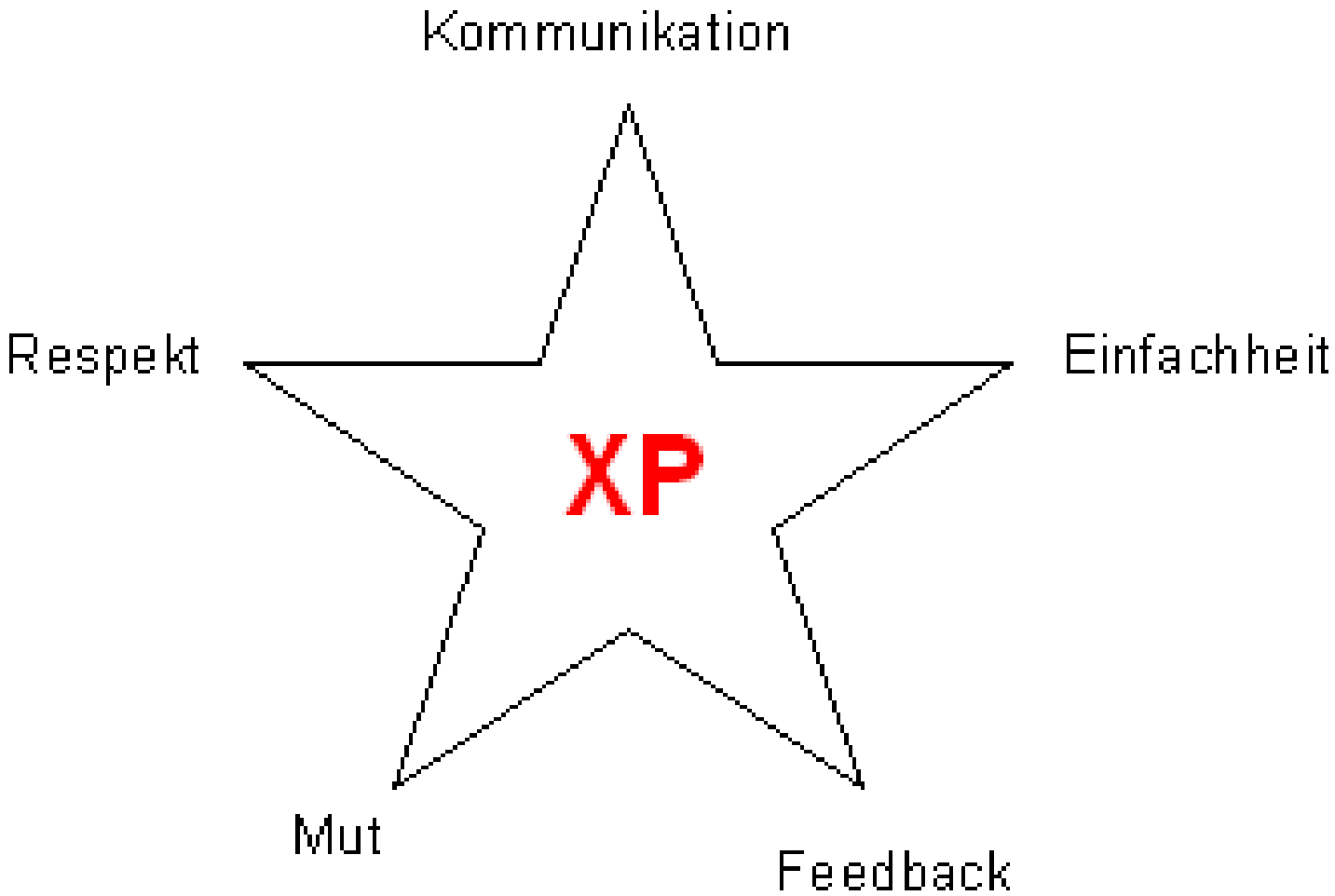
- Entwickler
  - Kunde verlangt zu viel.
  - Verantwortung aber keine Autorität.
  - Qualität Opfern für Termine.

Ängste

- Entwickler
  - Kunde verlangt zu viel.
  - Verantwortung aber keine Autorität.
  - Qualität Opfern für Termine.

Ängste

- Entwickler
  - Kunde verlangt zu viel.
  - Verantwortung aber keine Autorität.
  - Qualität Opfern für Termine.


