eXtreme Programming (XP)

Introducing XP

Recap

- Agile Manifesto
- http://www.agilemanifesto.org

- Do you think it Is it a useful statement about how to develop software systems?
- Would it be good for all types of system?

Recap

What do you know / remember about XP?

"Extreme Programming (XP) is about social change. It is about letting go of habits and patterns that were adaptive in the past, but now get in the way of us doing our best work. It is about giving up the defences that protect us but interfere with our productivity. It may leave us feeling exposed."

Kent Beck with Cynthia Andres (2005) Extreme Programming Explained 2nd Edition. Addison Wesley, Boston.

Source: ImproveIT, Flickr.
http://www.flickr.com/photos/improveit/1574023621/sizes/m/in/photostream/ Used under the Creative Commons Attribution-ShareAlike 2.0 Generic (CC BY-SA 2.0)

Overview

- Published by Kent Beck in 1999,
  - created earlier on C3 project
- Motivation behind XP
- The need to embrace change
- Perceived "cost of change" curve
- XP flattens the curve
- XP project lifecycle
- XP practices

Motivation behind XP: Problems with 'traditional' approaches

- Project success is just luck
- Integration nightmares
- Low sponsor expectations
- Smelly software: it's brittle, wrong features, missing features, buggy…

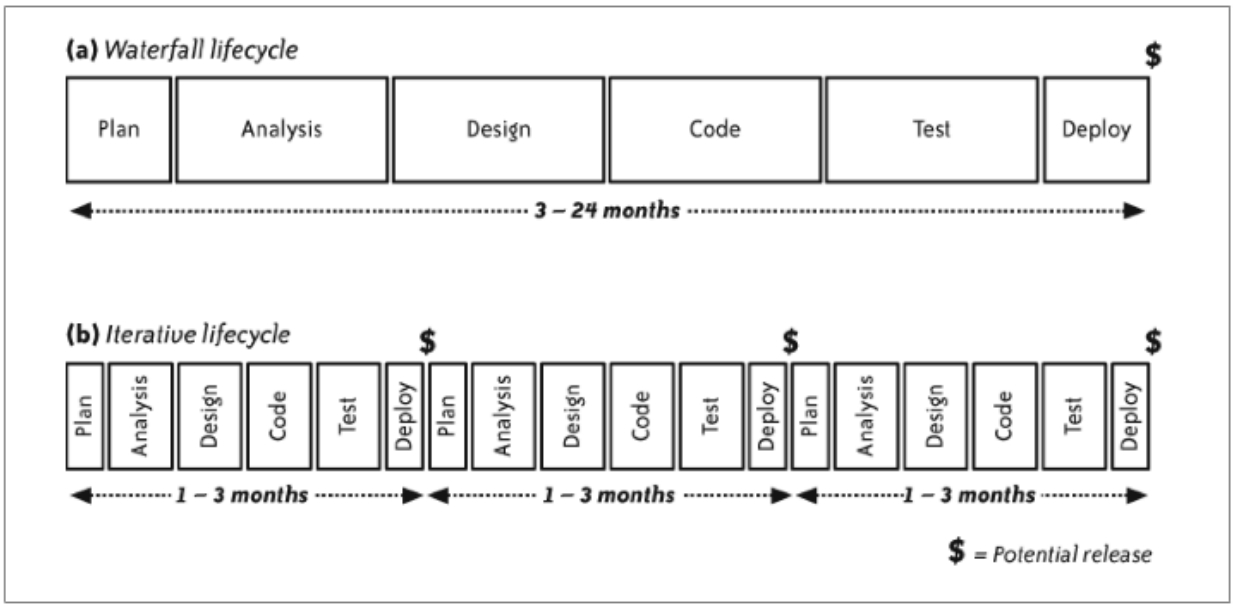


Figure 3-1. Traditional lifecycles

Source: Chapter 3 of Shore & Warden, p16

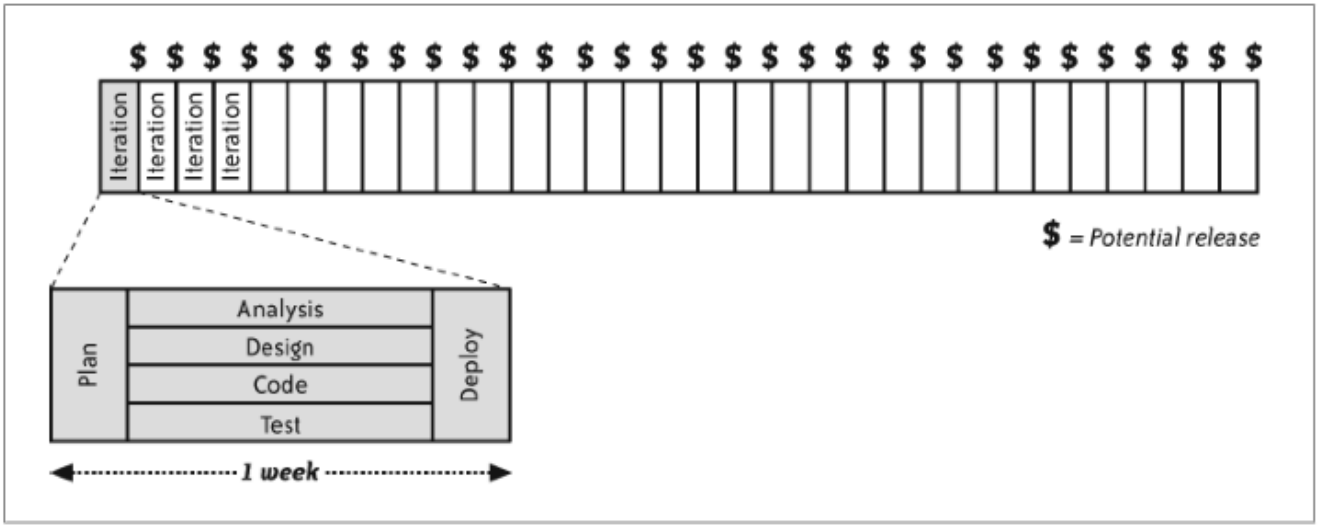
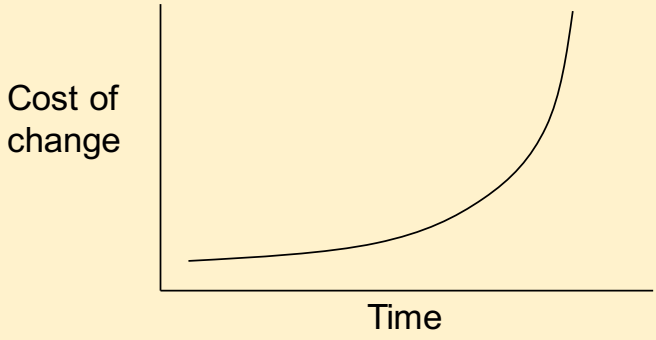


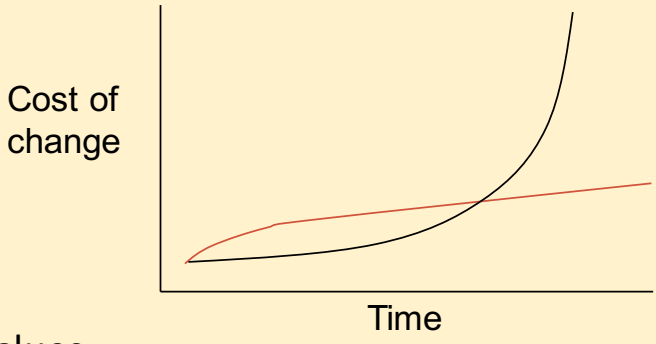
Figure 3-2. XP lifecycle

Source: Chapter 3 of Shore & Warden, p18

Perceived Cost of Change Curve



XP flattens the curve



- XP Values
  - Simplicity, Communication, Feedback, Courage, Respect (http://www.extremeprogramming.org/values.html)

XP Values

- **Simplicity**: Do what is needed, no more.
- **Communication**: XP demands communication
- **Feedback**: Continual feedback loop with stakeholders. Adapt process to fit the project.
- **Courage**: Courage to be open about progress. Courage to make changes.
- **Respect**: Between developers and between developers and stakeholders.

