

Agile - Lean - Extreme

Mark Windholtz





Intro



- Mark Windholtz
- Are you new to Agile ?
- Why Agile-Lean-Extreme
- What is Agile-Lean-Extreme



Extreme Programming - C3 project





Standard Industry Practice

- Most Software development is
 - chaotic
 - or bureaucratic
- “Code and Fix”; or wait while doing *other* tasks.
 - Long Integration Phase
 - Long Test Phase
 - High Defect Counts



Solution of the 1990's



- Rigorous Heavy Process
- Lots of Documents
- Lots of Control Points
- Blue Prints before you build

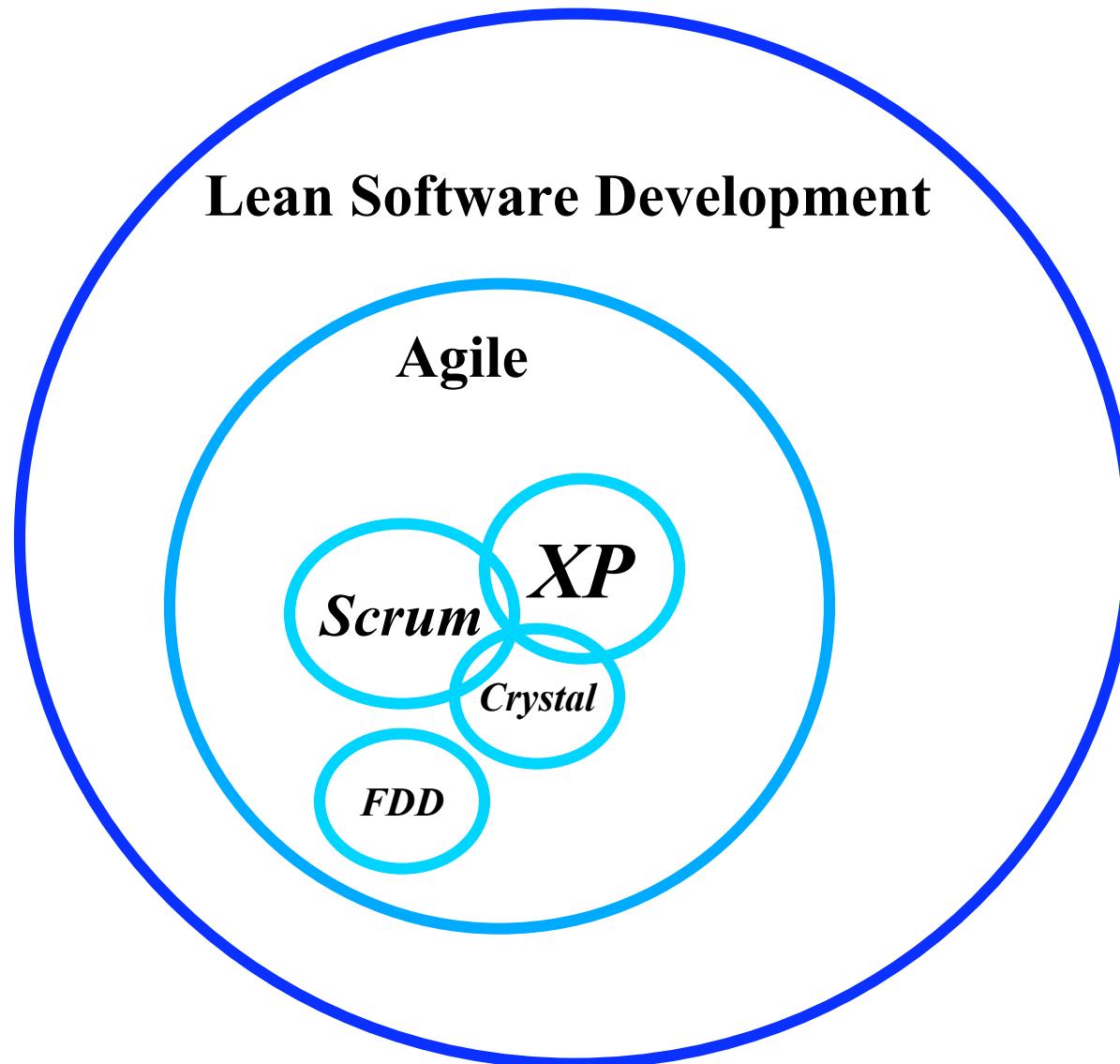


Why not common practice?

