

Faculty of Mathematics and Physics  
Charles University in Prague  
27<sup>th</sup> April 2015



C# Made Easy!

# Programming II

Workshop 10 – XP

# Workshop 10

## Outline

1. Test
2. Extreme Programming
3. Homework



# Test 10

## Test

**Find the test here (no-ads):**

<http://goo.gl/OP4G2j>

**Permanent link:**

<https://docs.google.com/forms/d/1Dw7DXoENalQWOyzvXzkhUxVenY4e4CcAlfaM3Gp3sXk/viewform>

**Time for the test:**

5 min

# Extreme Programming

## Software Development Methodology

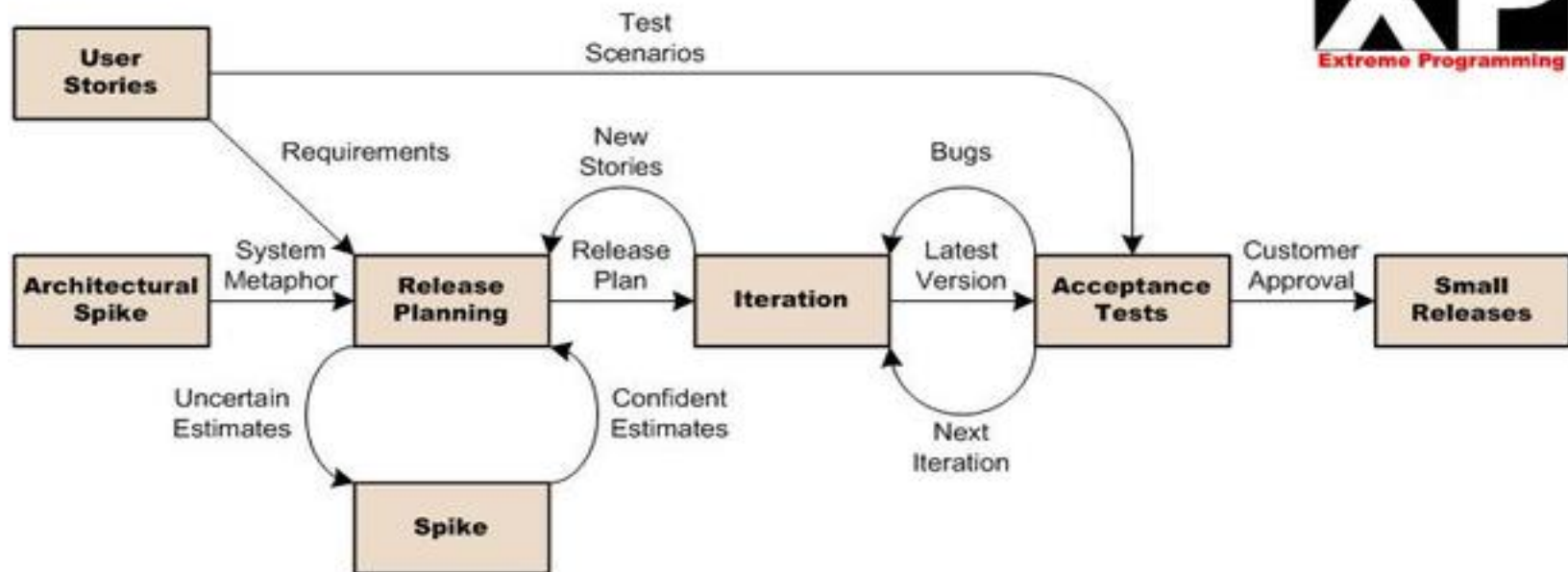


www.dilbert.com scottadams@aol.com

11-14-07 ©2007 Scott Adams, Inc./Dist. by UFS, Inc.

