Craft and Software Engineering

Glenn Vanderburg
InfoEther
glenn@infoether.com
@glv

Software Engineering, Software Craftsmanship

Management
vs.
Programmers?

A Caricature of Engineering

1. A software system can best be designed if the testing is interlaced with the designing instead of being used after the design.

2. A simulation which matches the requirements contains the control which organizes the design of the system.

3. Through successive repetitions of this process of interlaced testing and design the model ultimately becomes the software system itself. [...] in effect the testing and the replacement of simulations with modules that are deeper and more detailed goes on with the simulation model controlling, as it were, the place and order in which these things are done.

Unlike the first conference, at which it was fully accepted that the term software engineering expressed a need rather than a reality, in Rome there was already a slight tendency to talk as if the subject already existed.

And it became clear during the conference that the organizers had a hidden agenda, namely that of persuading NATO to fund the setting up of an International Software Engineering Institute.

However things did not go according to their plan. The discussion sessions which were meant to provide evidence of strong and extensive support for this proposal were instead marked by considerable scepticism, and led one of the participants, Tom Simpson of IBM, to write a splendid short satire on “Masterpiece Engineering”.

It was little surprise to any of the participants in the Rome conference that no attempt was made to continue the NATO conference series, but the software engineering bandwagon began to roll as many people started to use the term to describe their work, to my mind often with very little justification.

—*Brian Randell*

“Premature maturity”

[Programming] is not some kind of engineering where all we have to do is put something in one end and turn the crank.

—*Bruce Eckel*

"A Rational Design Process"

A. Establish and document requirements

B. Design and document the module structure

C. Design and document the module interfaces

D. Design and document the uses hierarchy

E. Design and document the module internal structures

F. Write programs

G. Maintain

The conversion of an idea to an artifact, which engages both the designer and the maker, is a complex and subtle process that will always be far closer to art than to science.

(Eugene S. Ferguson, Engineering and the Mind's Eye)

In engineering ... people design through documentation.

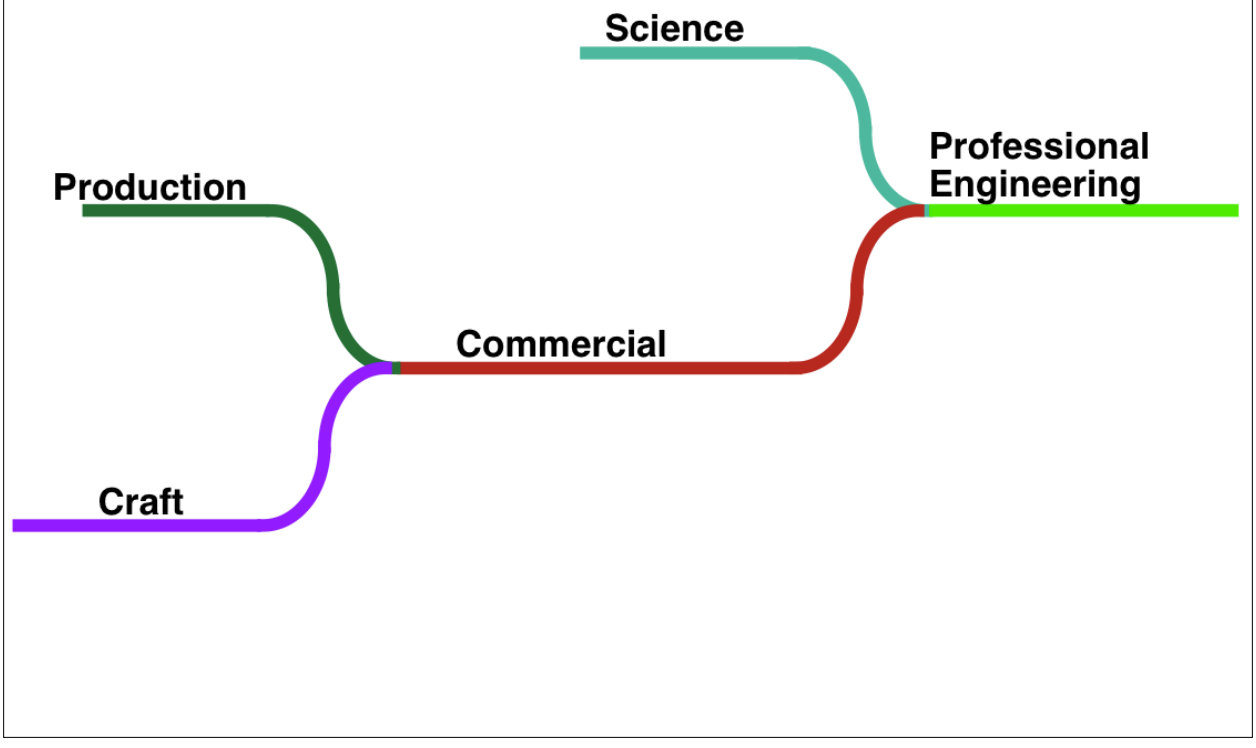
—David Parnas

Although the drawings appear to be exact and unequivocal, their precision conceals many informal choices, inarticulate judgments, acts of intuition, and assumptions about the way the world works.