Kommunikation
Respekt
Einfachheit
XP
Mut
Feedback

Best Practises

- Best Practieses
  - Keine Code-Besitzer

Best Practises

- Best Practieses
  - Keine Code-Besitzer
  - Refactoring

Best Practises

- Best Practieses
  - Keine Code-Besitzer
  - Refactoring
  - Keine Überstunden

Best Practises

- Best Practieses
  - Keine Code-Besitzer
  - Refactoring
  - Keine Überstunden
  - Small releases

Best Practises

- Best Practieses
  - Keine Code-Besitzer
  - Refactoring
  - Keine Überstunden
  - Small releases

Pair Programming

Pair Programming

- Keine Know How Monople (Bus-Faktor)
- Kürzere Einareitungszeiten

Pair Programming

- Keine Know How Monople (Bus-Faktor)
- Kürzere Einareitungszeiten
- Weniger Ablenkung beim Programmieren

Pair Programming

- Keine Know How Monople (Bus-Faktor)
- Kürzere Einareitungszeiten
- Weniger Ablenkung beim Programmieren
- *Besserer Code*

Pair Programming

- Keine Know How Monople (Bus-Faktor)
- Kürzere Einareitungszeiten
- Weniger Ablenkung beim Programmieren
- *Besserer Code*
- Studien

Pair Programming

- Keine Know How Monople (Bus-Faktor)
- Kürzere Einareitungszeiten
- Weniger Ablenkung beim Programmieren
- *Besserer Code*
- Studien

Test Driven Development

- Jede Funktion wird getestet.
- Test wird zuerst geschrieben.

Test Driven Development

- Jede Funktion wird getestet.
- Test wird zuerst geschrieben.
- Beim Einchecken wird alles getestet

Test Driven Development

- Jede Funktion wird getestet.
- Test wird zuerst geschrieben.
- Beim Einchecken wird alles getestet
- Kompleze System einfach ändern

Test Driven Development

- Jede Funktion wird getestet.
- Test wird zuerst geschrieben.
- Beim Einchecken wird alles getestet
- Kompleze System einfach ändern
- Performence tests, Securety tests, Scalierungs tests

