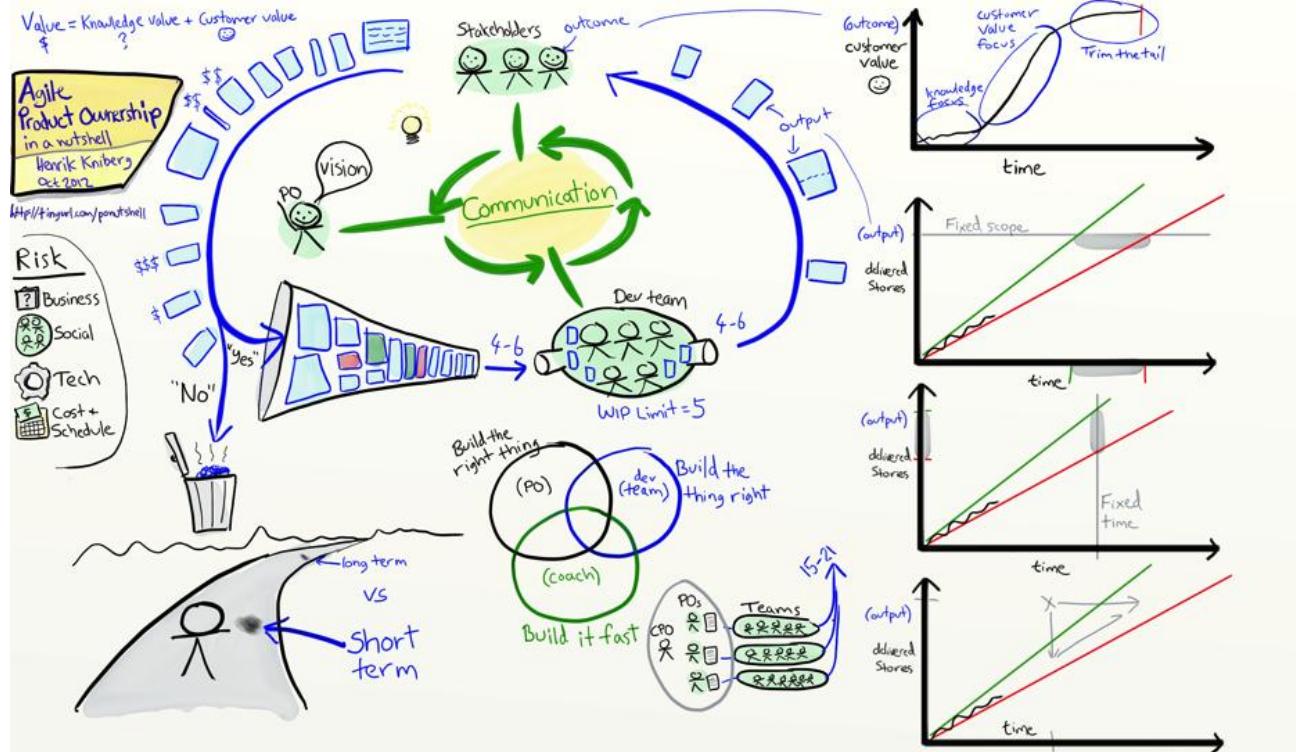


# **What is XP?**

# Agile in a nutshell



# Agile Manifesto

[www.agilemanifesto.org](http://www.agilemanifesto.org)

We are uncovering better ways of developing software by doing it and helping others do it.  
Through this work we have come to value:

**Individuals and interactions over processes and tools**

**Working software over comprehensive documentation**

**Customer collaboration over contract negotiation**

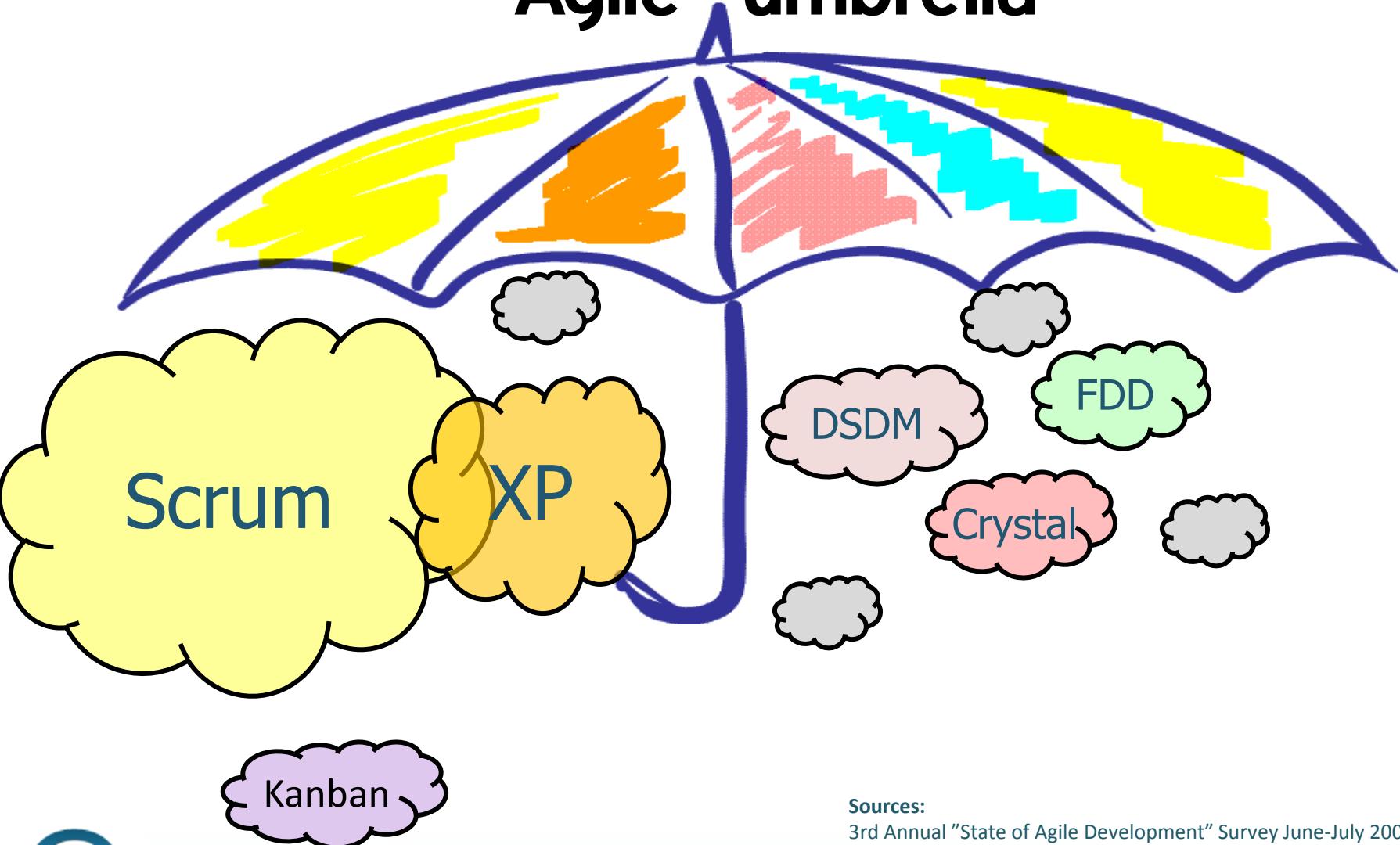
**Responding to change over following a plan**

That is, while there is value in the items on the right, we value the items on the left more.