(Eugene S. Ferguson, Engineering and the Mind's Eye)

The *defined process control model* requires that every piece of work be completely understood. A defined process can be started and allowed to run until completion, with the same results every time.

The *empirical process control model* provides and exercises control through frequent inspection and adaptation for processes that are imperfectly defined and generate unpredictable and unrepeatable outputs.



@skilldrick
Nick Morgan

I've invented a new term "Parasitic Credibility". Where certain fields attempt to link themselves to science in order to sound more real.

Aeroplanes are not designed by science, but by art, in spite of some pretence and humbug to the contrary. [...]

There is a big gap between scientific research and the engineering product which has to be bridged by the art of the engineer.

—J. D. North

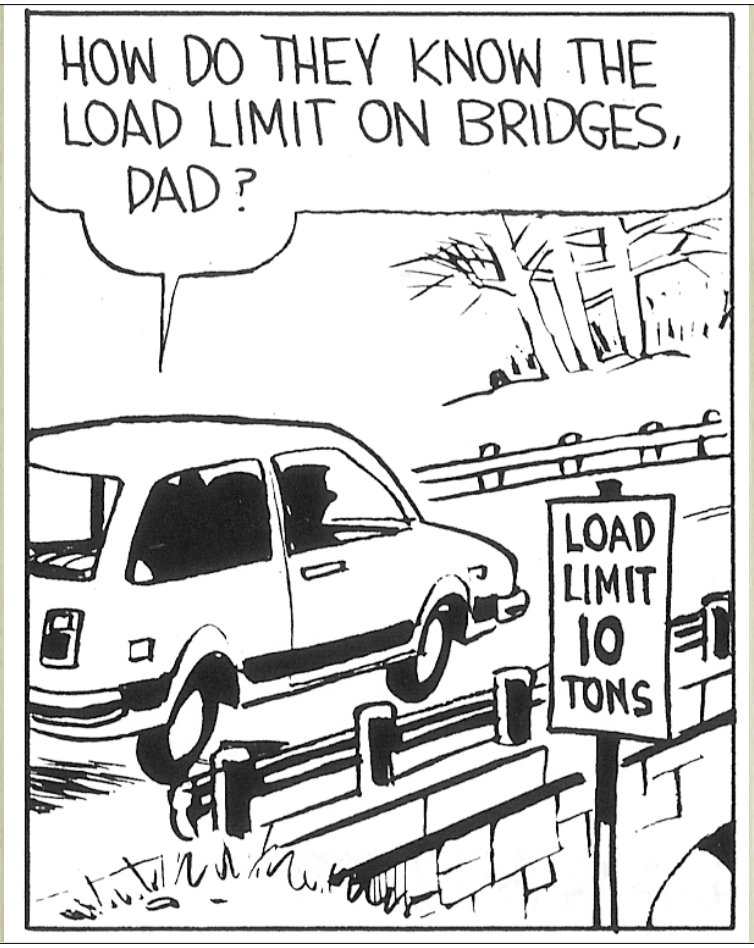
“You don’t know it’s right if you don’t have the math to prove it.”

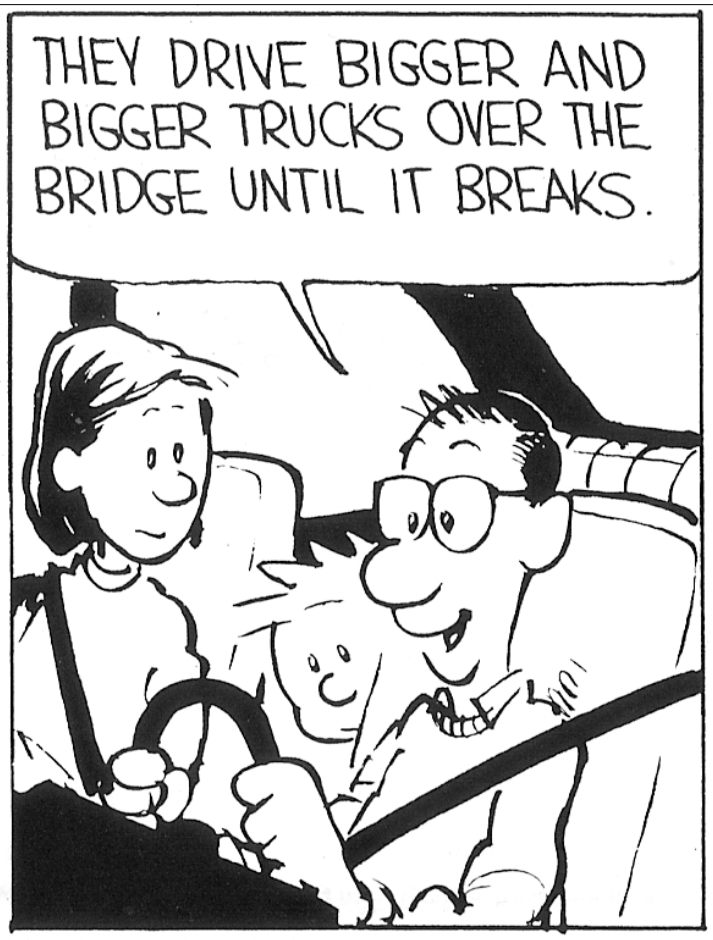
Structural analyses (indeed, any engineering calculations) must be employed with caution and judgment, because mathematical models are always less complex than actual structures, processes, or machines.

