

# Extreme Programming The Secret Sauce for High-speed, Quality Software

---

Paige Watson

[paige.watson@outlook.com](mailto:paige.watson@outlook.com)

```
mirror_mod = modifier_ob.mirror
#set mirror object to mirror
mirror_mod.mirror_object = mirror_ob

operation == "MIRROR_X":
    mirror_mod.use_x = True
    mirror_mod.use_y = False
    mirror_mod.use_z = False
operation == "MIRROR_Y":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

#selection at the end -add
mirror_ob.select= 1
modifier_ob.select=1
context.scene.objects.active = mirror_ob
print("Selected" + str(modifier_ob.name))
mirror_ob.select = 0
#set mirror object to mirror
mirror_mod.mirror_object = mirror_ob
print("please select exactly one mirror object")

-- OPERATOR CLASSES -----

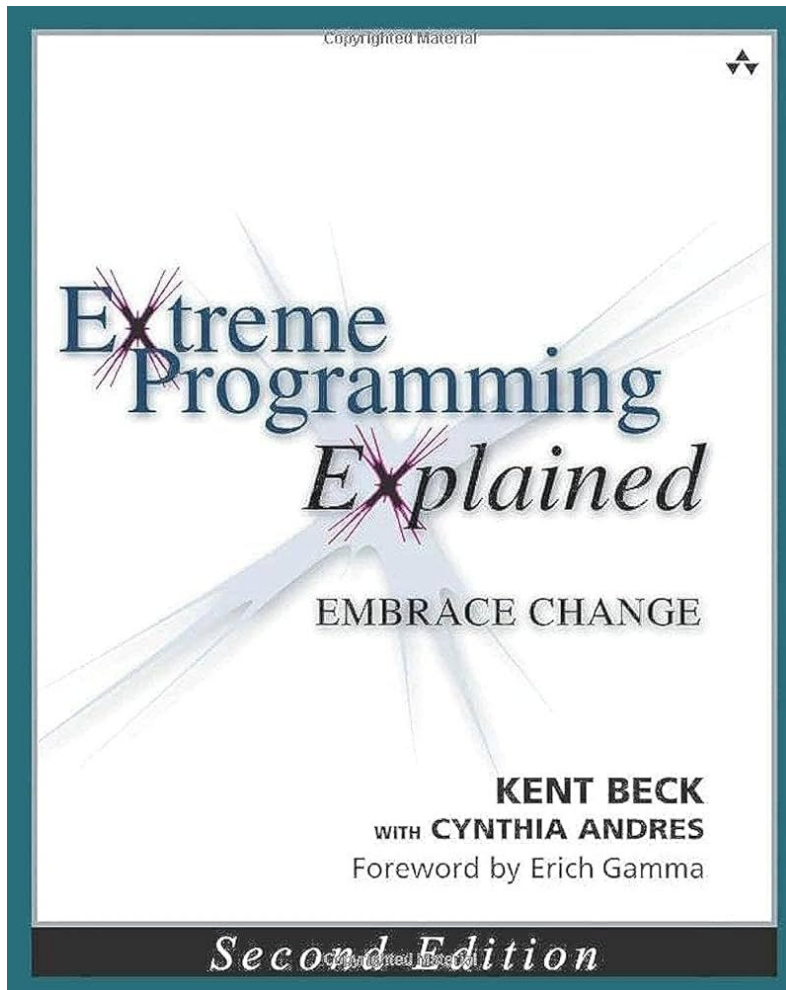
bpy.types.Operator):
    """Set X mirror to the selected object.mirror_mirror_x"""
    bl_label = "Mirror X"

    def execute(self, context):
        if context.active_object is not None:
```

# Introduction to Extreme Programming

---

BRIEF OVERVIEW OF XP,  
AND KEY PRINCIPLES.