- Not Very Successful
- Conflicting Goals



Landscape



- Business Model
 - Lean
- Approach
 - Agile
- Technique
 - XP
 - Scrum



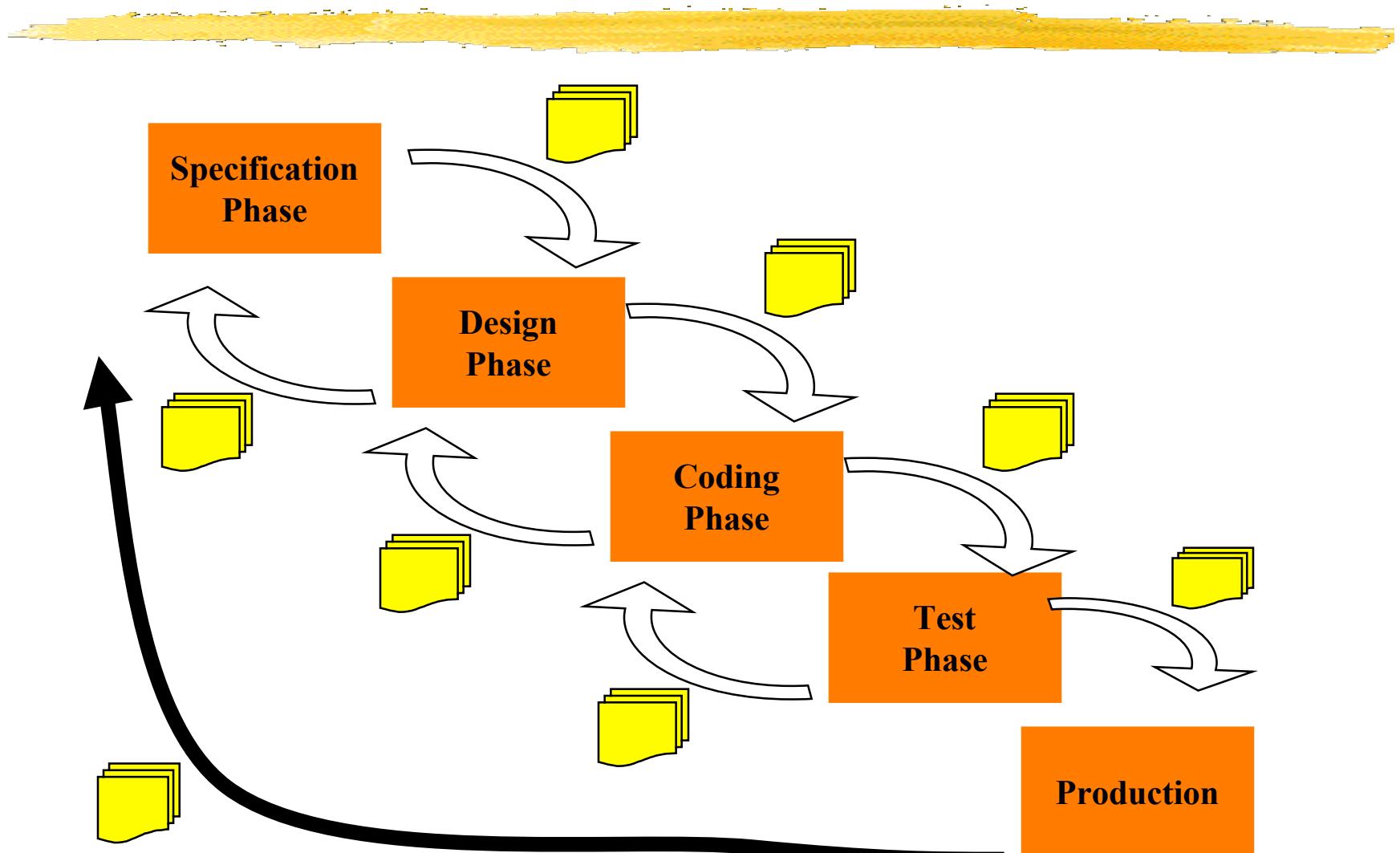
Lean: Inventory is Waste

- Inventory is a non-producing Asset
- Inventory reduces responsiveness
- Any Artifact not producing Current Income
- Features under development are Inventory
 - ✓ We don't know **if** it will work
 - until it **does** work (Risk)
 - ✓ High Administrative costs
 - ✓ Higher (Longer) Return on Investment
 - Consider: Net Present Value
 - ✓ Time between Idea and Market feedback





