Sprint0 Presentation

2023 Spring, SWPP

8조

강병준
김민주
양현서
이상원

CONTENTS

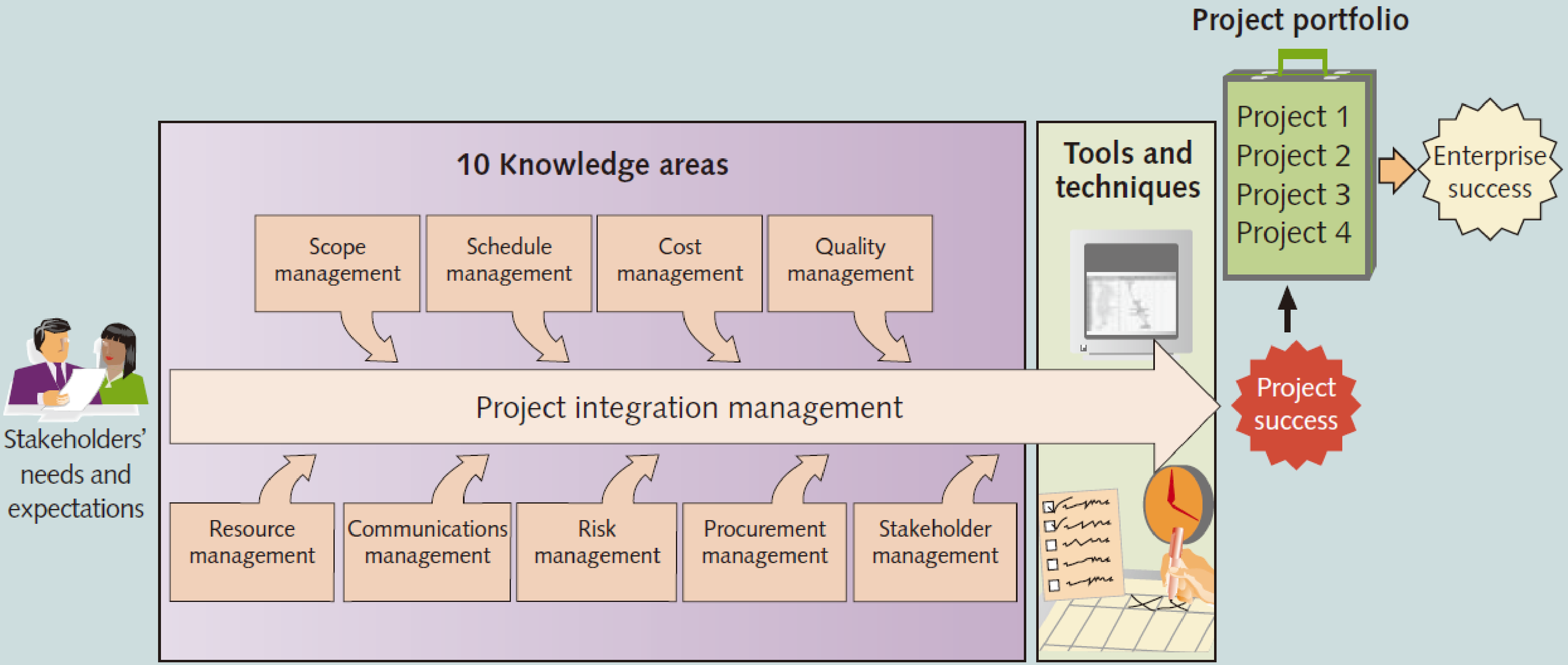
INTRODUCTION

01 Team

02 Teammates: **강병준, 김민주, 양현서, 이상원**

03 Reason for choosing this class

EXPECTATION



OPTIMIZATION

01 Dead Code Elimination

```
define i32 @foo(i32 %x, i32 %y) {
entry:
%a = add i32 %x, 5
%b = mul i32 %a, %y
%c = add i32 %a, %b
%d = mul i32 %a, %a
%e = mul i32 %b, %b
%f = add i32 %d, %e
ret i32 %f
}
```

```
define i32 @foo(i32 %x, i32 %y) {
entry:
%a = add i32 %x, 5
%b = mul i32 %a, %y
%d = mul i32 %a, %a
%e = mul i32 %b, %b
%f = add i32 %d, %e
ret i32 %f
}
```