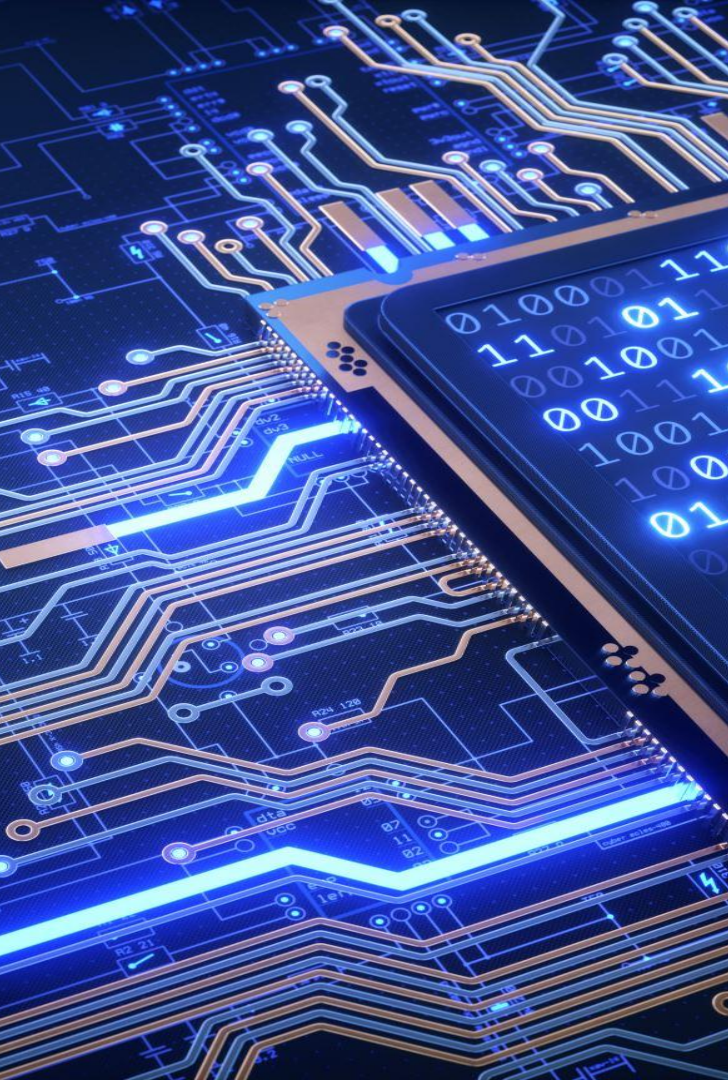
# Extreme Programming Explained

---

Kent Beck - 2000

CHALLENGES IN  
SOFTWARE  
DEVELOPMENT  
THAT XP  
ADDRESSES.