Lean: Phases generate Inventory





Lean: 7 Wastes of Software development

-
1. Inventory
 - Software In Process (SIP)
 2. Defects
 3. Over-production (programming *just in case*)
 4. Extra Process Steps
 5. Motion getting the needed information
 6. Waiting for information
 7. Transportation - handoffs



Example 1

- 3 applications into production
- Web portals
- *very complex employee transition management*
- Delivers every 2 weeks to production

- SIP ~10
 - *One Defect in last 9 months*



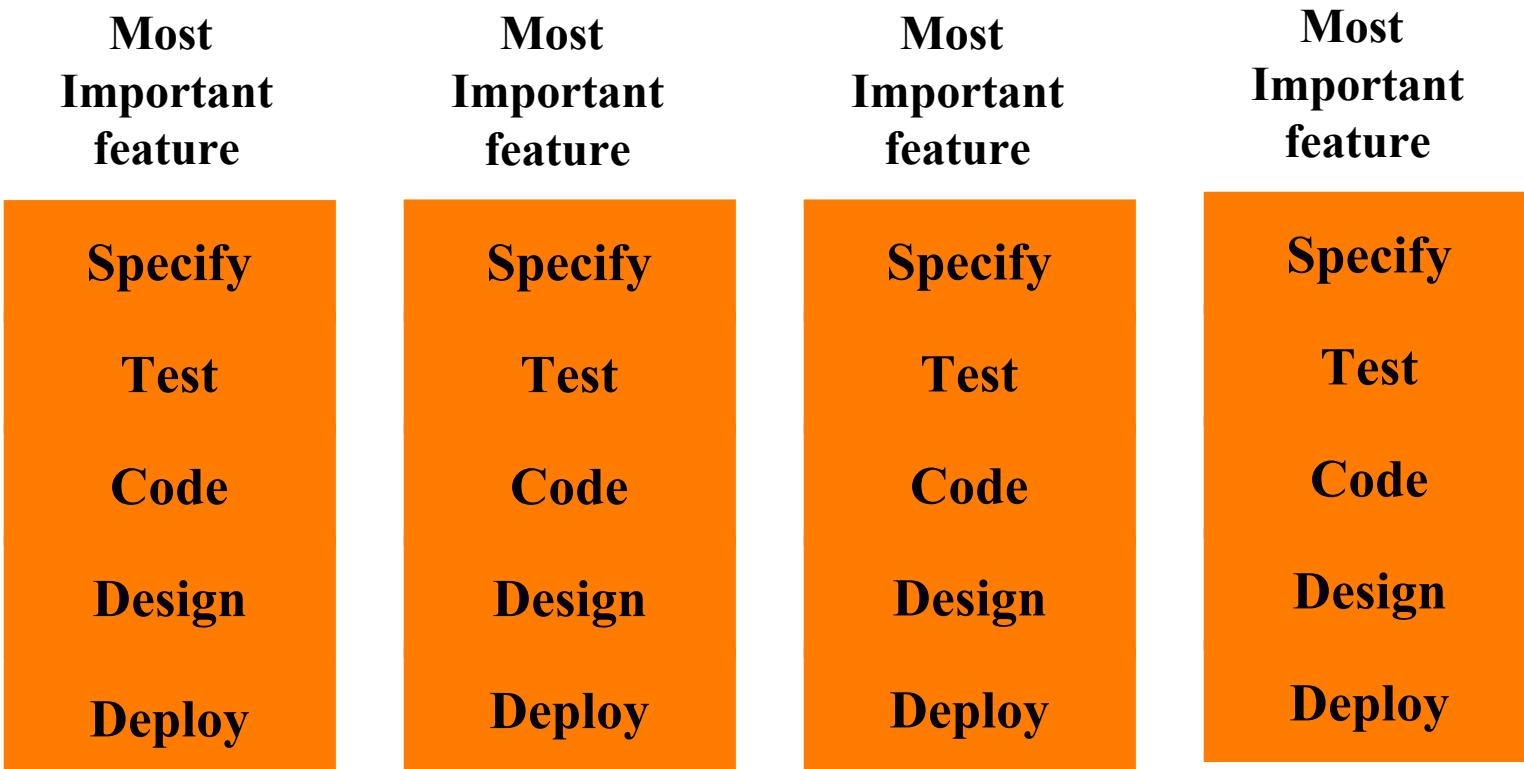
Agile: Name your #1 problem?

