

バーンダウン・チャートの紹介 ～Don't burn out the project～

(株)永和システムマネジメント
角谷信太郎

XPと進捗

- ペアで作業中はずいぶん、目の前のタスクに集中しがち
- いま、プロジェクトはどこへ向かっているのか？
 - ひとめで把握したい
- 進捗会議？
 - システムは会議室でつくられるんじゃない
- 2:8の法則
 - 80%は2日で終わったが、残り20%に8日かかった!

そこで、バーンダウン・チャートの出番

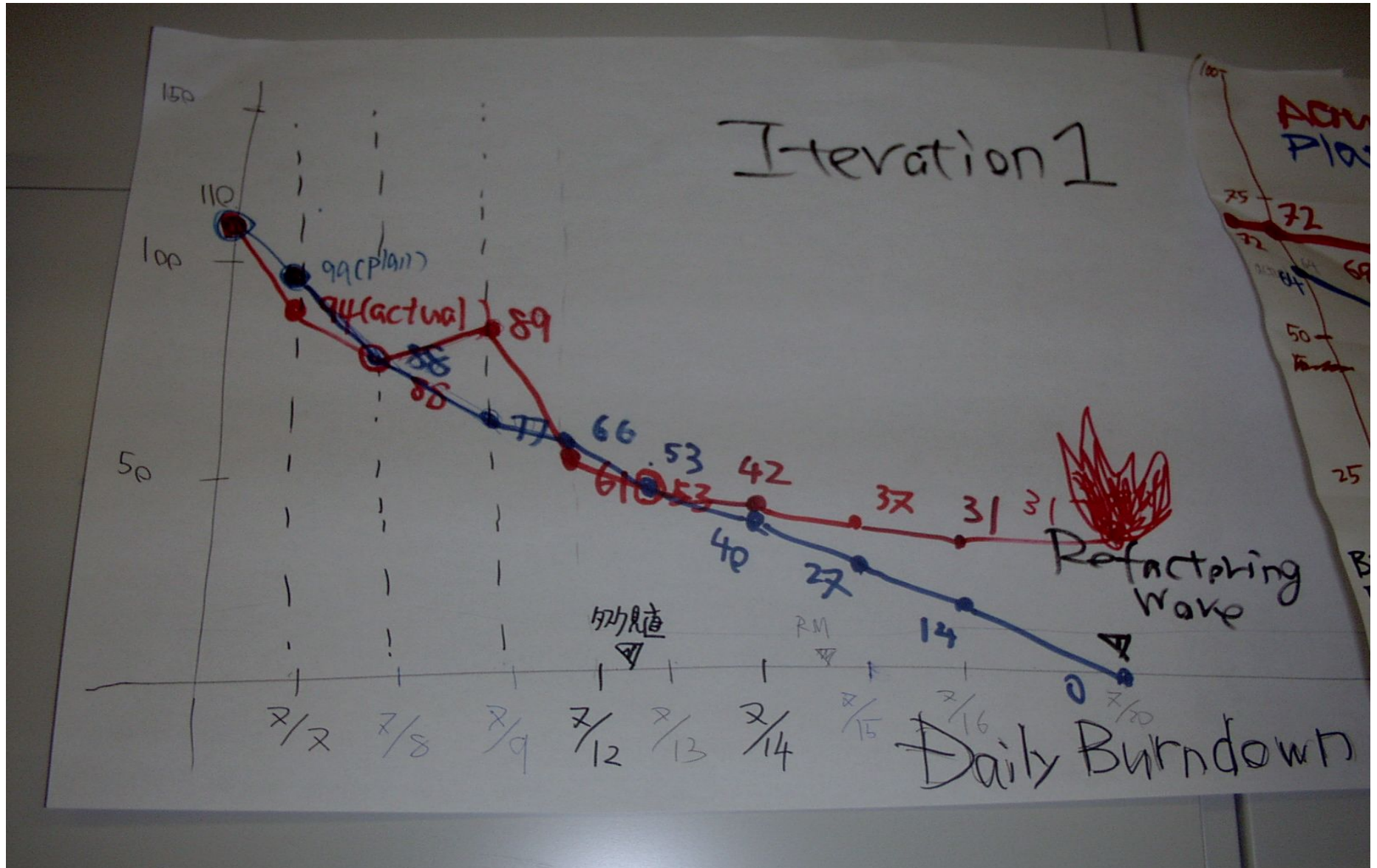
プロジェクトの紹介

- 某社向けワークフロー風フレームワーク構築(J2EE)
- プロセスはXP(ほぼフルスタック)
- 3.5名のプログラマ(XPer:1名)
 - 角谷はマネジャ、トラッカ、プログラマ(0.5)
- 1イテレーション/2週間
- 現在は2イテレーション目の中間。
 - このイテレーションの後、初回リリース