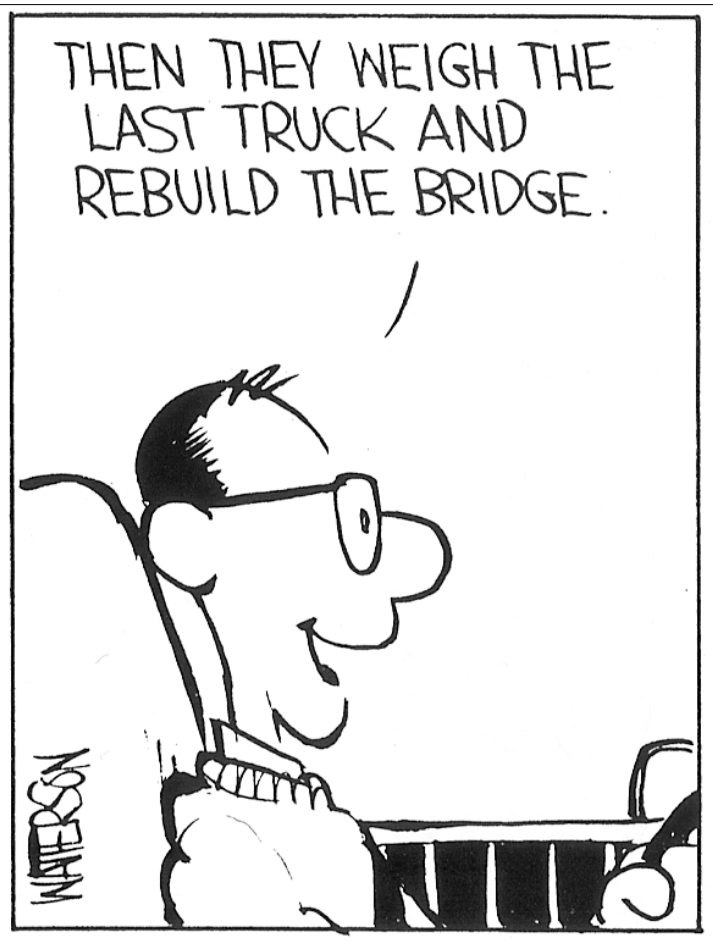
(Eugene S. Ferguson, Engineering and the Mind’s Eye)

Engineering is not the art of constructing. It is rather the art of not constructing: or, it is the art of doing well with one dollar what any bungler can do with two.

—*Arthur Mellen Wellington*




THEY DRIVE BIGGER AND BIGGER TRUCKS OVER THE BRIDGE UNTIL IT BREAKS.


THEN THEY WEIGH THE LAST TRUCK AND REBUILD THE BRIDGE.
WATTERSON


OH. I SHOULD'VE GUESSED.
DEAR, IF YOU DON'T KNOW THE ANSWER, JUST TELL HIM!

Mathematical modeling was introduced as a cost-saving measure.

Engineering is the art of directing the great sources of power in nature for the use and convenience of man.

—*Institution of Civil Engineers*

Structural engineering is
the science and art
of designing and making,
with economy and elegance,
[...] structures so that they
can safely resist the forces to
which they may be subjected.