- That's Easy!
- Typical s/w project 6-12 months
- Quiz:
 - One year from now,
 - what will be your top 10 business problems?
- Ummm ??



Agile: Just In Time Software

- Let Customer choose Value
- Deep Focus on quickest payback





Delivering to Deadline



XP is about delivering to deadline.

- Every two weeks you release something
- and every two weeks you check your plan.

You get really good at getting stuff done.

- James Grenning, ObjectMentor



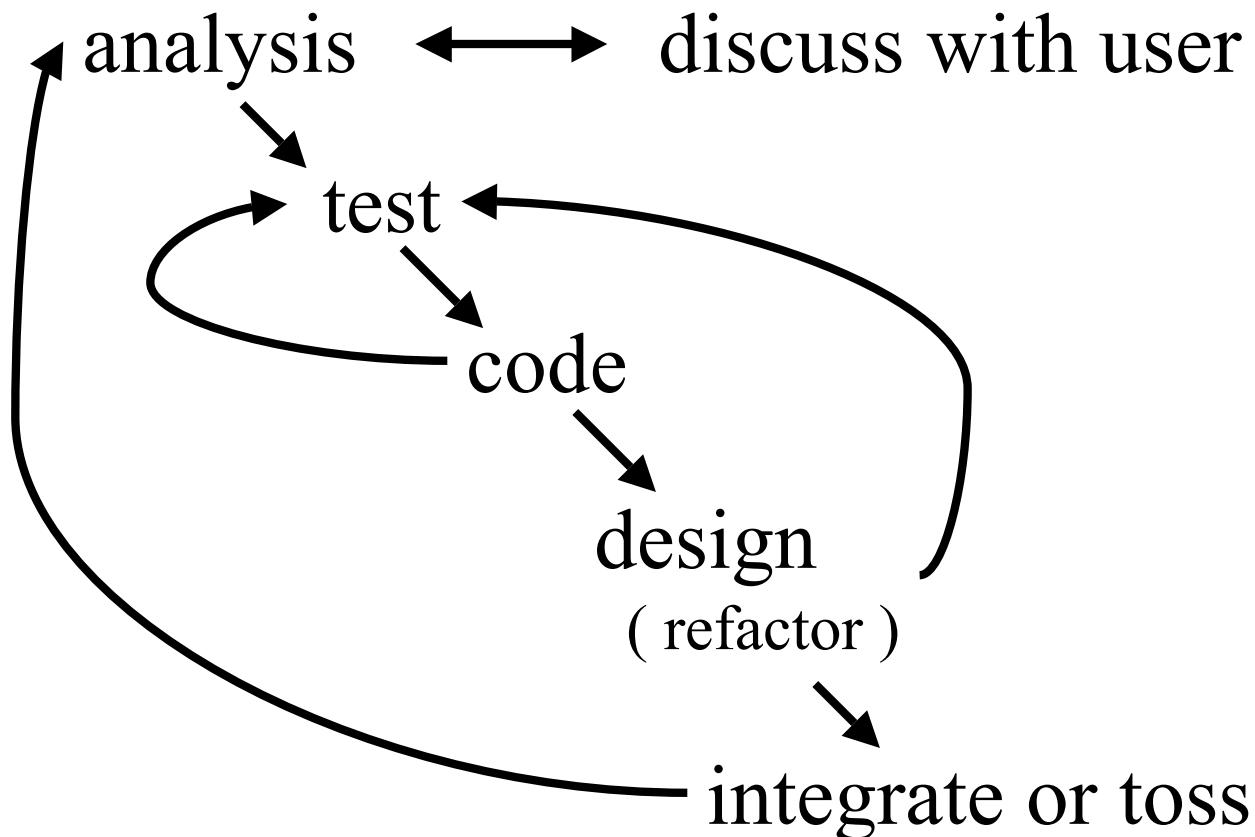


A Day in the life of XP

- 15 minute team stand-up meeting
- Get a story from task board
- Pair-up
- Talk with the customer sitting with the team
- Hold programming episodes