# Principles behind the Agile Manifesto

- Our highest priority is to **satisfy the customer** through early and continuous delivery of valuable software.
- **Welcome changing requirements**, even late in development. Agile processes harness change for the customer's competitive advantage.
- **Deliver working software frequently**, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- **Business people and developers must work together** daily throughout the project.
- Build projects around **motivated individuals**. Give them the environment and support they need, and **trust** them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is **face-to-face conversation**.
- **Working software** is the primary measure of progress.
- Agile processes promote **sustainable development**. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to **technical excellence** and **good design** enhances agility.
- **Simplicity**--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements, and designs emerge from **self-organizing teams**.
- At regular intervals, the team **reflects on how to become more effective**, then tunes and adjusts its behavior accordingly.

# Agile "umbrella"



## Sources:

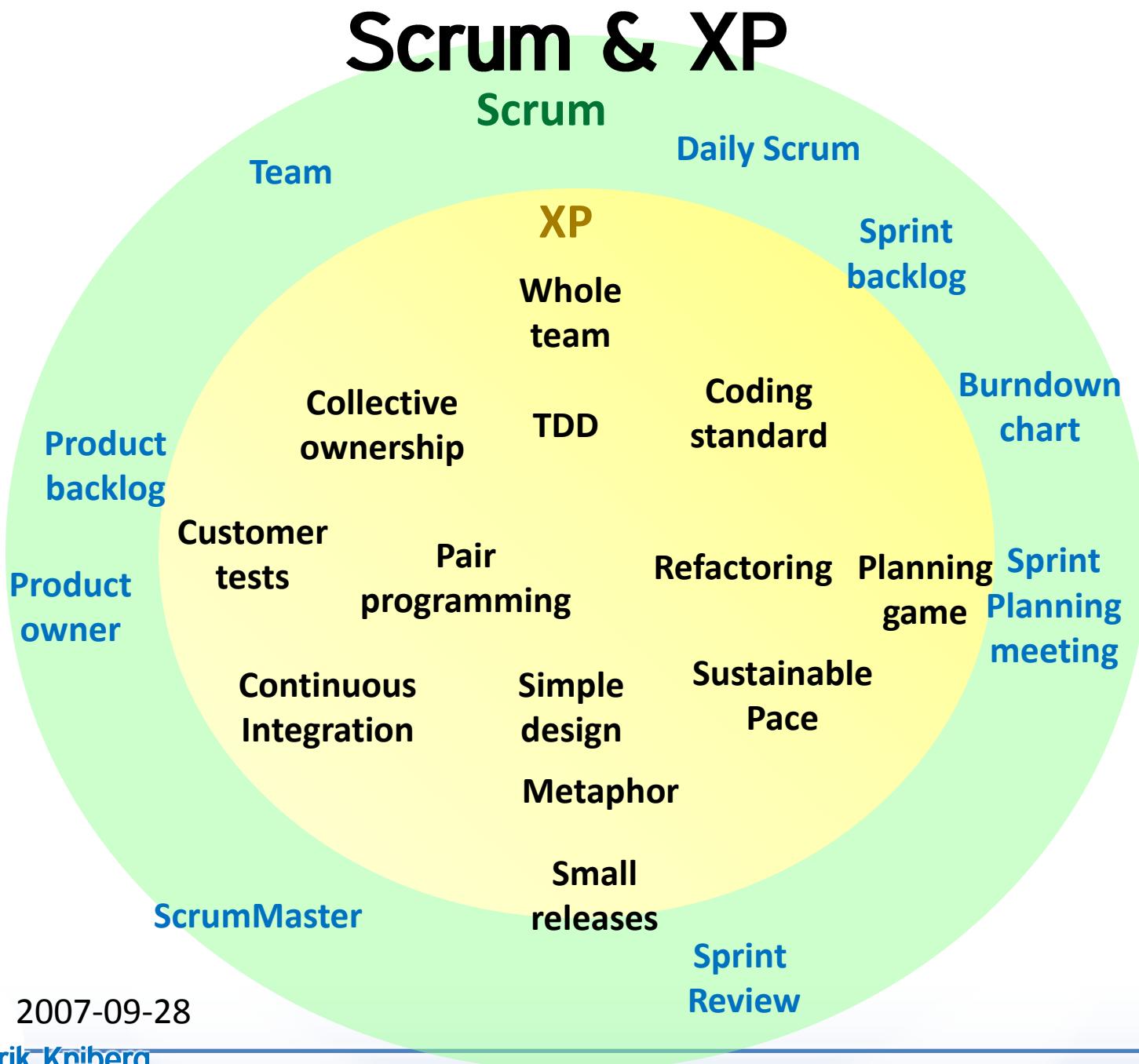
3rd Annual "State of Agile Development" Survey June-July 2008

