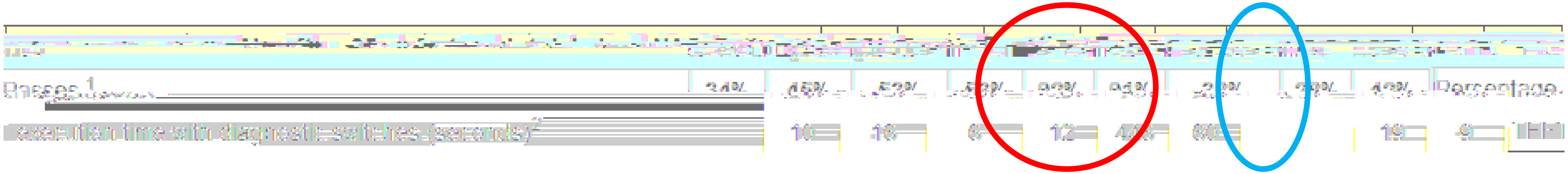
Verification of Fortran Codes

Verification Features of Fortran Compilers

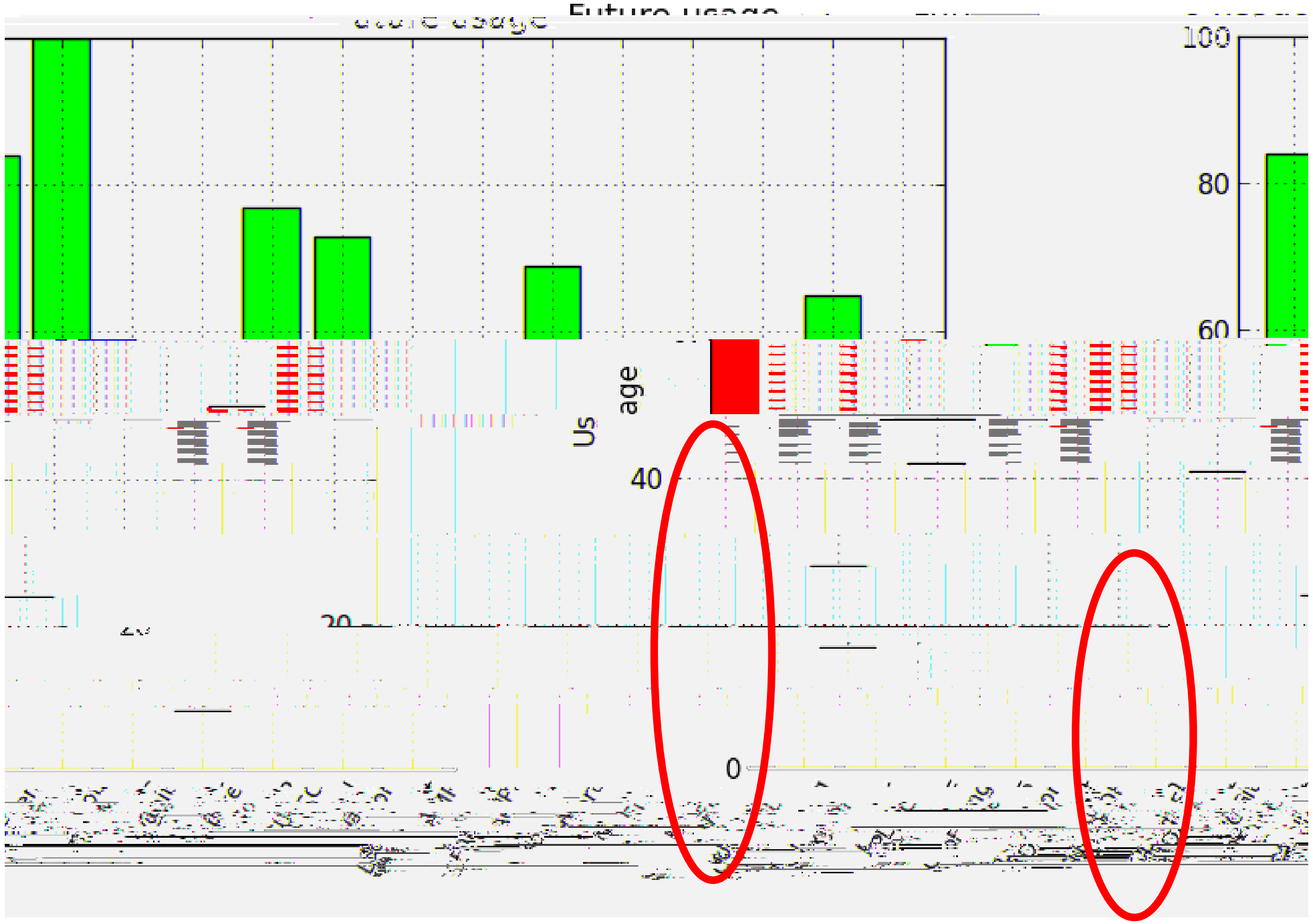


Passes[1]	34%	45%	53%	53%	93%	94%	23%	[illegible]	43%	Percentage
execution time with diagnostic switches (seconds)[2]	10	16	6	12	48	60		19	9	[illegible]

