

XP

eXtreme Programming (XP) is a software development discipline by Kent Beck in 1996. It is based upon four values:

- communication,**
- simplicity,**
- feedback, and**
- courage.**

It stresses continual communication between the customer and development team members by having an on-site customer decides what will be built and in what order. You embody simplicity by continually refactoring code and producing a minimal set of non-code artifacts. Many short releases and continual unit testing are the feedback mechanisms. Courage mean doing the right thing, even when it is not the most popular thing to do. This means being honest about what you can and cannot do.

XP's twelve practices:

- The planning game
- Small releases
- Metaphor
- Simple design
- Testing
- Refactoring
- Pair programming
- Collective ownership
- Continuous integration.
- Forty-hour week.
- On-site customer
- Coding standards

The planning game

Determine the features in the next release through a combination of stories and technical estimates.

Small releases

Release the software often to the customer with small incremental versions.

Metaphor

- The metaphor is a simple shared story or description of how the system works.

Simple design

Keep the design simple by keeping the code simple. Continually look for complexity in the code and remove it at once.

Testing

- The customer writes tests to test the stories. Programmers write tests to test anything that can break in the code. You write test before you write code.

Refactoring

- This is a simplifying technique to remove duplication complexity from code.

Pair programming

- Teams of two programmers at a single computer develop all of the code. One writes the code, or *drives*, while the other reviews the code from correctness and understandability at the same time.

Collective ownership

- Everyone owns all of the code. This means everyone has the ability to change any code at any time.

Continuous integration

- Build and integrate the system several times a day whenever any implementation task is completed.

Fourty hour week

- Programmers cannot work at peak efficiency if they are tired. Overtime is never allowed for two consecutive weeks.

On -site customer

- A real customer works in the development environment full-time to help define the system, write tests, and answers questions.

Coding standards

- The programmers adopt a consistent coding standard.

XP phases

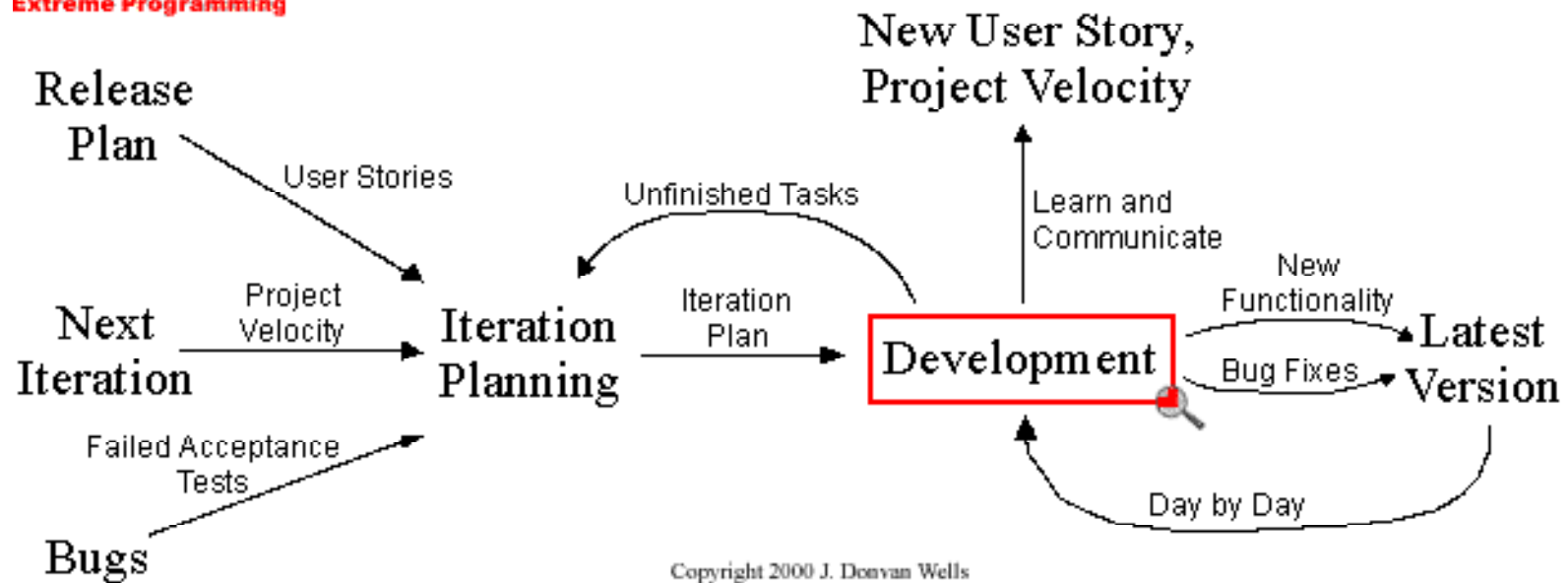
- **Exploration**
- **Commitment**
- **Steering**