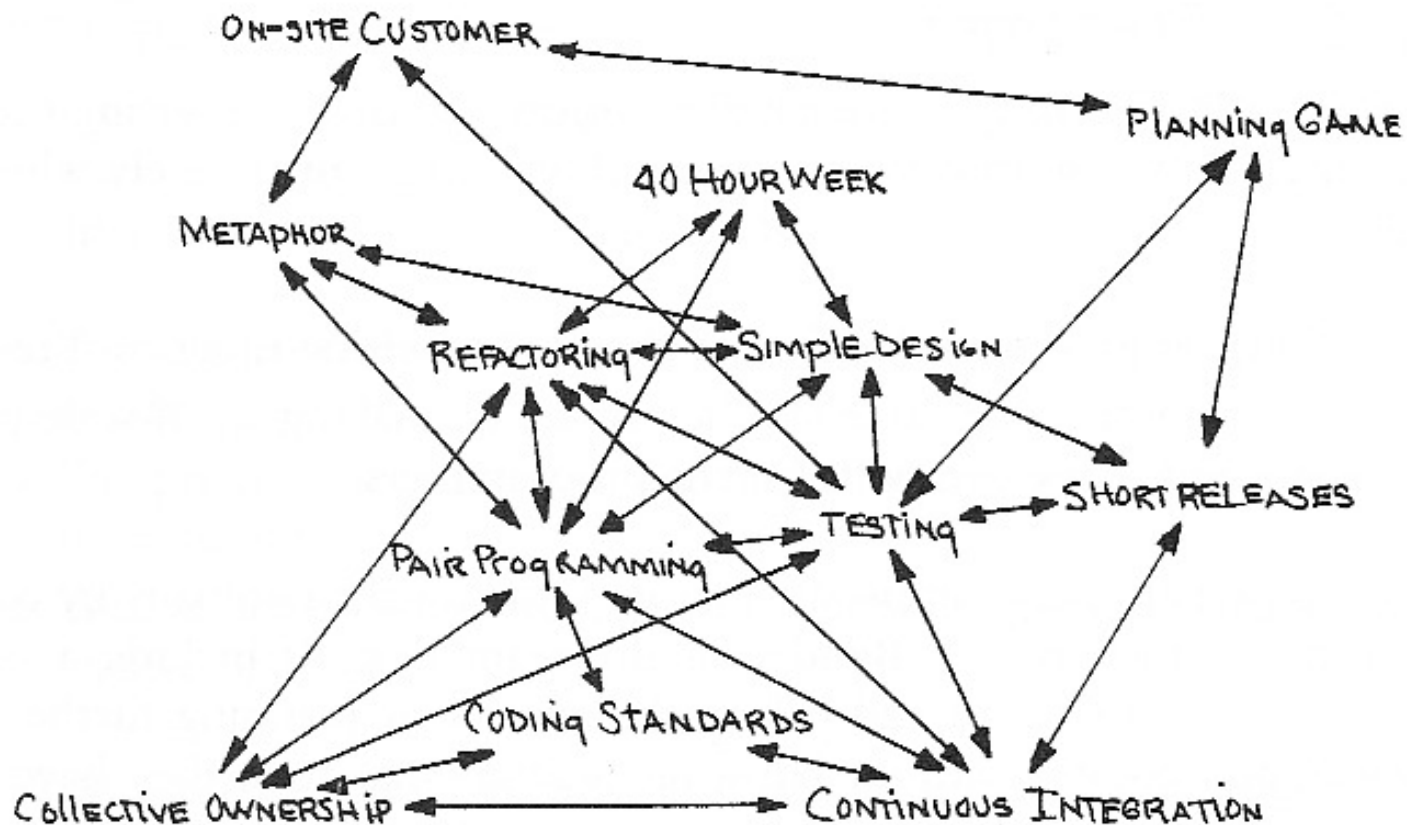
# Extreme Programming

## Software Development Methodology



# Extreme Programming

## Software Development Methodology



# Extreme Programming

## Software Development Methodology

- Important for teams!
  - Which we sort-of lack here...
- Alas, we're going to investigate the following:
  - The Code is The Documentation
    - => The code should speak for itself
  - Pair Programming
    - => Helps you to focus your thoughts and write bug-less code

# Extreme Programming

## The Code is The Documentation

- Apart from obvious naming conventions...
- Name your temporary variables well

```
public Node Add(int num) {  
    Node n = new Node(num);  
    ...  
    return n;  
}
```

```
public Node Add(int num) {  
    Node result = new Node(num);  
    ...  
    return result;  
}
```