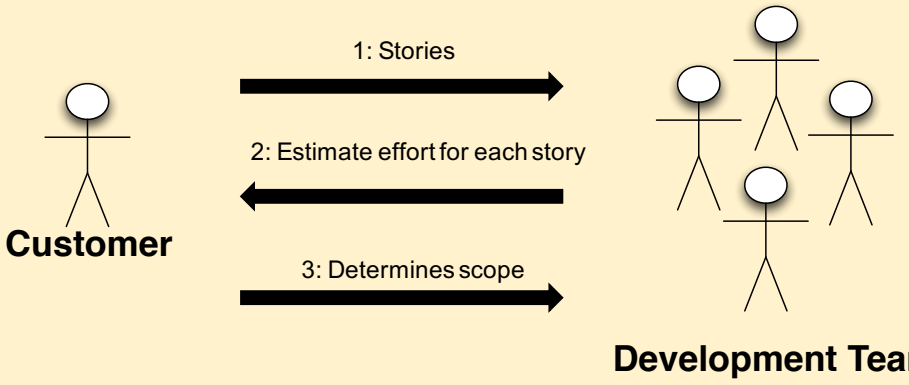
The XP project lifecycle

http://www.extremeprogramming.org/map/project.html

XP

If it is worth doing, do it a lot.

Practice 1: Planning Game



Practice 2
Test Driven Development

- Unit Testing – Tests the logic of the code
- Acceptance Testing – Tests the logic of the application
- Key is automation where possible.
  - Shore and Warden break this into additional categories, including exploratory testing.

Practice 3
Pair Programming

Design and development in pairs

Practice 4
Merciless Refactoring

Improving code without changing its functionality

Practice 5
Simple Design

- Incremental Design
- Find the simplest design that will solve the problem
- YAGNI
  - “You aren’t going to need it” principle

Practice 6
Collective Code Ownership

Everybody owns the code

If you break the code, you need to fix the code.

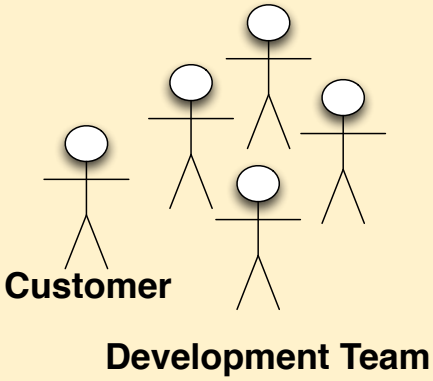
If you need to change the code, you can do that (but do speak to the team…)

Practice 7
Continuous Integration

Integrate the code in an integration platform several times a day.

Practice 8
On-site Customer

- The customer works as part of the team



The other four practices

- Practice 9: Small releases: providing business value as quickly as possible
- Practice 10: Sustainable development: avoid burnout and workforce attrition
- Practice 11: Apply "coding" standards: common look-and-feel to project artefacts
- Practice 12: System metaphor: Loosely relates to idea of architecture, e.g. shopping trolley and checkout metaphor

I will write about these on Blackboard later this week

Additional Practices

- Daily Stand-up Meeting
- Retrospectives

- Shore and Warden talk about Clarifying Practices – ones which "mature teams practice instinctively"

I will write about these on Blackboard later this week

Review

- "Social Change"
- XP started on the C3 Project, with Kent Beck as one of the three developers
- Five XP values
- 12 (original) practices
  - Additional clarifying practices