# The Need for Speed and Quality in Software Development

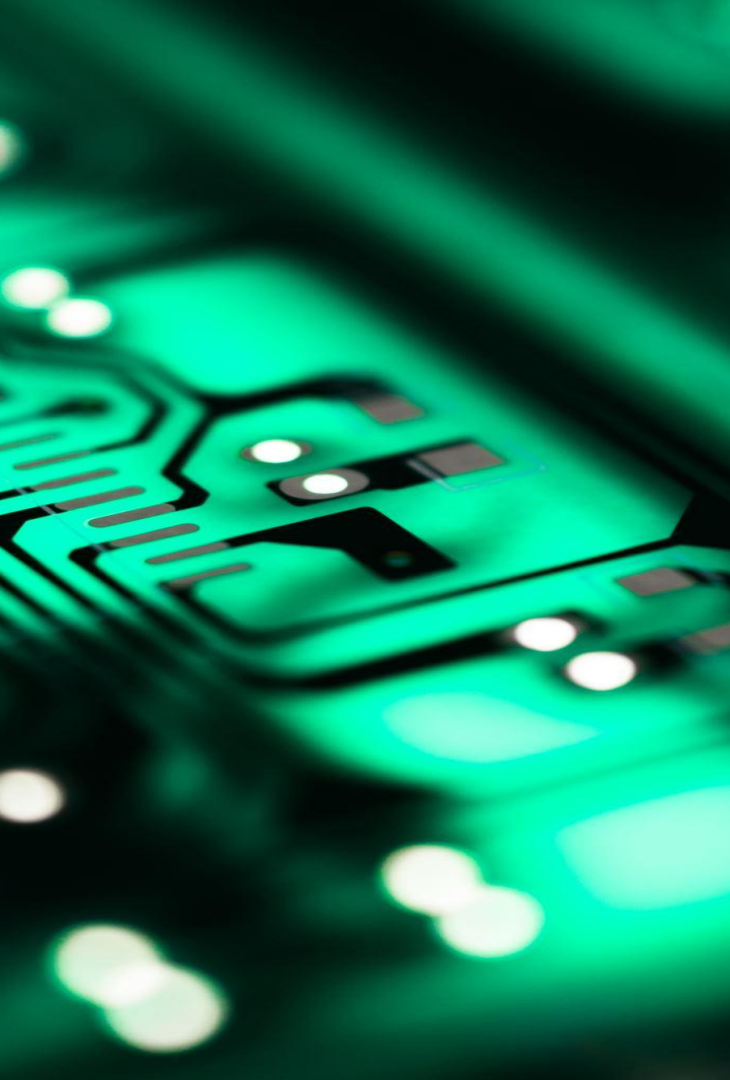


# The Core Practices of XP

---

OVERVIEW OF PRACTICES LIKE  
PAIR PROGRAMMING,  
TEST-DRIVEN DEVELOPMENT,  
CONTINUOUS INTEGRATION.

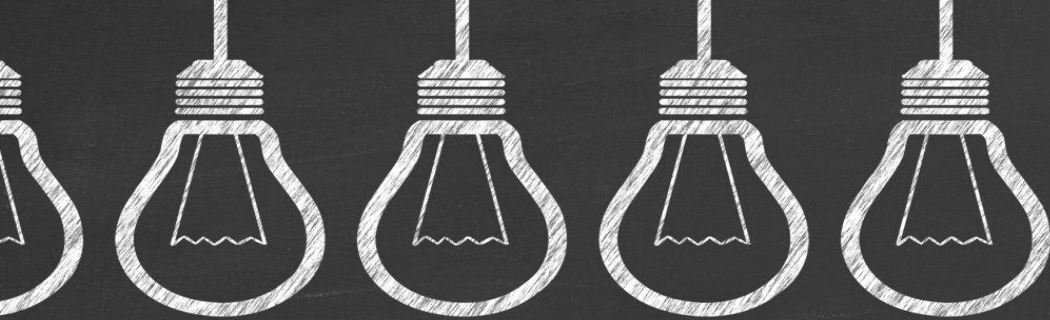




# Pair Programming: Two Heads are Better than One

---

BENEFITS OF  
COLLABORATIVE PROGRAMMING  
(PAIR/ENSEMBLE/WHOLE TEAM)



```
1 describe('rolling stock collection', () => {
2   let collection: RollingStockCollection;
3   describe('when empty', () => {
4     beforeEach(() => {
5       collection = new RollingStockCollection([]);
6     });
7
8     it('returns count 0', () => {
9       expect(collection).toHaveItemCount(0);
10    });
11    it('is empty', () => {
12      expect(collection).toBeEmpty();
13    });
14  });
15 }
```