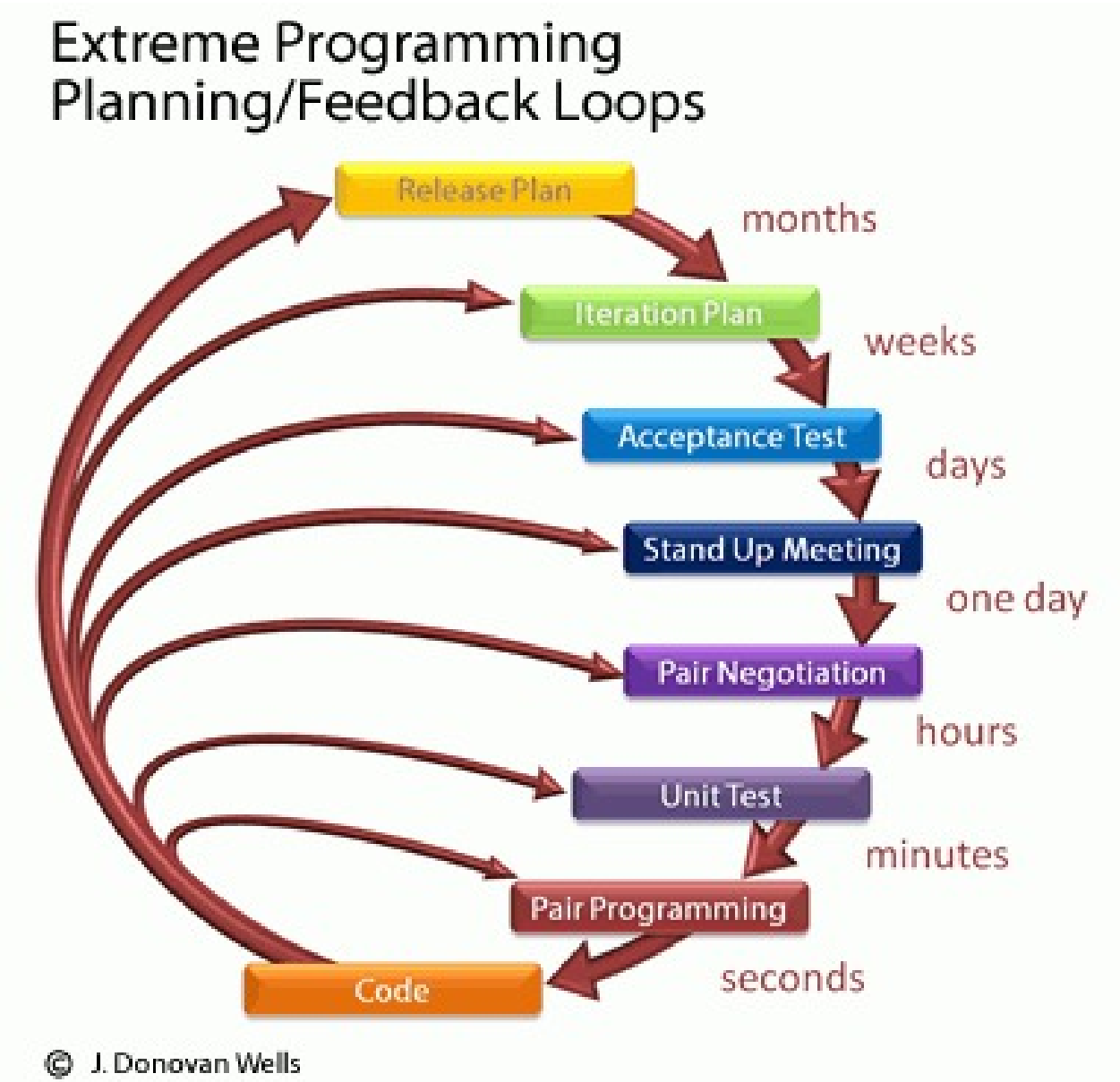
Test Driven Development

- Jede Funktion wird getestet.
- Test wird zuerst geschrieben.
- Beim Einchecken wird alles getestet
- Kompleze System einfach ändern
- Performence tests, Securety tests, Scalierungs tests

Rollen

Rollen	Aufgaben
Kunde	Entscheidet, was gemacht wird, gibt Rückmeldung
Entwickler	Entwickelt das Produkt
Projektmanager	Führung des Teams
Benutzer	Wird das zu erstellende Produkt nutzen

Prozesse


Extreme Programming
Planning/Feedback Loops
Release Plan
months
Iteration Plan
weeks
Acceptance Test
days
Stand Up Meeting
one day
Pair Negotiation
hours
Unit Test
minutes
Pair Programming
seconds
Code
© J. Donovan Wells

Release Plan

- alle 2-3 Monate
- User Storys
  - Karten erstellen
  - Aufwand
  - Risiko
  - usw.

User Storys

Story Text	First estimate	Second estimate	Random	MC value	Effort
As a player, I can start a new game.	1	2	0.355887328	1	1
As a player, I can restore a saved game.	1	3	0.676350279	3	6
As a player, I can select the computer's playing strength.	3	4	0.023389388	3	7
As a player, I can play against a weak engine that recognizes rings.	5	3	0.495721631	5	18
As a player, I can play against a weak engine that recognizes bridges.	1	2	0.887072368	2	5
As a player, I can play against a weak engine that recognizes forks.	3	2	0.799498425	2	4
As a player, I can play against a medium-strength engine.	2	2	0.438575689	2	5
As a player, I can play against a strong engine.	3	2	0.641099939	2	5
As a new player, I want access to an online help system.	2	1	0.382573772	2	4
As a player, I want the computer to recognize a winning shape.	1	4	0.439142889	1	1
As a player, I want a nice looking splash screen when the program starts.	2	3	0.316511457	2	4
As a player, I want nice looking background art that integrates with the game boards.	3	2	0.667404818	2	4
As a player, I'd like to add annotations to saved games.	6	7	0.614763336	7	41
As a player, I'd like to ask for a hint sometimes.	7	5	0.324489743	7	40
As a new player, I want to be able to view an interactive tutorial for the game.	7	5	0.33931133	7	51

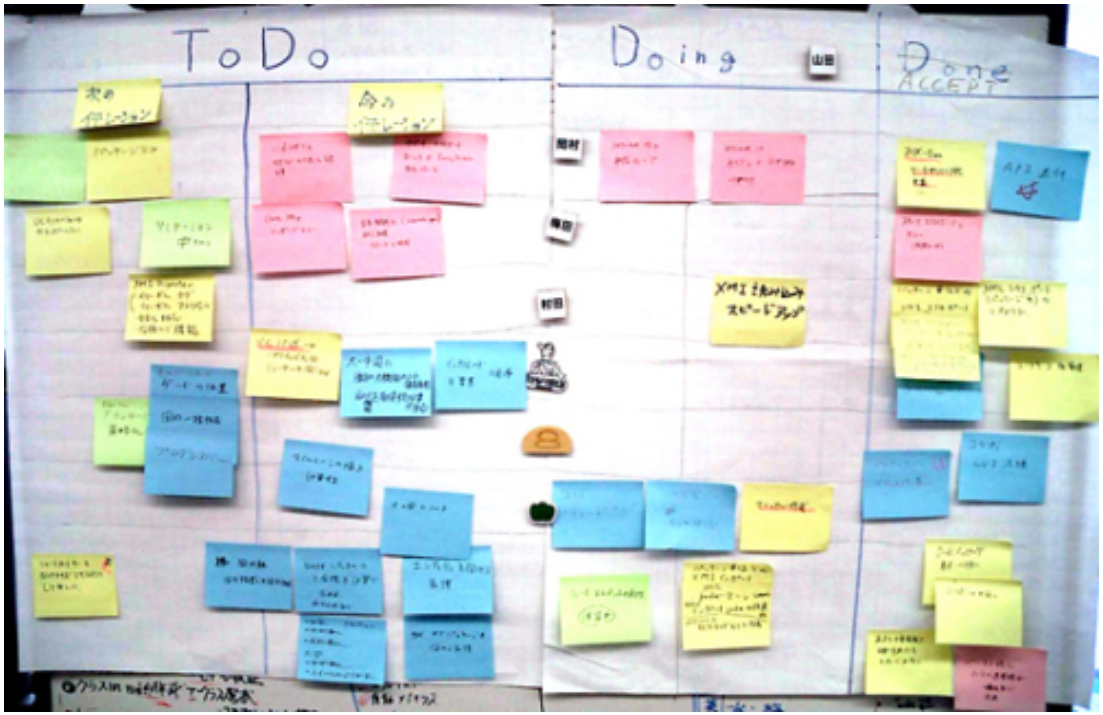
Table 2 - User stories with estimations, MC values, and effort in hours