- Monitor the automated build & test system
- Deliver a few fully tested, integrated business Stories
- Go home at a reasonable time



XP Episode Cycle





Business to Code Process



Product is a bunch of ...

Release is a bunch of ...

Iteration is a bunch of ...

Acceptance Tested Stories is a bunch of ...

Episode is a bunch of ...

Test is a bunch of ...

Code



XP Values

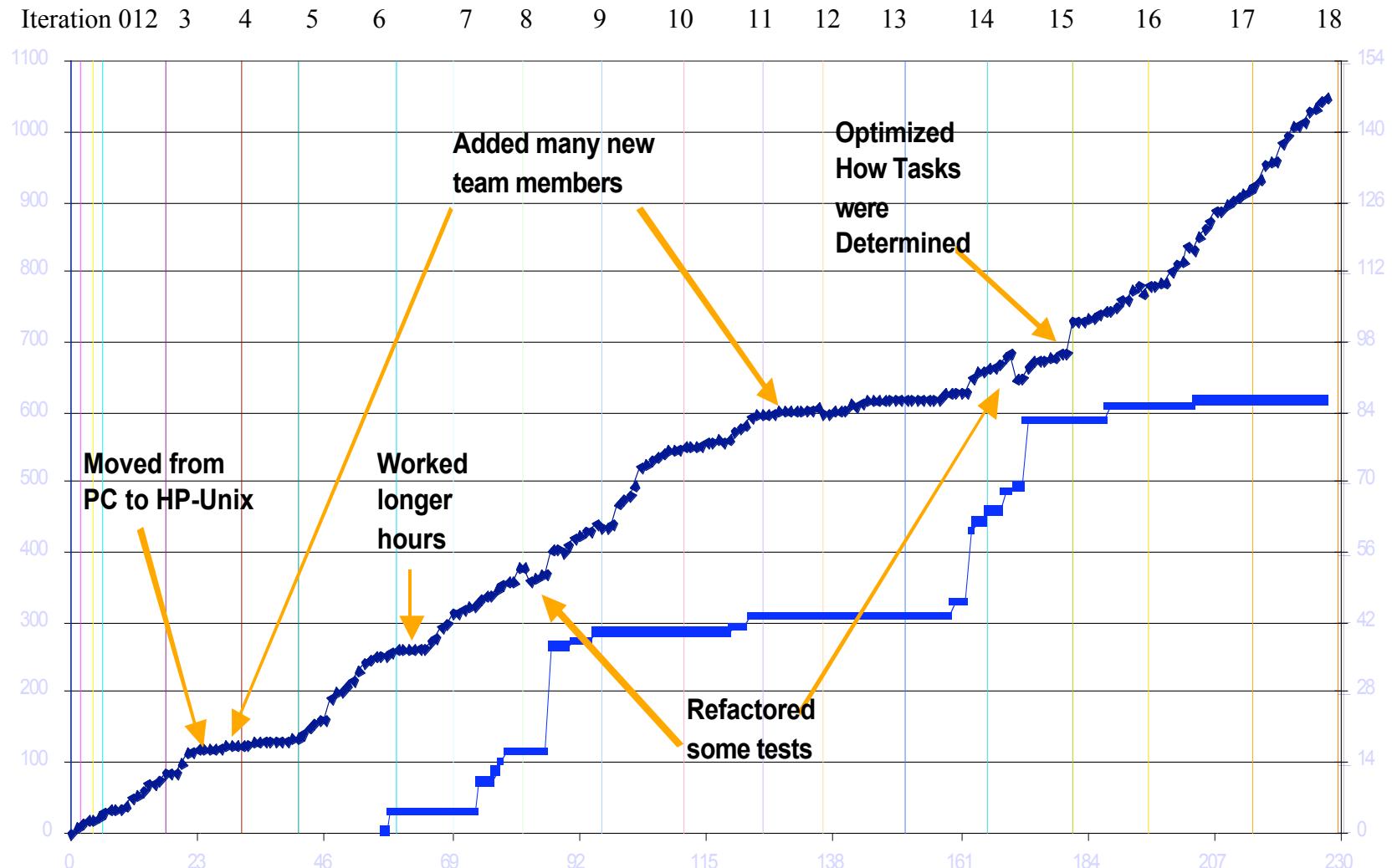
- *Communication* - visibility
- *Simplicity* - don't solve problems you don't have
- *Feedback* - start with the end in mind
- *Courage* - effective Action in the face of fear
- *Respect* - if people don't care, the project will fail