—*Structural Engineer's Association*

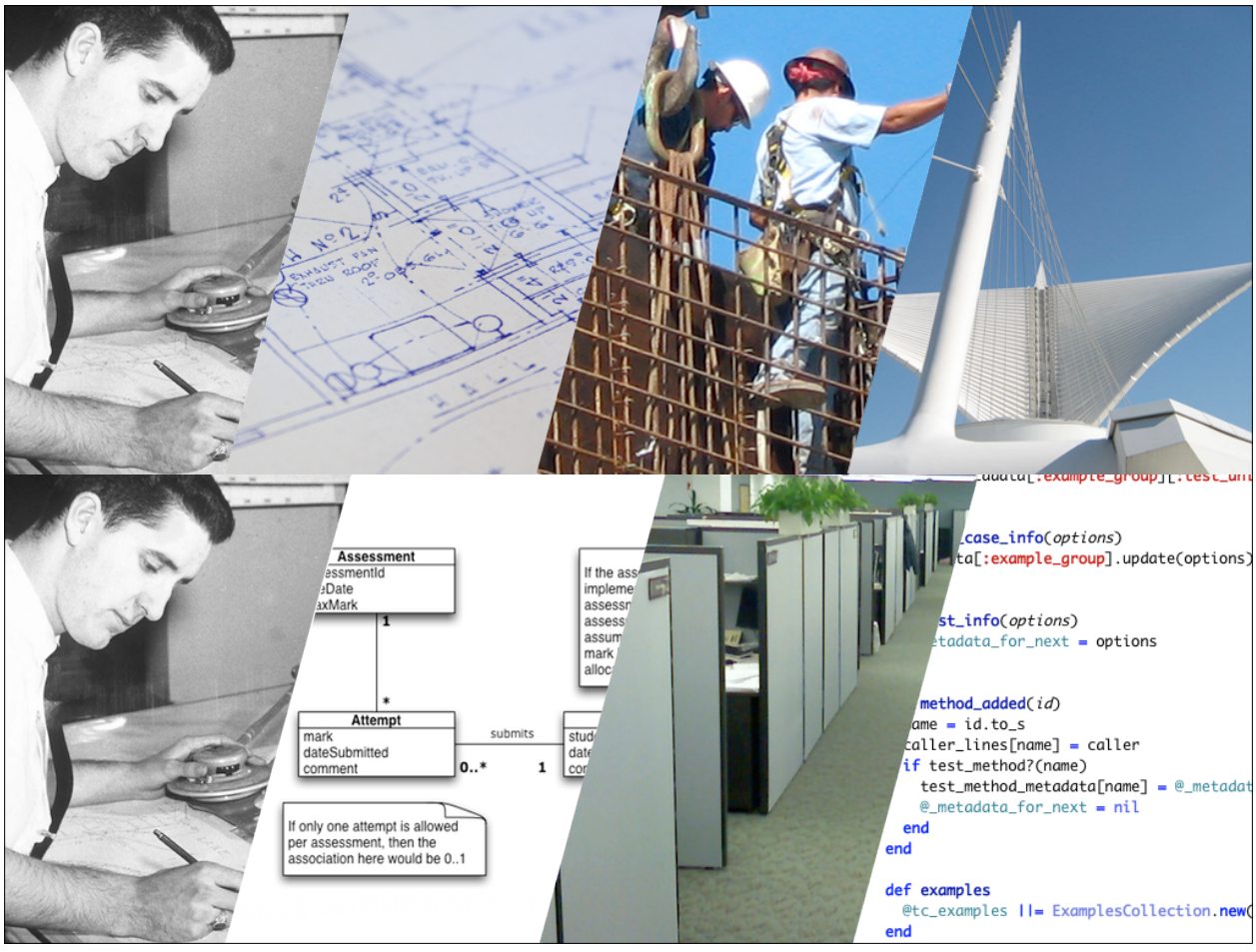
Different Engineering Disciplines are *Different*

- Different materials, physical effects, forces
- Different degrees of complexity in requirements, designs, processes, and artifacts
- Varied reliance on formal modeling and analysis vs. experimentation, prototyping, and testing
- Varied use of defined and empirical processes

Real
Software Engineering

Software engineering is
the science and art
of designing and making,
with economy and elegance,
[...] systems so that they can
readily adapt to the situations to
which they may be subjected.

Software engineering will be *different* from other kinds of engineering.