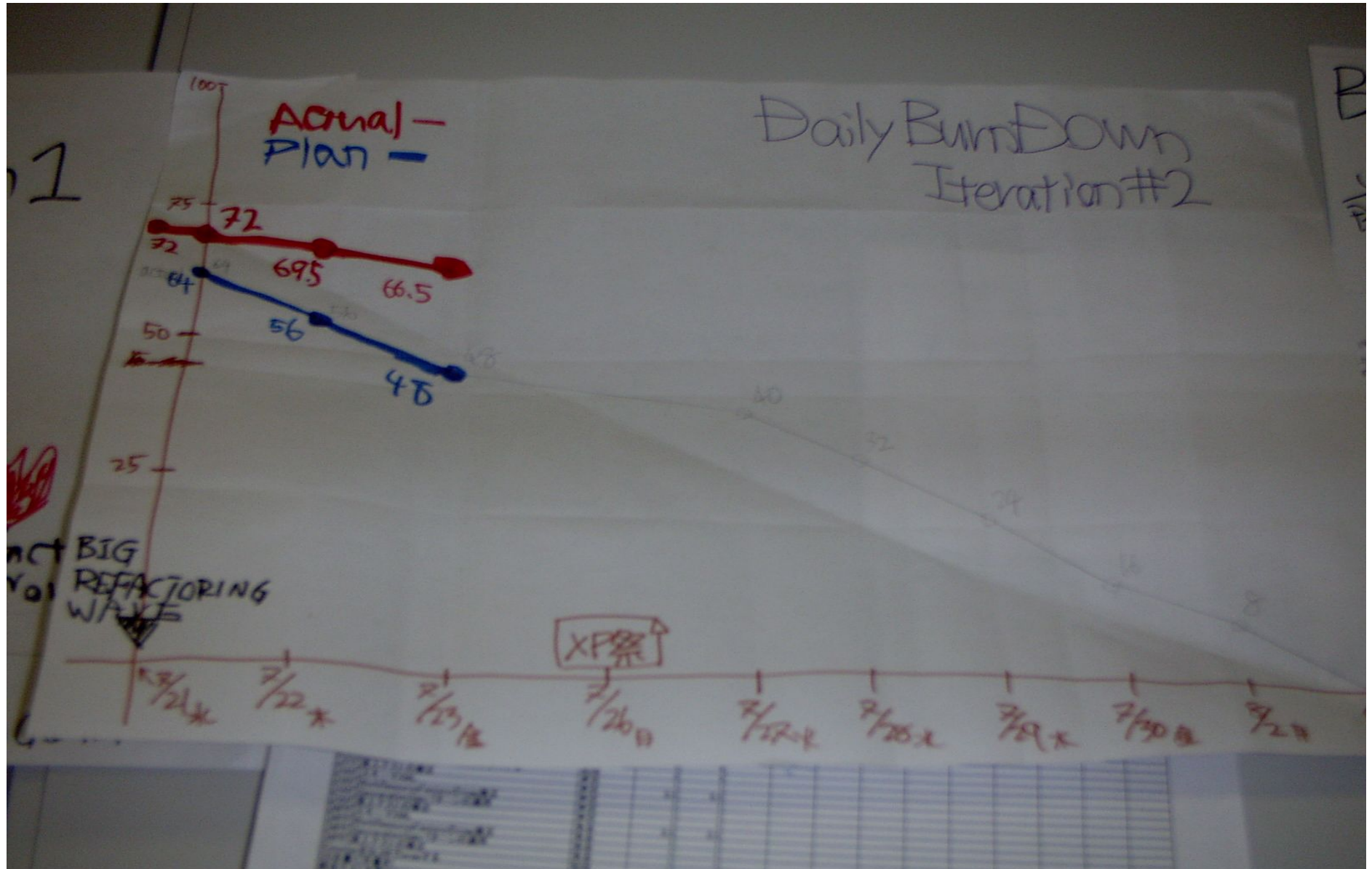
バーンダウンチャート

- バーンダウン(burndown)-> 全焼、鎮火(?)
 - ひとめでわかる進捗把握法
 - 概念は『リーンソフトウェア開発』で知る
 - ↑ 追記:本書でもバーンダウンチャートでした
 - 名前は『User Stories Applied』からイタダキ
 - すべてのプロジェクトは本質的に「火事」である
- 横軸に時間、縦軸に残タスク量をプロット
 - 予想線(青)と実績線(赤。火のメタファ)。
 - 上から下へ。キーワードは収束(**convergence**)
 - 日単位(残タスク), イテレーション単位(残ストーリー)
 - 手書きしたものを壁に貼る