# Extreme Programming

## The Code is The Documentation

- Apart from obvious naming conventions...
- Avoid obvious I, J, K variable names in for-loops

```
for (int i = 0;  
    i < lines.Count();  
    ++i) {  
...  
}
```

```
for (int lineIndex = 0;  
    lineIndex < lines.Count();  
    ++ lineIndex ) {  
...  
}
```

# Extreme Programming

## The Code is The Documentation

- Apart from obvious naming conventions...
- Document the idea behind the code, not what the code is doing
- Document contracts

```
// Returns ROOT node  
Public NODE GetRoot();
```

```
// Returns ROOT of the tree that is  
guaranteed to remain the same  
throughout the life of the TREE  
object  
Public NODE GetRoot()
```

# Extreme Programming

## The Code is The Documentation

- BEWARE!
- *The code is the imperfect translation into a programming language of the programmer's imperfect understanding about what the program should do.*
- ⇒ If unsure how to code your idea, write down your idea/objective in plain language (e.g. as a comment to a class, a method or code block) and leave it there after you code it
  - And after you code your idea/objective, review your comment if it still holds!
- *The documentation is a set of hypotheses to be tested and not a set of axioms to be trusted. I think that this brings even more light to what you are trying to show.*
- ⇒ You will never know whether the method/class/sub-system behaves as documented / expected until you try
- ⇒ If you are unsure about the technology, do not go on wild implementing features in real-project, play with the technology elsewhere, safely

# Extreme Programming

## Pair Programming

- Two roles: Driver and Navigator
- Driver
  - Writes the code
- Navigator
  1. Reviews each line of the code
    - Typos
    - Coding standards
    - Bugs
  2. Thinks about “next step”
- Let's form pairs!

# Extreme Programming

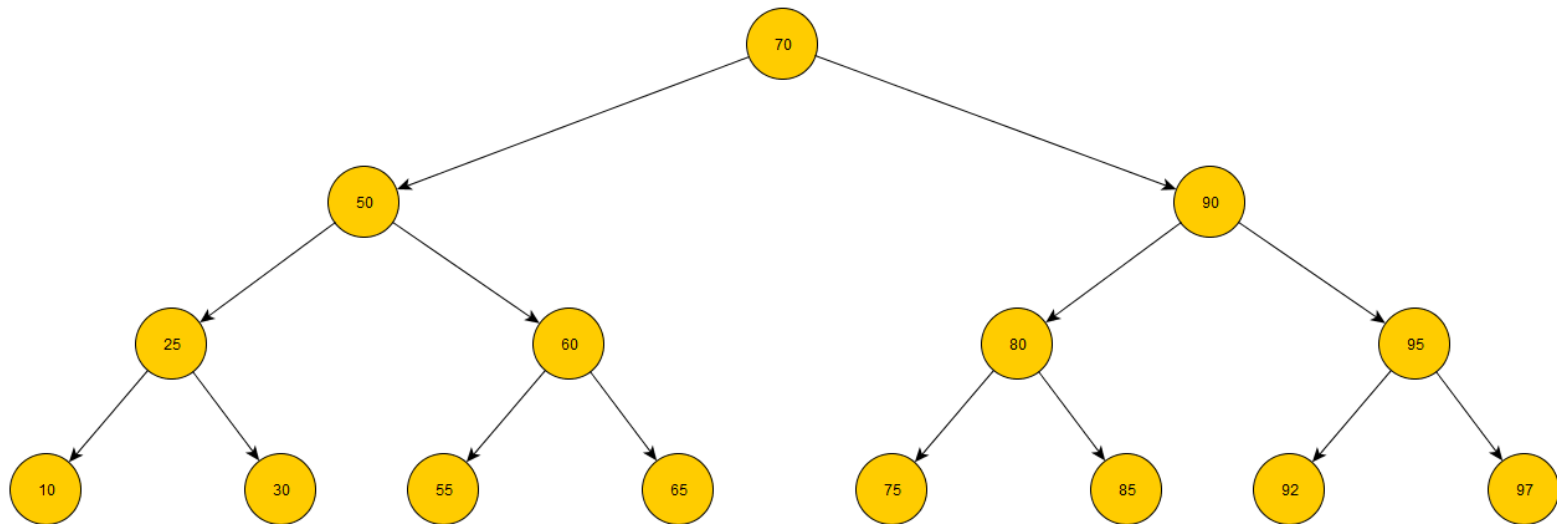
## Task – Visualization of Binary Search Tree