XP iteration



Iteration

Zoom Out

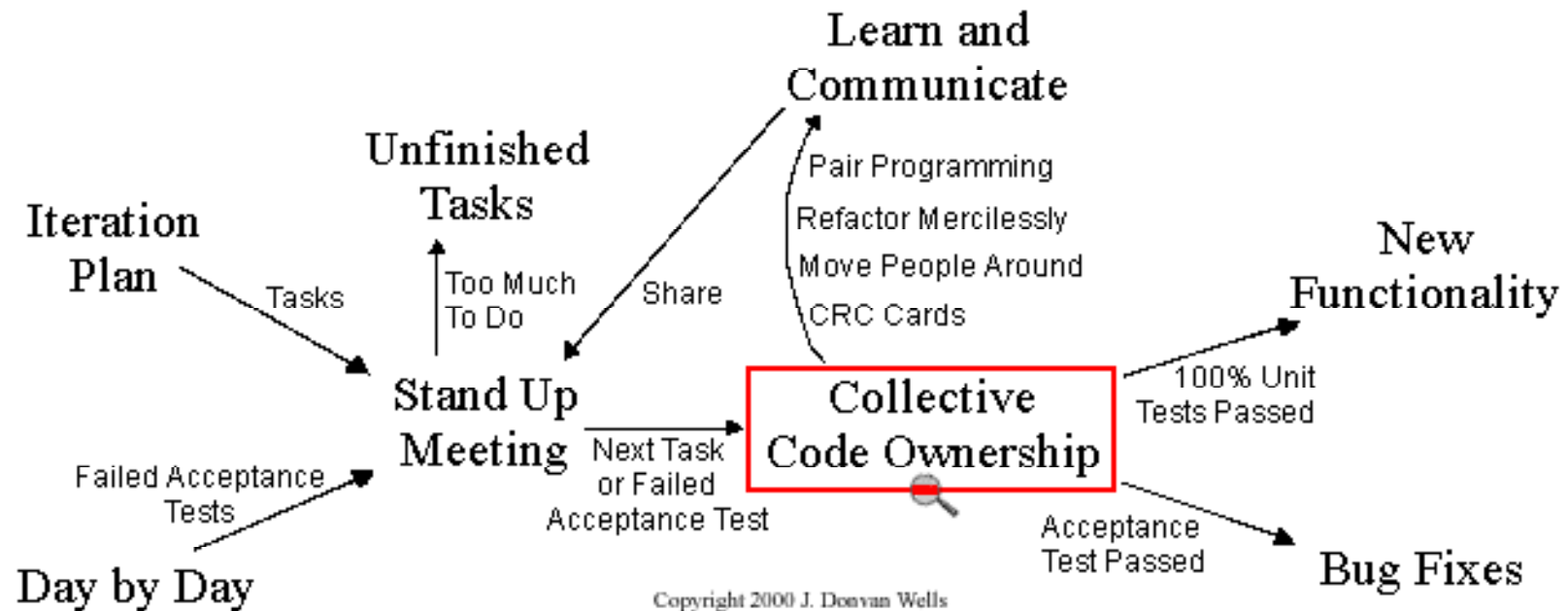


XP: development



Development

Zoom Out

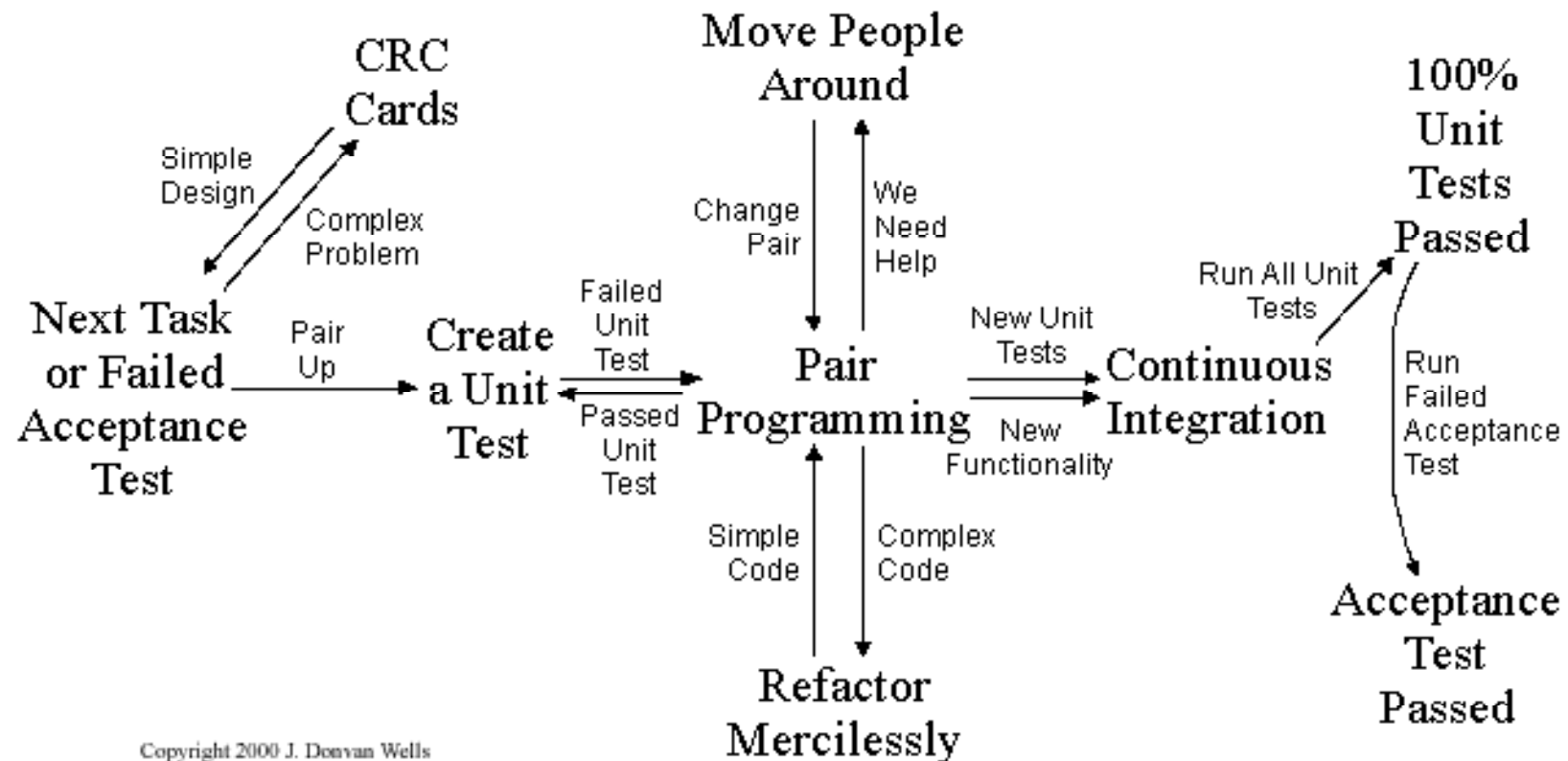


Collective Code Ownership



Collective Code Ownership

Zoom Out



Planning

- **User stories are written.**
- **Release planning creates the schedule.**
- **Make frequent small releases.**
- **The Project Velocity is measured.**
- **The project is divided into iterations.**
- **Iteration planning starts each iteration.**
- **Move people around.**
- **A stand-up meeting starts each day.**
- **Fix XP when it breaks.**

Designing

- **Simplicity.**
- **Choose a system metaphor.**
- **Use CRC cards for design sessions.**
- **Create spike solutions to reduce risk.**
- **No functionality is added early.**
- **Refactor whenever and wherever possible.**

Coding

- **The customer is always available.**
- **Code must be written to agreed standards.**
- **Code the unit test first.**
- **All code is pair programmed.**
- **Only one pair integrates code at a time.**
- **Integrate often.**
- **Use collective code ownership.**
- **Leave optimization till last.**
- **No overtime.**

Testing

- **All code must have unit tests.**
- **All code must pass all unit tests before it can be released.**
- **When a bug is found tests are created.**
- **Acceptance tests are run often and the score is published.**