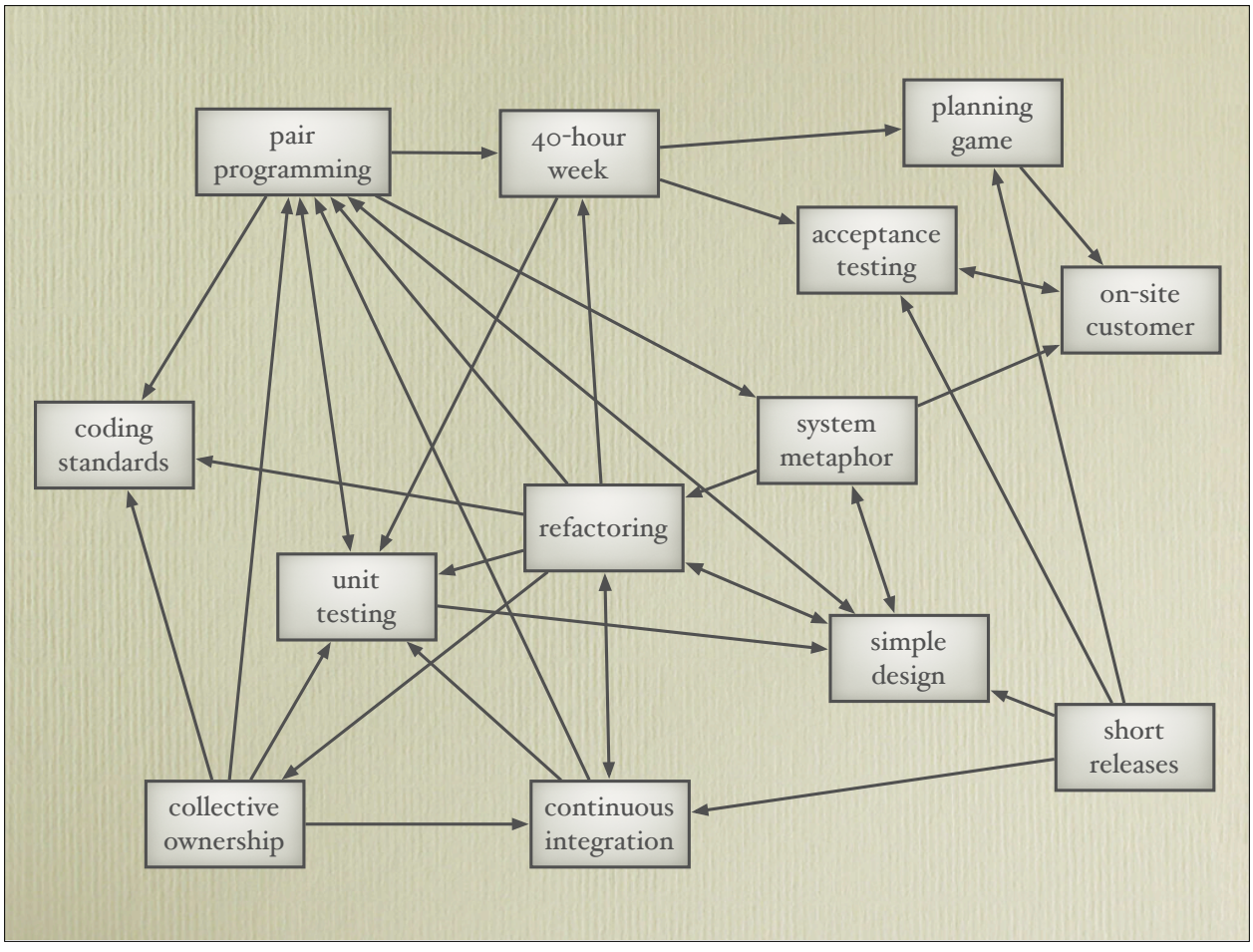
divided by	2	3	4	7
9	4.5	3	2.5	1.29
10	5.0	3.33	2.5	1.43
11	5.5	3.66	2.75	1.57
12.6	6.3	4.2	3.15	1.8
22	11.0	7.33	5.5	3.14
100	50.0	33.33	25.0	14.29

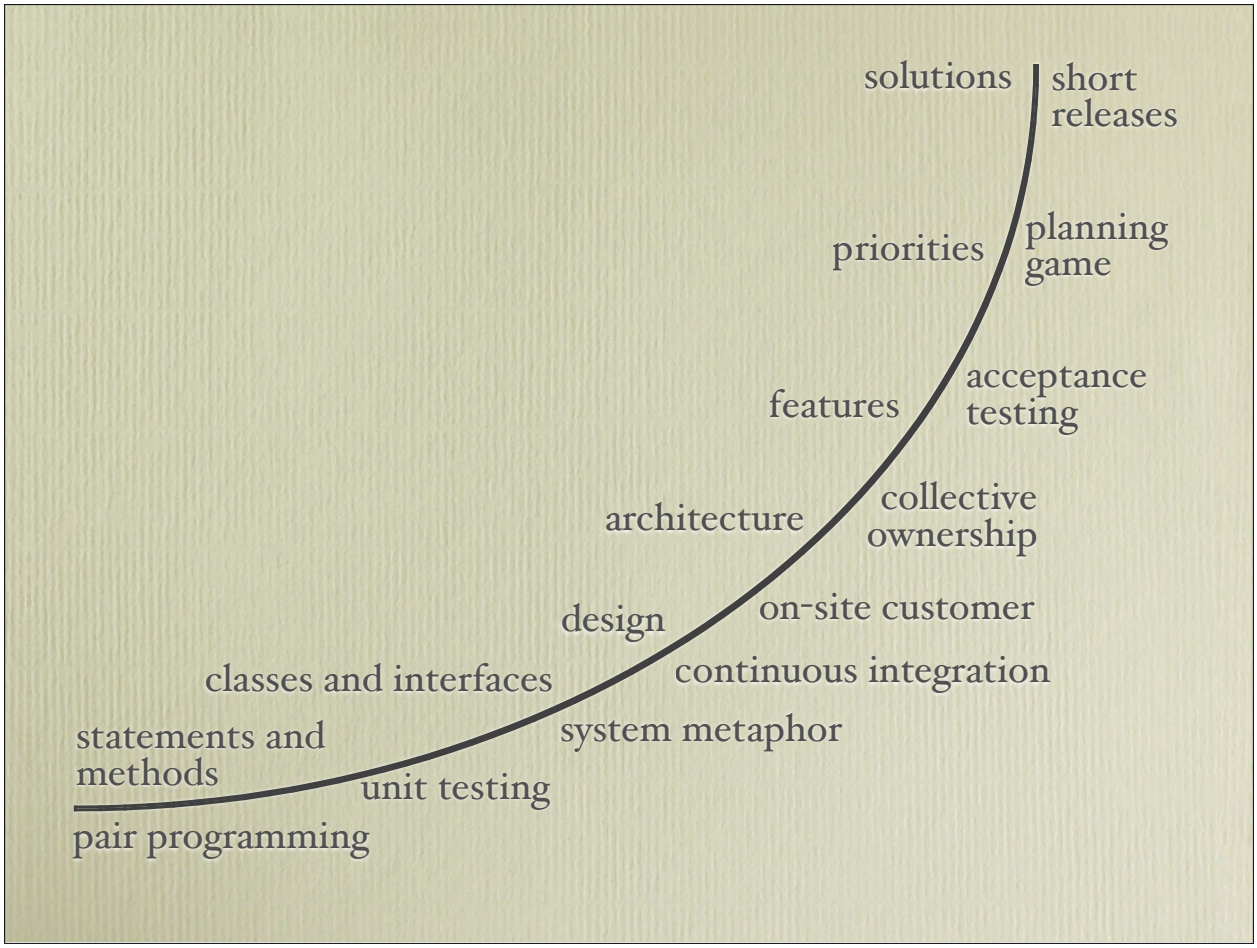
```
Feature: Addition
  In order to avoid silly mistak
  As an error-prone person
  I want to divide two numbers

  Scenario Outline: Divide two nu
    Given I have entered <input_
    And I have entered <input_2>
    When I press "divide"
    Then the result should be <r

  Examples:
    | input_1 | input_2 | result  |
    | 10      | 2       | 5.0     |
    | 12.6    | 3       | 4.2     |
    | 22      | 7       | ~=3.14  |
    | 9       | 3       | <5      |
    | 11      | 2       | 4<_<6   |
    | 100     | 4       | 32      |
```