- Download the template: <http://goo.gl/WkLMWR>
- Provide a way to visualize a binary search tree
  - Binary tree-like layout
    - Node as a circle with a number in its center
    - Edges between parent-child
  - Repaint on screen resize
  - Always fit into “the entire window”
  - And be warned... the customer will likely need to change this layouting in the future!
    - => Try to separate “drawing commands” from the “layouting algorithm”

# Extreme Programming

## The Task – Visualization of Binary Search Tree

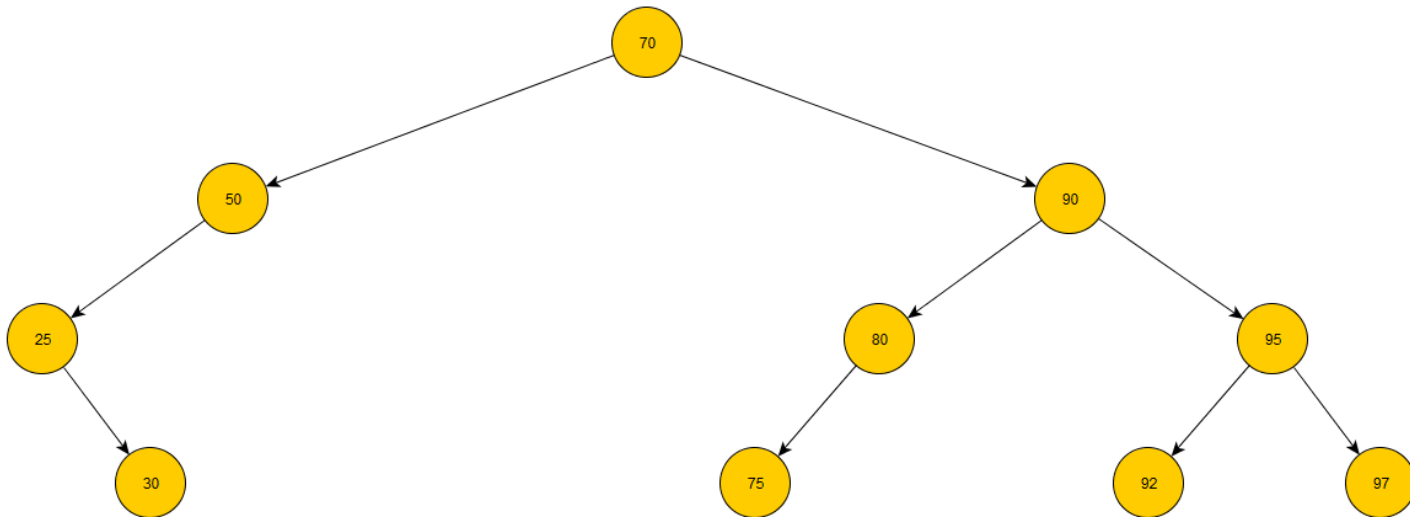
- Fixed Layout



# Extreme Programming

## The Task – Visualization of Binary Search Tree

- Fixed Layout
  - Tree Height / Layer-depth determines the layout



# Extreme Programming

## The Task – Visualization of Binary Search Tree

- GOOD LUCK!
  1. Decide on Driver & Navigator
  2. Analyze existing code base together
  3. Analyze the task together and come up with solution for the layouting algorithm
  4. Design an architecture for algorithm implementation
    - Beware, the layouting algorithm will likely be changed in the future
    - But do not overengineer this!
  5. Code it!