OPTIMIZATION

02 Constant Propagation

```
define i32 @main() {
%a = alloca i32
store i32 5, i32* %a
%b = load i32, i32* %a
%c = add i32 %b, 7
ret i32 %c
}
```

```
define i32 @main() {
ret i32 12
}
```

OPTIMIZATION

03 Loop Unrolling

```
define void @loop_without_op() {
entry:
  %i = alloca i32
  store i32 0, i32* %i
  br label %loop
loop:
  %i_val = load i32, i32* %i
  %cond = icmp slt i32 %i_val, 3
  br i1 %cond, label %body, label %end
body:
  %i_val_inc = add nsw i32 %i_val, 1
  store i32 %i_val_inc, i32* %i
  br label %loop
end:
  ret void
}
```

```
define void @loop_without_op() {
entry:
  %i = alloca i32
  store i32 0, i32* %i
  br label %loop
loop:
  %i_val = load i32, i32* %i
  %cond = icmp slt i32 %i_val, 3
  br i1 %cond, label %body, label %end
body:
  %i_val_inc = add nsw i32 %i_val, 1
  store i32 %i_val_inc, i32* %i
  br label %loop
end:
  ret void
} | define void @loop_unrolled() {
entry:
  %i = alloca i32
  store i32 0, i32* %i
  br label %loop
unroll:
  %i_val_1 = add nsw i32 %i_val, 1
  store i32 %i_val_1, i32* %i
  %i_val_2 = add nsw i32 %i_val_1, 1
  store i32 %i_val_2, i32* %i
  %i_val_3 = add nsw i32 %i_val_2, 1
  store i32 %i_val_3, i32* %i
end:
  ret void
}
```

OPTIMIZATION

04 Loop Termination (Reduce Branch Penalty)

```
define void @loop_without_op() {
entry:
%i = alloca i32
store i32 0, i32* %i
br label %loop

loop:
%i_val = load i32, i32* %i
%cond = icmp slt i32 %i_val, 3
br i1 %cond, label %body, label %end

body:
%i_val_inc = add nsw i32 %i_val, 1
store i32 %i_val_inc, i32* %i
br label %loop

end:
ret void
}
```

```
define void @loop_branch_op() {
entry:
%i = alloca i32
store i32 0, i32* %i
br label %loop

loop:
%i_val = load i32, i32* %i
%cond = icmp sgt i32 %i_val, 3
br i1 %cond, label %end, label %body

body:
%i_val_inc = add nsw i32 %i_val, 1
store i32 %i_val_inc, i32* %i
br label %loop

end:
ret void
}
```

OPTIMIZATION

05 Load-Use Hazard

```
define i32 @calc(i32* %p1, i32* %p2, i32* %p3) {
entry:
%val1 = load i32, i32* %p1
%mul = mul i32 %val1, 8
%val2 = load i32, i32* %p2
%val3 = load i32, i32* %p3
%sub = sub i32 %mul, %val2
%result = sub i32 %sub, %val3
ret i32 %result
}
```

```
define i32 @calc(i32* %p1, i32* %p2, i32* %p3) {
entry:
%val1 = aload i32, i32* %p1
%val3 = aload i32, i32* %p3 ;cost 1
%val2 = load i32, i32* %p2    ;cost 20
%mul = mul i32 %val1, 8   ;cost 1
%sub = sub i32 %mul, %val2  ;cost 5
%result = sub i32 %sub, %val3
ret i32 %result
}
```

OPTIMIZATION

06 Add / Sub <-> Inc / Dec

```
%a_val = load i32, i32* %a
%sum = add nsw i32 %a_val, 3
store i32 %sum, i32* %a
```

```
%a_val = load i32, i32* %a
%sum = incr i32 %a_val
%sum = incr i32 %a_val
%sum = incr i32 %a_val
store i32 %sum, i32* %a
```

OPTIMIZATION

07 Shift -> Mul / Div

```
%a_val = load i32, i32* %a
%shifted = shl i32 %a_val, 3
store i32 %shifted, i32* %a
```

```
%a_val = load i32, i32* %a
%product = mul i32 %a_val, 8
store i32 %product, i32* %a
```

Q & A