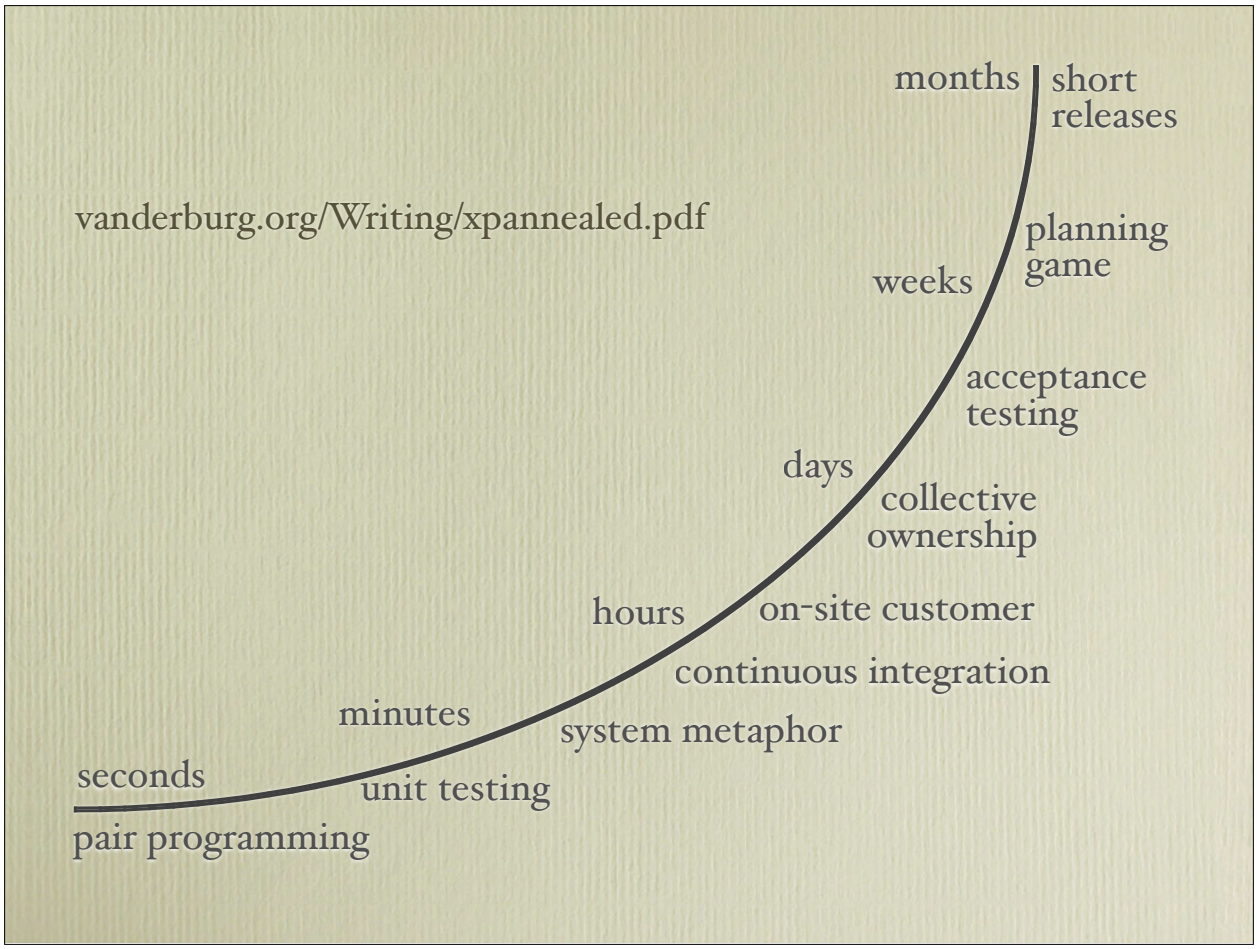
eg.Division		
numerator	denominator	quotient?
10	2	5.0
12.6	3	4.2
22	7	~=3.14
9	3	<5
11	2	4<_<6
100	4	32 expected / 25 actual

```
    assert_in_delta  3.1
    assert           5
    assert           4..
    assert_equal     32,
  end
end
```

```
ers" do
 do
 should == 5.0
 should == 4.2
 should be_close(3.14, 0.01)
  (9 / 3)    .should < 5
  (11/2)     .should satisfy{|n| n > 4 && n < 6 }
  (100 / 4) .should == 32
 end
end
```