# Extreme Programming

## Homework – Flexible Layout for Binary Tree

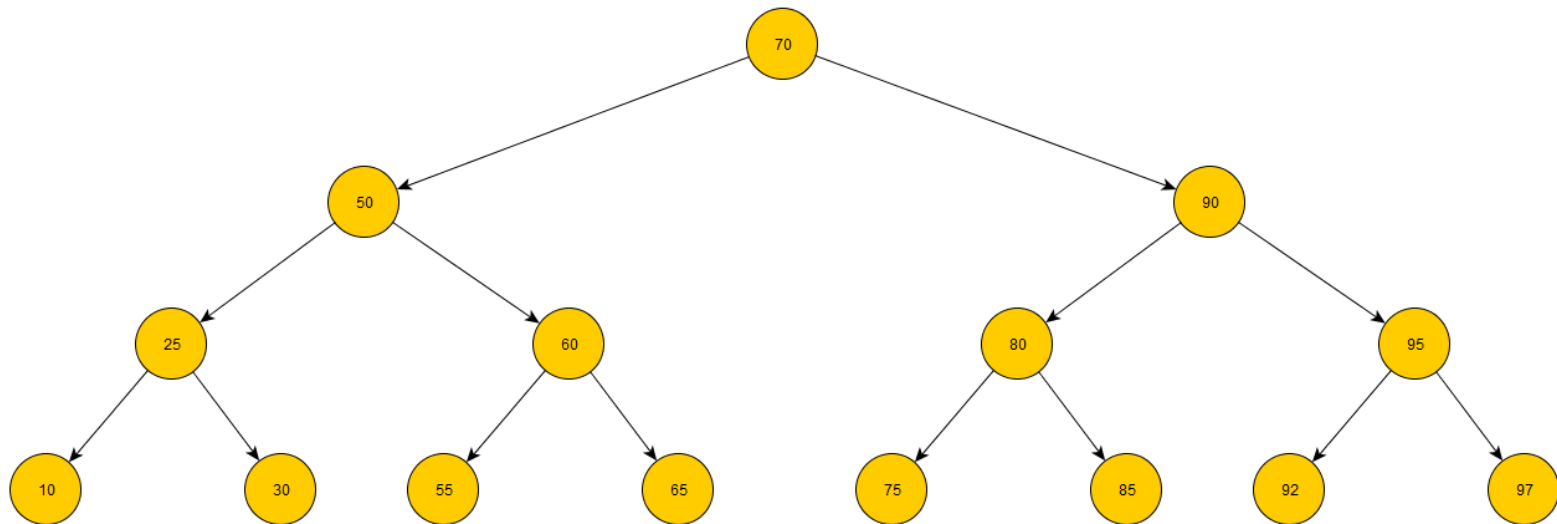
Continue the work on your code alone and:

1. Provide a way to add “multiple numbers comma separated” at once (new text box, new button)
2. Implement flexible layout for the tree