Usage of Verification Tools

Only 11 (7%) out of 155 Fortran developers are using verification tools;

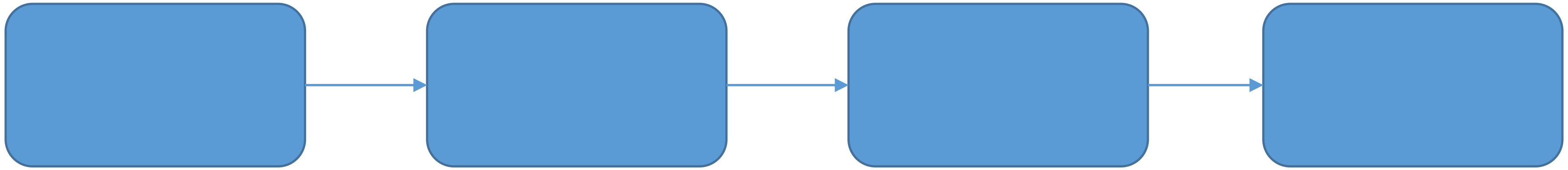
What Interests Fortran Programmers?



Fortran Verification Workflow (1)

```
nagfor -C=all -C=undefined -info -g -gline
```

Fortran Verification Workflow (2)



Fortran Verification Tools

Fortran Array Bug

```
real, dimension(3) :: eng, aero
do i = 1, 3 ! 1 = port, 2 = centre, 3 = starboard
  aero = eng(i)
end do
! modern and correct version
aero(:) = eng(:)
```

Precision Bugs (1)

```
real(kind=REAL32) :: a, geom, v, g_p
a = geom * v ** (2/3) ! calculate surface area
g_p = 6.70711E-52

real(kind=REAL64) :: theta
real(kind=REAL32) :: x
x = 100.0_REAL64 * cos( theta )
```

Precision Bugs (2)

```
real(kind=REAL64) :: d
real(kind=REAL32) :: x, y
d = sqrt( x**2 + y**2 )
```

Forcheck Dummy Argument Checking

```
subroutine foo( a, b )
  real :: a
  real, optional :: b
```

Forcheck Dummy Argument Intent Checking

intent

```
forchk -intent arg_test.f90
```

```
B
**[870 I] dummy argument has no INTENT attribute
         (INTENT(IN) could be specified)
```

Forcheck

Runtime Checking

NAG Compiler Optional Argument Detection

```
nagfor -C=present arg_test.f90 -o arg_test.exe
```

```
call foo( a )
subroutine foo( a, b )
  real, intent(out) :: a
  real, intent(in), optional :: b
  a = b**2
end subroutine foo
```

```
Runtime Error: arg_test.f90, line 14: Reference to OPTIONAL
argument B which is not PRESENT
```

NAG Compiler Dangling Pointer Detection

```
nagfor -C=dangling p_check.f90 -o p_check.exe
```

```
real, dimension(:), allocatable, target :: vec
real, dimension(:), pointer :: vec_p

allocate( vec(1:100) )
vec_p => vec; deallocate( vec )
print *, vec_p(:)
```

```
Runtime Error: p_check.f90, line 12: Reference to dangling pointer
VEC_P
Target was DEALLOCATEd at line 10 of pointer_check.f90
```

NAG Compiler Undefined Variable Detection

```
nagfor -C=undefined undef_test.f90 -o undef_test.exe
```

```
real, dimension(1:11) :: array
array(1:10) = 1.0
print *, array(1:11)
```

```
Runtime Error: undef_test.f90, line 7: Reference to
undefined variable ARRAY(1:11)
Program terminated by fatal error
```

NAG Compiler Procedure Argument Detection

```
nagfor -C=calls sub1.f90 -o sub1.exe
```

```fortran
integer, parameter :: x = 12
call sub_test( x )
subroutine sub_test( x )
  integer :: x
  x = 10
end subroutine sub_test
```

```
Runtime Error: sub1.f90, line 13: Dummy argument X is
associated with an expression - cannot assign
```

NAG Compiler Integer Overflow Detection

```
nagfor -C=intovf
```

Conclusion

Verification as a service?