実例: イテレーション#1

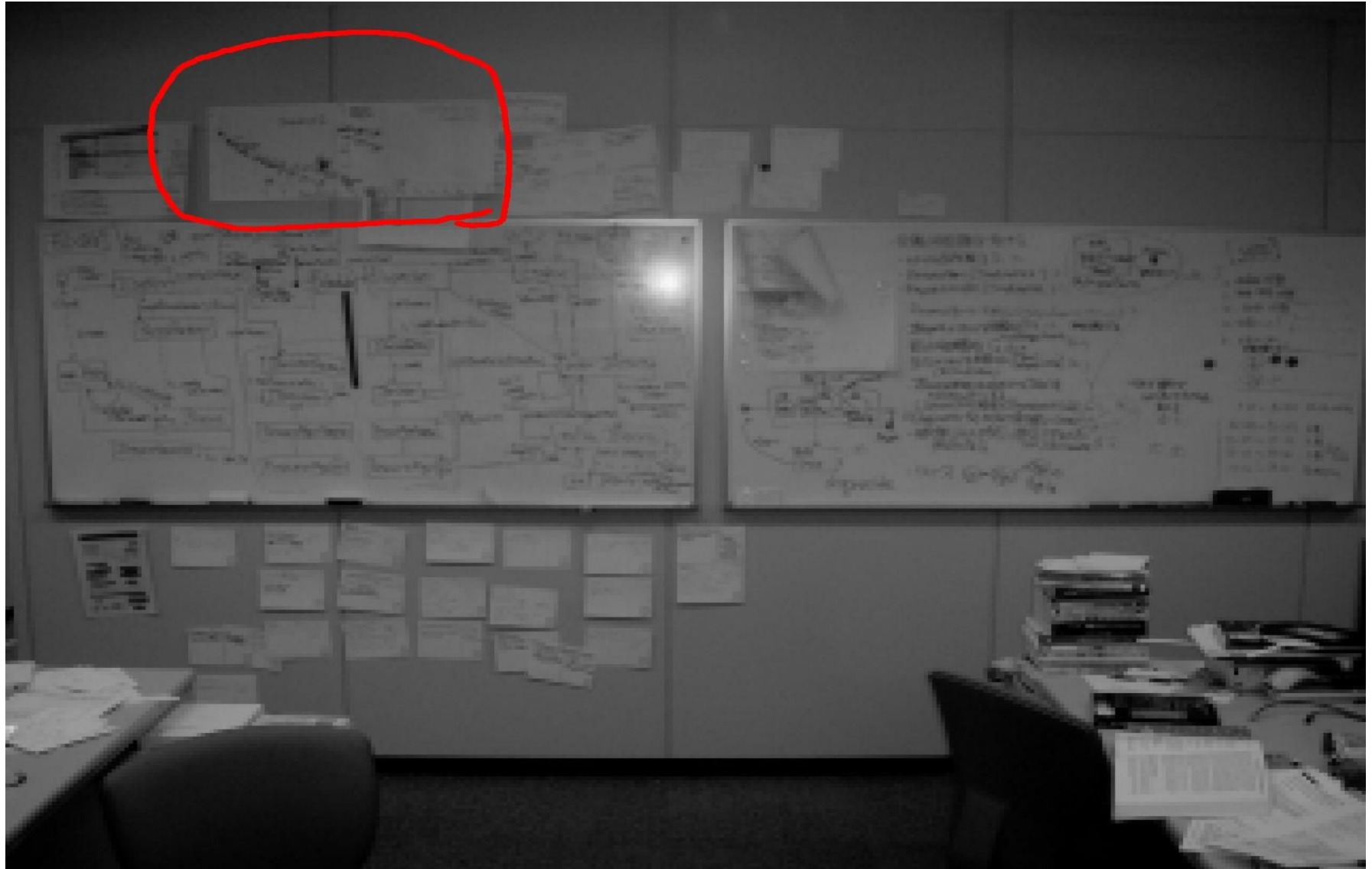


実例: イテレーション#2



Paint it Black!

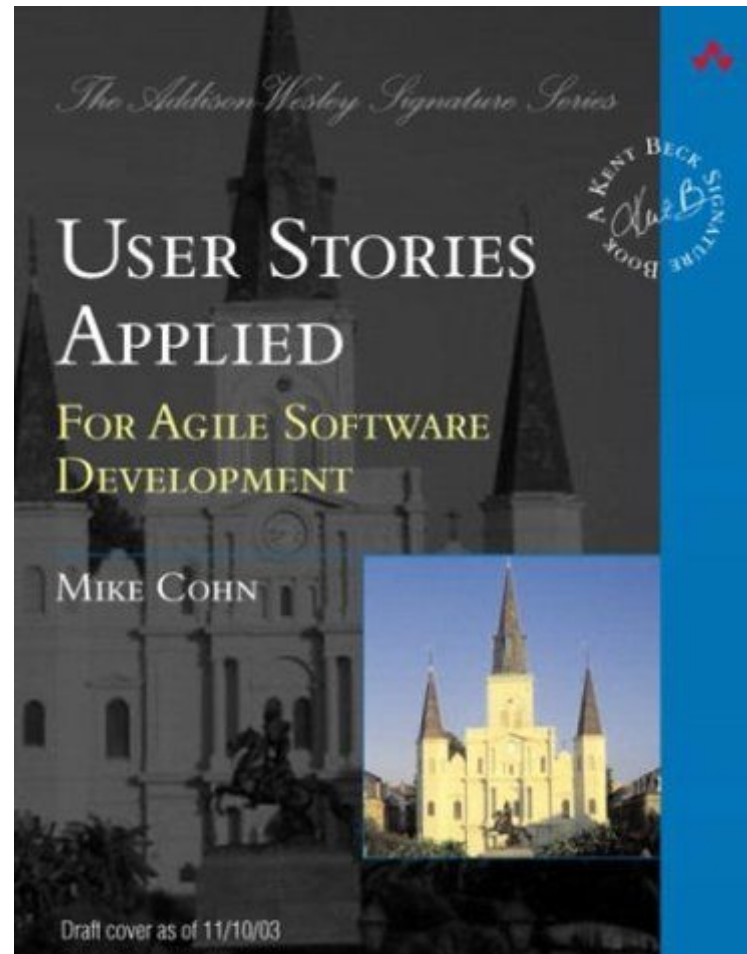
(なんでも壁に貼っちゃえ!)



雑感

- 初期イテレーションはVelocityがブレる
 - 「3, 4イテレーション回すまでトレンドを見ちゃダメ」(『User Stories Applied』)
- 適正なサイズへのタスク分割が難しい
- タスク終了条件の見極めも難しいですね
- 実行環境・制約となる外部ライブラリの落とし穴
 - 過信禁物。ちゃんとスパイクしよう！
 - まだハマってます

『User Stories Applied』



『リーンソフトウェア開発』



——失礼。プログラミングに戻らないと