solutions
short releases
priorities
planning game
features
acceptance testing
architecture
collective ownership
design
on-site customer
classes and interfaces
continuous integration
statements and methods
system metaphor
unit testing
pair programming


months
short releases
vanderburg.org/Writing/xpannealed.pdf
planning game
weeks
acceptance testing
days
collective ownership
hours
on-site customer
continuous integration
minutes
system metaphor
seconds
unit testing
pair programming

Assumptions Once True, But No Longer

- Code is hard to read
- Code is hard to change
- Testing is expensive

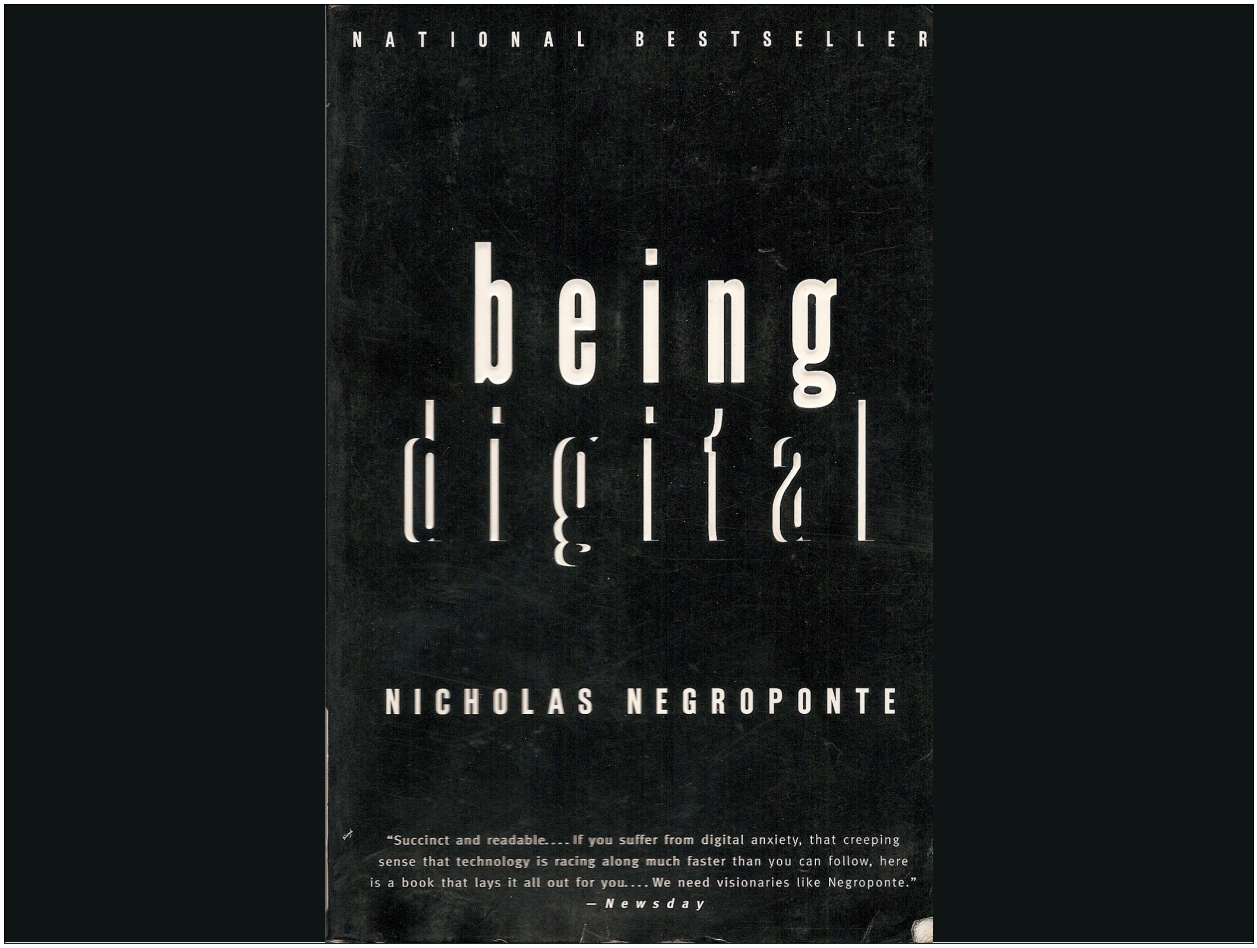
Assumptions Once Believed But Never True