User Storys

Iterations Plan

- alle 1-3 Wochen
- Es wird bestimmt welech Users Storys gemacht werden
- User Storys werden in Engineerings Tasks unterteilt.
  - Technische Anweisung
- Kunde kann jederzeit Storys anpassen, neue erstellen oder wegwerfen.
- Statistiken über letzte Iteration. Velocity!

Engineering Tasks

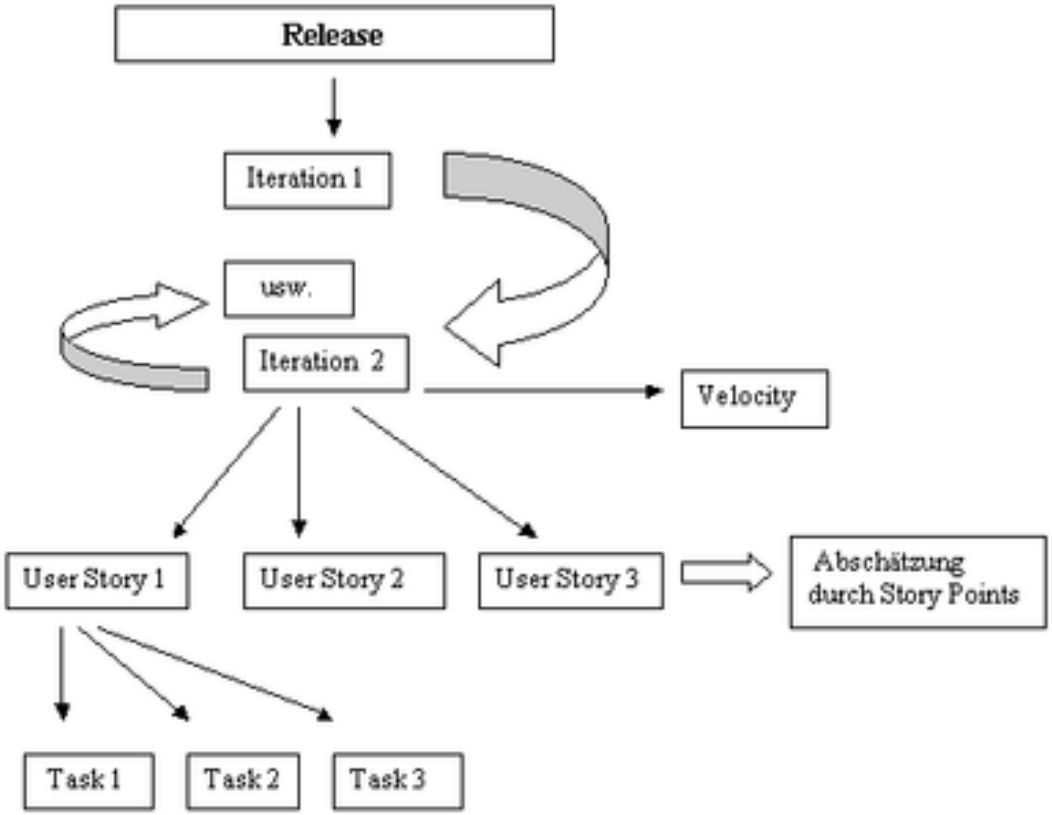


Standup-Meeting

- Jeden Morgen
- Erzählen was man gemacht hat. Probleme!
- Wie lange hat man noch an seinem Task.
- Engineerings Tasks werden verteilt.
  - Jeder Schätzt wie lange er dafür braucht.

Standup Meeting

Gesamtübersicht



The End