- 3061 respondents
- 80 countries

# XP values

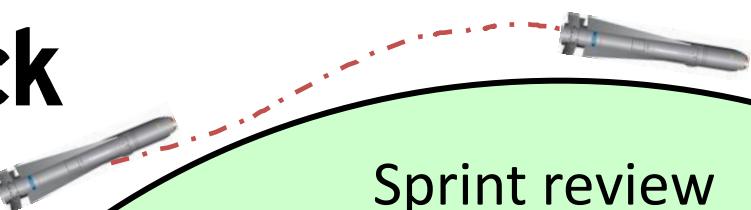
- Simplicity
- Communication
- Feedback
- Respect
- Courage

# Scrum & XP



2007-09-28

# Feedback loops

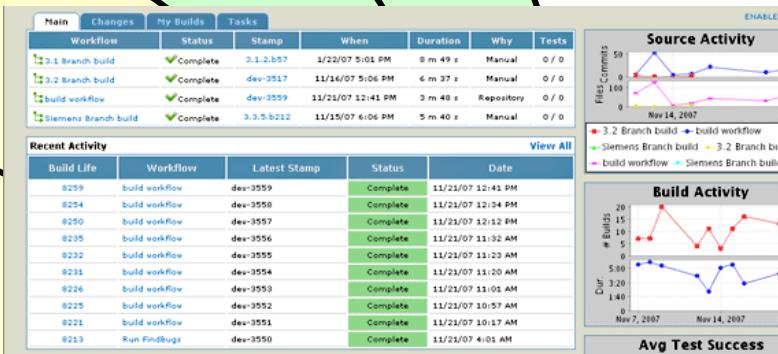


Sprint review



Daily Scrum

Continuous integration



Unit test

Pair  
programming



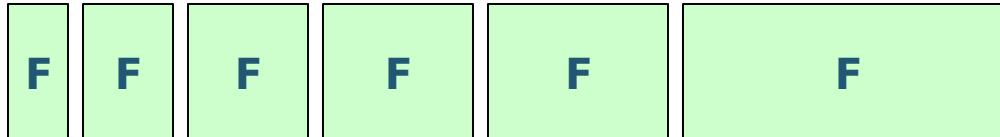
# Agile architecture

Timeline →

Don't do: Quick 'n dirty features

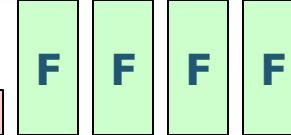
=> Slow 'n dirty

=> Entropy death? Big bang rewrite?