# Test-Driven Development: Ensuring Quality from the Start

THE IMPACT OF TDD ON SOFTWARE  
QUALITY

TRUSTING YOUR CODE



# Continuous Integration: Keeping the Software Healthy

---

AUTOMATED TESTING  
FASTER RELEASE CYCLES  
ENHANCED CODE QUALITY





# XP Teams: Faster, Better, Happier

---

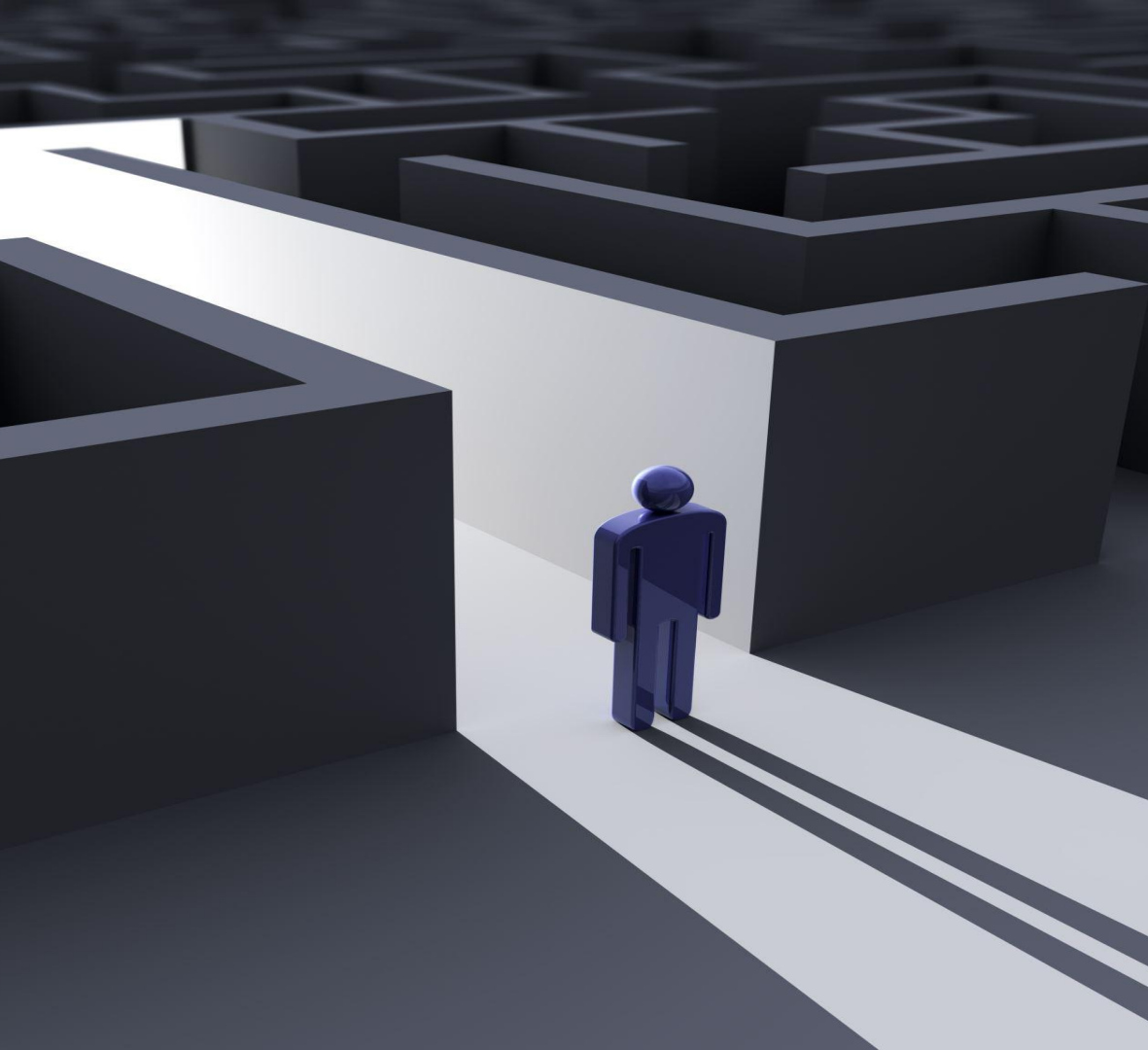
TEAM MORALE  
TEAM EMPOWERMENT  
REDUCED FEAR  
CLARITY AND FOCUS



# Real-world Success Stories

HEALTHCARE INSURANCE SURVEY  
ENGINE

POINT OF SALE APP



# Common Misconceptions and Challenges in XP

ADDRESSING MYTHS:

XP IS ONLY ABOUT CODING

PAIR PROGRAMMING IS INEFFICIENT

XP TEAMS DON'T PLAN OR DESIGN

CONSTANT REFACTORING LEADS TO  
DELAY

RESISTANCE TO CULTURAL CHANGE\*\*

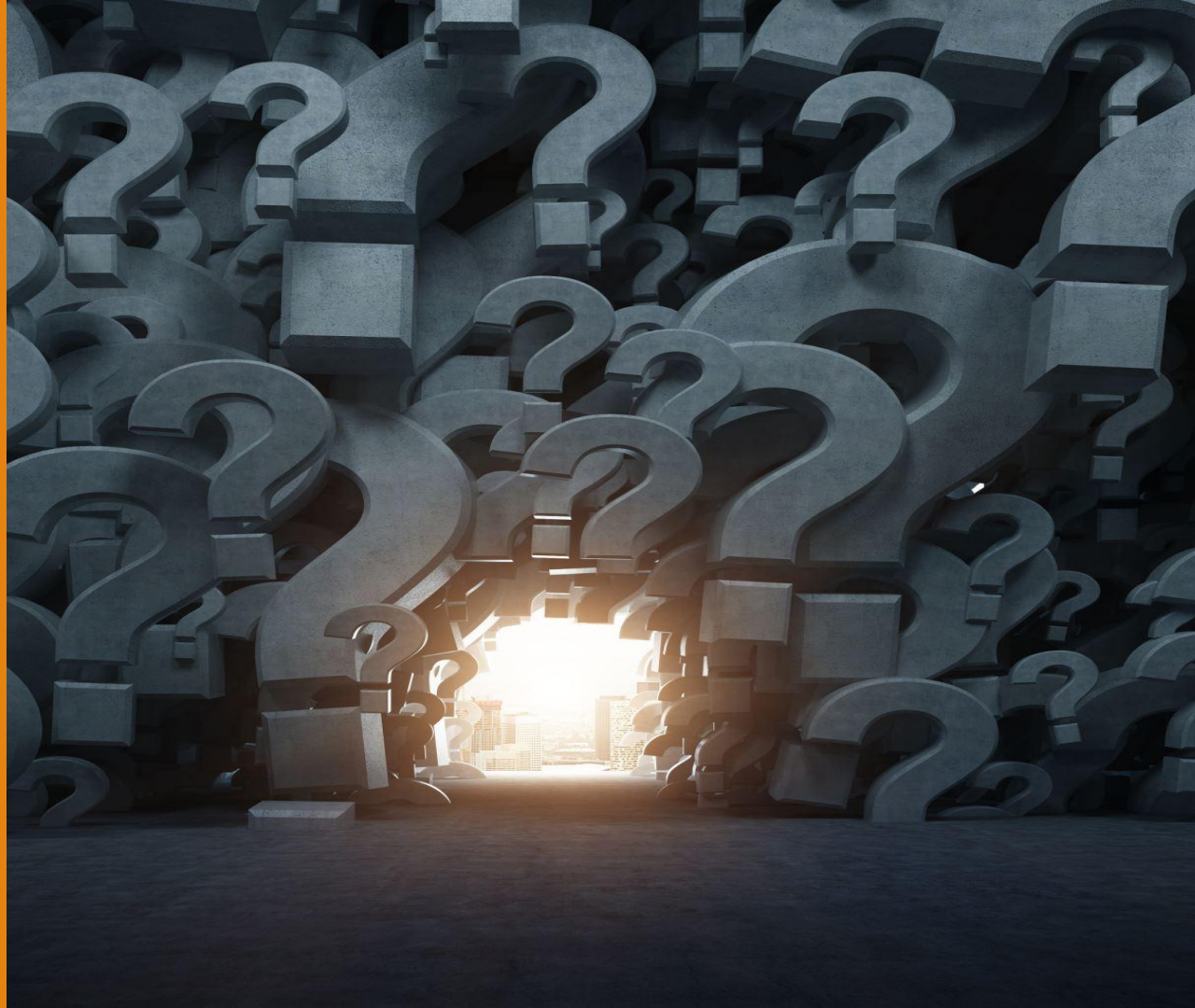
# Embrace the Extreme for Excellence

QUALITY PRACTICES CREATE QUALITY  
SOFTWARE

MORE RAPID, LESS BUGS, MORE  
EMPOWERING



Q&A





# Paige Watson

**Email:**  
[paige.watson@outlook.com](mailto:paige.watson@outlook.com)

**LinkedIn:**  
<https://www.linkedin.com/in/paige-is-xp/>

**Calendly:**  
<https://calendly.com/paige-watson>

**GitHub:**  
<https://github.com/MyTurnyet/Talks>

