Unit 03
eXtreme Programming

eXtreme Programming and Agile Software Development

2011

Doctrine of Fair Use

- This PowerPoint is prepared for educational purpose and is strictly used in the classroom lecturing.
- We have adopted the "Fair Use" doctrine in this PowerPoint which allows limited copying of copyrighted works for educational and research purposes.

[7] Testing (Test First)

- Programmers write unit tests minute by minute.
- These tests are collected and they must all run correctly.

The Test/Code Cycle in XP

(1) Write an automated test case. Often developers write unit tests that test each method and everything that could possibly break, for example the logic of add(a,b) function can be easier to break add(4,-5) than add(4,5):

```
test() {
  if add(967, -11) == 956
  then return true
  else return false
}
```

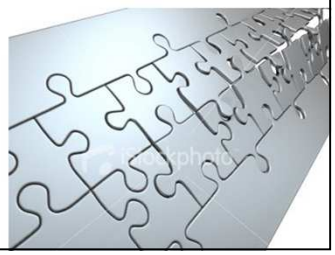
(2) Run the test case test(). It will return false since the add() function has not yet been written.

(3) Write only enough to make the add(a, b) function pass the test case.

(4) After refactoring the add(a, b) function each time, run the test case again.

[8] Continuous Integration

- New code is integrated with the current system after no more than a few (hours, days).
- When integrating, the system is built from scratch and all tests must pass or the changes are discarded.



[9] Pair Programming

- All production code is written by two people at one screen/keyboard/mouse.
- Driver: writing the design and code
- Observer: actively examines the work of the driver

Note

Pair Programming is a significant focal point of XP

One person is the "driver" and has control of the pencil/keyboard/mouse and is writing the design and code

The other person, the "observer," continuously and actively examines the work of the driver – watching for defects, thinking of alternatives, looking up resources, and considering strategic implications of the work at hand

Pair Programming

Things we do while pair programming

- Making design decisions
- Implementing code (test and production)
- Reviewing code (test and production)
- Testing code
- Refactoring code (test and production)
- Communicating, with your partner and indirectly with the whole team
- Educating each other and, again, indirectly the whole team

[10] Collective Code Ownership

- Collective Code Ownership means that any pair of programmers can make any change to any part of the source code at any time.
- No one person "owns" any part of the code.

Note

Collective Code Ownership encourages everyone to contribute new ideas to all segments of the project. Any developer can change any line of code to add functionality, fix bugs, or refactor.

No one person becomes a bottle neck for changes.

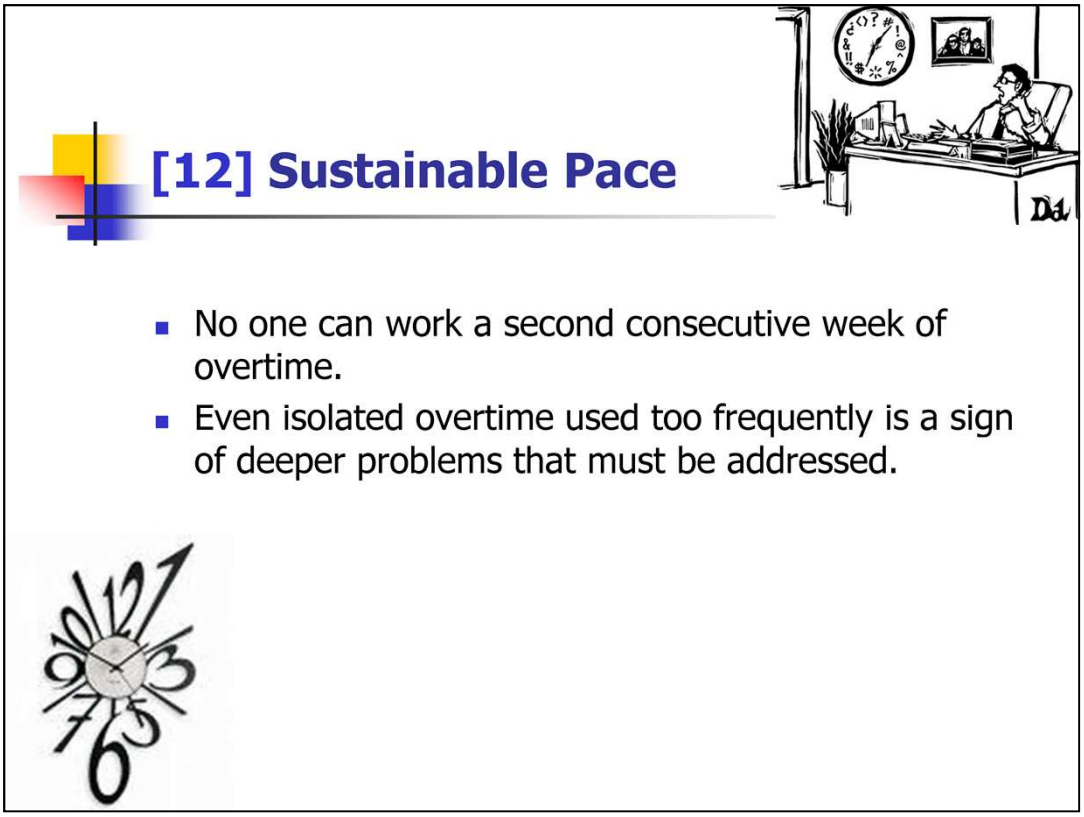
[11] Coding Standards

- The purpose of coding standards is to produce software that has a consistent style, independent of the author.

Note

In XP, code is collectively owned by all XP developers, not any individual. In order for the team to share all of the code, there must be a single coding standard to which everyone adheres.

This allows the XP team to be comfortable working on any of the code, since it all looks the same.



Note

The practice was originally called **40-hour Week.**

Tired programmers make more mistakes.

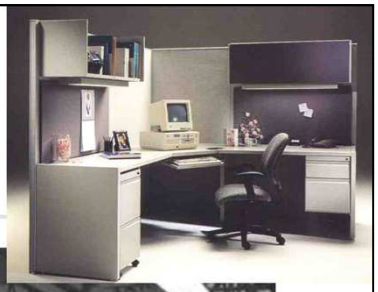
Extreme Programming advocates only working overtime if it is effective, and certainly not for an extended period of time.

Even isolated overtime used too frequently is a sign of deeper problems that must be addressed.

Workplace

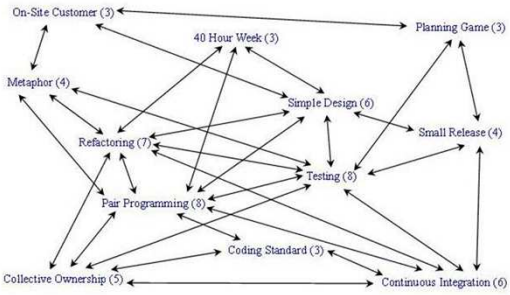
- All furniture against the walls is removed.
- Tables for computers are placed near the center.
- The walls are covered in giant whiteboards, and whiteboard-like static-cling sheets are used as wallpaper elsewhere.

Workplace



Group Discussion

- Which practices are controversial?
- Why?
- Which are “must practices” in programming?
- Why?



XP Adoption

Questions

- How do you start XP?
- Should we adopt all XP practices over a night?

Group Discussion

- Phase I _____
- Phase II _____
- Phase III _____
- Phase IV _____