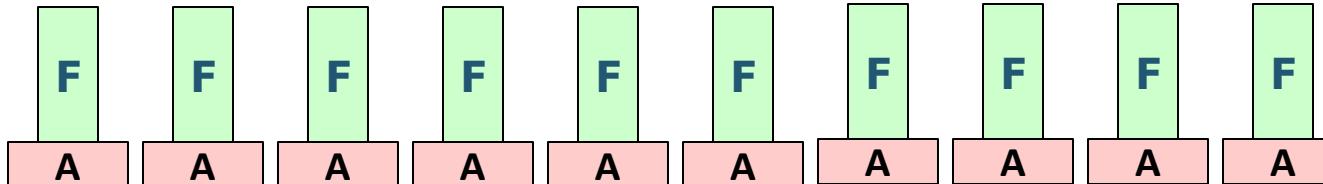


Don't do: Big Up Front Design

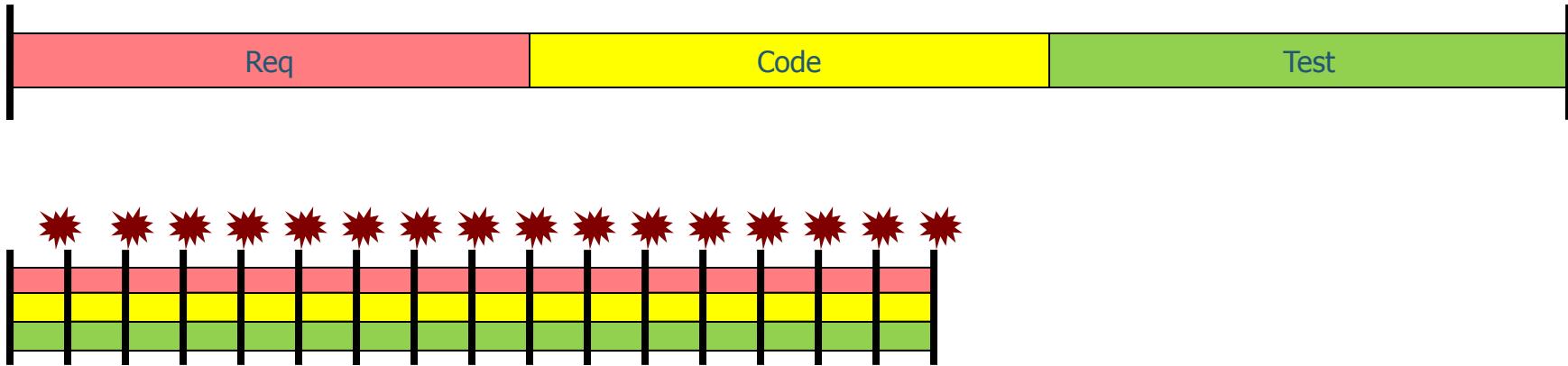
Architecture



Do: Find a balance



# Releasing must be REALLY easy



# Clean code & build environment



```
import java.sql.Connection;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;

public class Dog {
    private Executor executor = Executors.newFixedThreadPool(10);
    private int CACHE_SIZE = 50;

    public Dog()
    {
        try
        {
            Class.forName("oracle.jdbc.ThinDriver");
            connection =
DriverManager.getConnection("jdbc:oracle:thin:@prod", "admin",
"beefhead");
            statement = connection.prepareStatement("insert into Dog
values (?, ?, ?)");
            } catch (ClassNotFoundException e) {}

new Thread().start();
}

public void woof(Person woofCaller) {
Connection connection = null;
PreparedStatement statement = null;
try {
    connection =
DriverManager.getConnection("jdbc:oracle:thin:@prod", "admin",
"beefhead");
    statement = connection.prepareStatement("insert into Dog
values (?, ?, ?)");
    statement.setLong(1, System.currentTimeMillis());
    statement.setString(2, person.getName());
    statement.setString(3,
person.getPhoneNumber().getNumber());
    statement.executeUpdate();
}
}
}

Connection a =
```

public class Dog {  
 private final String name;  
 private int woofCount = 0;  
  
 public Dog(String name) {  
 this.name = name;  
 }  
  
 public void woof() {  
 ++woofCount;  
 }  
}

Main	Changes	My Builds	Tasks
Workflow	Status	Stamp	
3.1 Branch build	✓ Complete	3.1.2.b57	
3.2 Branch build	✓ Complete	dev-3517	
build workflow	✓ Complete	dev-3559	11/21/07 12:41 PM 3 m 48 s Repository 0 / 0
Siemens Branch build	✓ Complete	3.3.5.b212	11/15/07 6:06 PM 5 m 40 s Manual 0 / 0