Sit Together / Informative Workspace



Unit & Functional Tests w/ Notes





Example 2

- www.lifeware.ch
- 4000 tests run with every change
- Changes go into production every evening
- Only needed workflows implemented
- Policy redemption not coded until requested
- Low cost of operation
- SIP = 1





Why is Agile not common practice?

- Requires a Culture shift



Culture of Fear

- Documentation reduces fear
- Sign-offs and distribute Blame
- Phases give *illusion* of progress

Mistakes & Defects ?

- Mistakes are not acceptable
 - No one got fired for choosing
 - ✓ IBM or Microsoft
 - ✓ Developing in Phases
- Defects are expected
 - Debug phase is scheduled



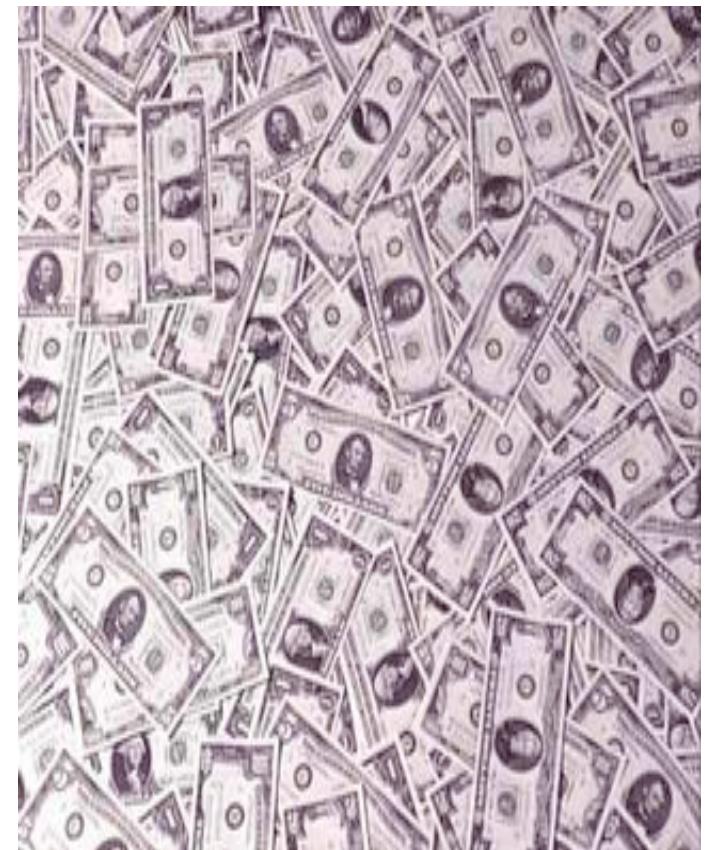


Culture of Greed

- Don't miss an opportunity
- Invest in small doses
- Expect Early ROI

Mistakes & Defects ?

- Mistakes are expected
 - Make them fast and recover
 - Communicate them
 - Learn
- Defects are unacceptable





Reference

Extreme Programming Explained: Embrace Change 2nd. Edition

by Kent Beck

Lean Software Development (An Agile Toolkit)

by Mary Poppendieck



Delivering to Deadline

XP is about delivering to deadline.

- Every two weeks you release something
- and every two weeks you check your plan.
You get really good at getting stuff done.

- James Grenning, ObjectMentor