# Extreme Programming

## Homework – Flexible Layout for Binary Tree

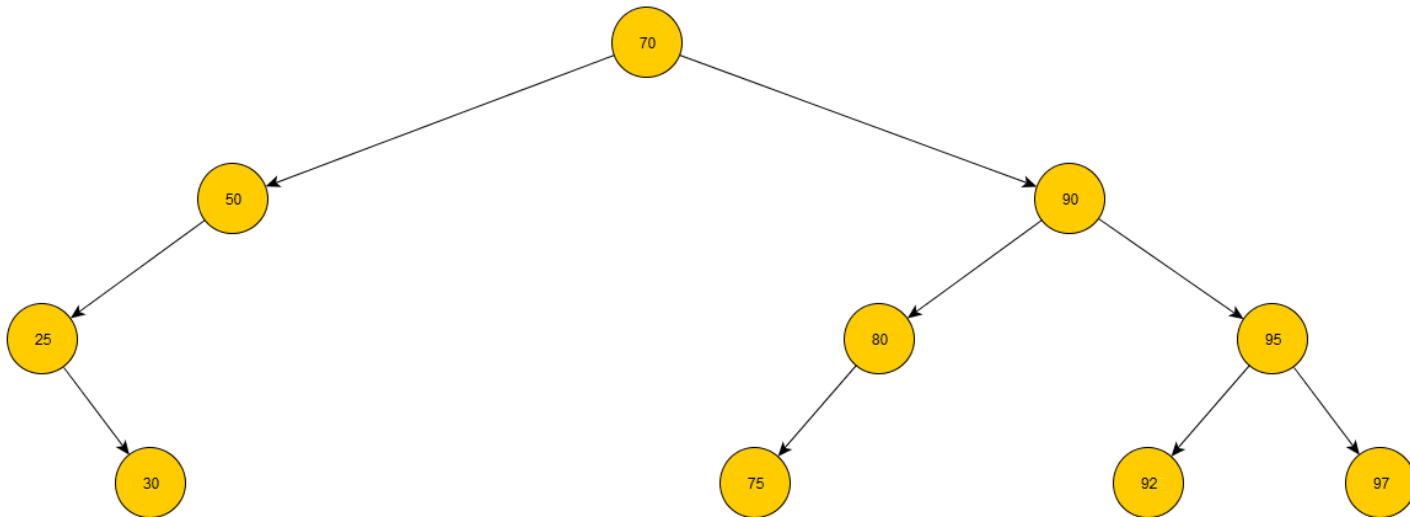
- Fixed Layout



# Extreme Programming

## Homework – Flexible Layout for Binary Tree

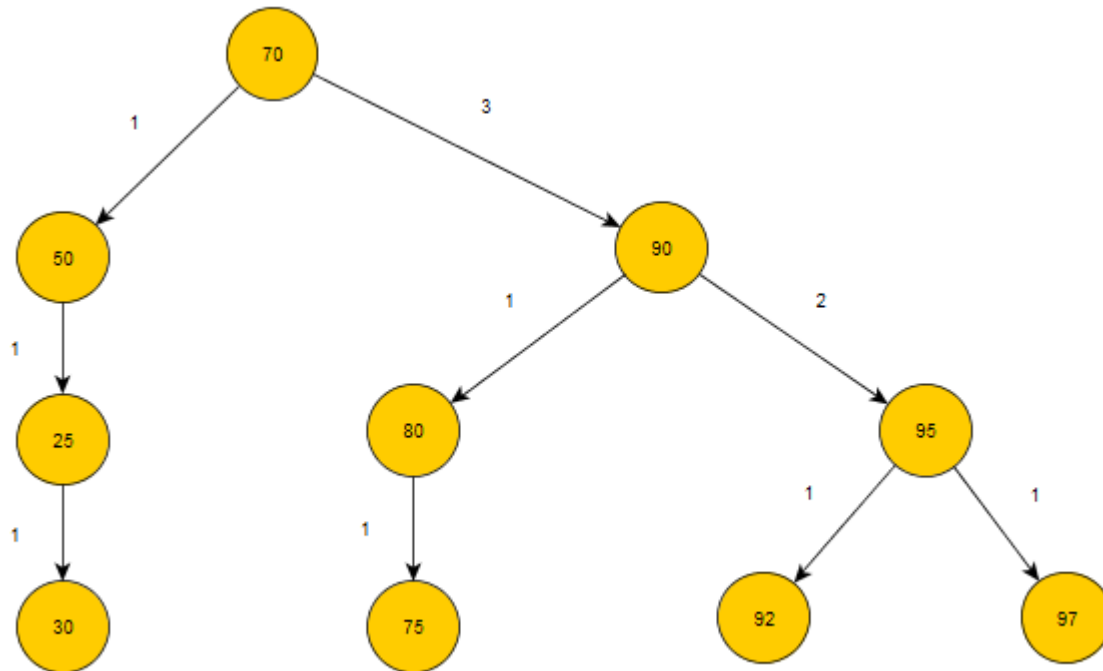
- Fixed Layout
  - Tree Height / Layer-depth determines the layout



# Extreme Programming

## Homework – Flexible Layout for Binary Tree

- Flexible Layout
  - Sub-tree width determines the layout



# Assignment 10

## Send me an email

- Email: [jakub.gemrot@gmail.com](mailto:jakub.gemrot@gmail.com)
- Subject: **Programming II – 2015 – Assignment 10**
- Zip up the whole project and send it
- You WILL NOT find the assignment in CoDex!
- Deadline:
  - 4.5.2015 23:59
- Points: 10 + 5 (meeting the deadling)

# Questions?

I sense a soul in search of answers...

- Sadly, I do not own the patent for perfection (and will never do)
- In case of doubts about the assignment or some other problems don't hesitate to contact me!
  - Jakub Gemrot
    - [gemrot@gamedev.cuni.cz](mailto:gemrot@gamedev.cuni.cz)