Recent Activity

Build Life	Workflow	Latest Stamp	Status	Date
8259	build workflow	dev-3559	Complete	11/21/07 12:41 PM
8254	build workflow	dev-3558	Complete	11/21/07 12:34 PM
8250	build workflow	dev-3557	Complete	11/21/07 12:32 PM
8235	build workflow	dev-3556	Complete	11/21/07 11:32 AM
8232	build workflow	dev-3555	Complete	11/21/07 11:23 AM
8231	build workflow	dev-3554	Complete	11/21/07 11:20 AM
8226	build workflow	dev-3553	Complete	11/21/07 11:01 AM
8225	build workflow	dev-3552	Complete	11/21/07 10:57 AM
8221	build workflow	dev-3551	Complete	11/21/07 10:57 AM
8213	Run FindBugs	dev-3550	Complete	11/21/07 4:01 AM

# Clean & simple code

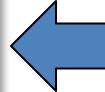
Simple code:

1. Passes all tests
2. No duplication
3. Readable
4. Minimal

Simple is hard!

~~Code is an asset~~  
~~All code is cost!~~  
Some code is value.

```
public class Dog {  
    private final String name;  
    private int woofCount = 0;  
  
    public Dog(String name) {  
        this.name = name;  
    }  
  
    public void woof() {  
        ++woofCount;  
    }  
}
```



```
import java.sql.Connection;  
import java.util.concurrent.ExecutorService;  
import java.util.concurrent.Executors;  
  
public class Dog {  
    private Executor executor = Executors.newFixedThreadPool(18);  
    private int CACHE_SIZE = 50;  
  
    public Dog()  
    {  
        try  
        {  
            Class.forName("oracle.jdbc.ThinDriver");  
            connection = DriverManager.getConnection("jdbc:oracle:thin:@prod", "admin",  
"beefhead");  
            statement = connection.prepareStatement("insert into Dog values (?, ?, ?)");  
        } catch (ClassNotFoundException e) {}  
  
        new Thread().start();  
    }  
  
    public void woof(Person woofCaller) {  
        Connection connection = null;  
        PreparedStatement statement = null;  
        try {  
            connection = DriverManager.getConnection("jdbc:oracle:thin:@prod", "admin",  
"beefhead");  
            statement = connection.prepareStatement("insert into Dog values (?, ?, ?)");  
            statement.setLong(1, System.currentTimeMillis());  
            statement.setString(2, person.getName());  
            statement.setInt(3, person.getPhoneNumber().getNumber());  
            statement.executeUpdate();  
        }  
        }  
        Connection a = DriverManager.getConnection("jdbc:oracle:thin:@prod", "admin", "beefhead");  
        b = a.prepareStatement("select * from Dog where name = '" + name + "'");  
        c = b.executeQuery();  
        if (c.next()) {  
            String foundName = c.getString("name");  
            PhoneNumber phoneNumber = new PhoneNumber(c.getString("woofCount"));  
            Person person = new Person(foundName, phoneNumber);  
            return person;  
        } else {  
            return new Person("", null);  
        }  
    } catch (SQLException e) {  
        return null;  
    } catch (IllegalArgumentException x) {  
        throw x;  
    }  
}  
  
public List<Person> getAll() {  
    Connection connection = DriverManager.getConnection("jdbc:oracle:thin:@prod", "admin",  
"beefhead");  
    Statement statement = connection.prepareStatement("insert into Dog values (?, ?, ?)");  
    statement.setLong(1, System.currentTimeMillis());  
}  
if (statement != null) {  
    if (c.next()) {  
        String foundName = c.getString("name");  
        PhoneNumber phoneNumber = new PhoneNumber(c.getString("woofCount"));  
        Person person = new Person(foundName, phoneNumber);  
        return person;  
    } else {  
}
```



Embrace  
Change!

Henrik  
Kniberg