- All engineering is like structural engineering
- Programming is like building
- Modeling and analysis are about correctness

The Reality of Software Engineering

- Software is very unlike bridges and buildings.
- Additional complexity hinders requirements, design, and approval.
- Source code *is* a model.
- Building and testing our interim designs is effectively free.
- Empirical processes are rational for software.

Software Engineering and Craft



Design by Artisans

- Artisans may produce documents to help themselves think.
- But they build what is in their heads.

Design by Engineers

- Engineers produce documents to help themselves think.
- But they mostly produce documents to convey the design to builders.

Bending moments in nodes (kNm)			Displacements (mm)
Vehicle dynamics			
Node 9	Node 17	Node 25	
1,042.48	− 585.56	1,076.93	2.29
Earthquake through support acceleration – accelerogram from EC8			
Node 8	Node 17	Node 26	
− 721.62	1,201.01	− 722.39	1.422
Spectral analysis			
Node 7	Node 17	Node 27	
− 153.62	228.42	− 151.14	0.29

EQUATIONS

BEAM SHEAR (ONE-WAY)

Reinforced

$\phi V_c = \phi 2bd\sqrt{f'_c}$

$\phi = 0.75$

plain

$\phi V_n = \phi \frac{4}{3} b h \sqrt{f'_c}$

$\phi = 0.55$

PERIPHERAL = PUNCHING = 2 WAY SHEAR

Reinf'd

$\phi V_c = \phi 4 b_o d \sqrt{f'_c}$

$\phi = 0.75$

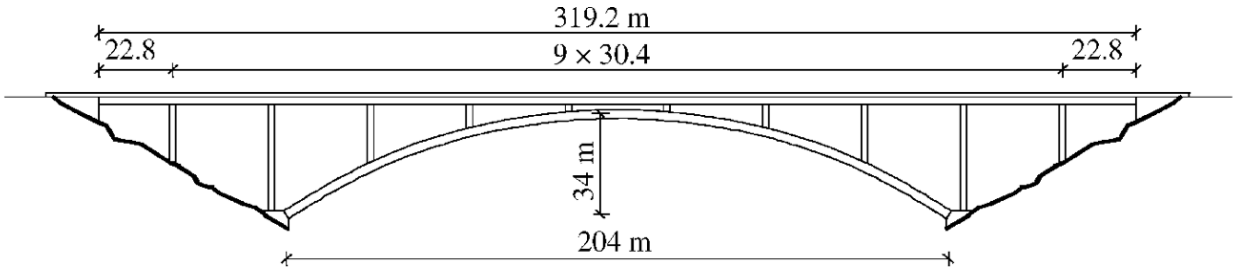
plain

$\phi V_n = \phi \frac{8}{3} b_o h \sqrt{f'_c}$

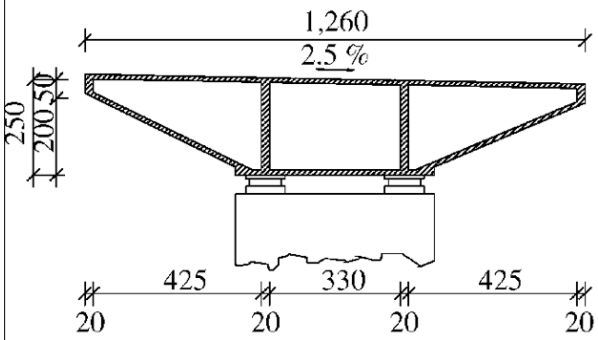
$\phi = 0.55$

b_o is at $\frac{h}{2}$ ...

h = thickness minus 2 if cast against earth

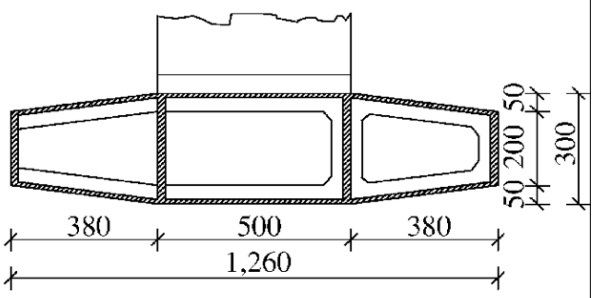


Cross-section at deck

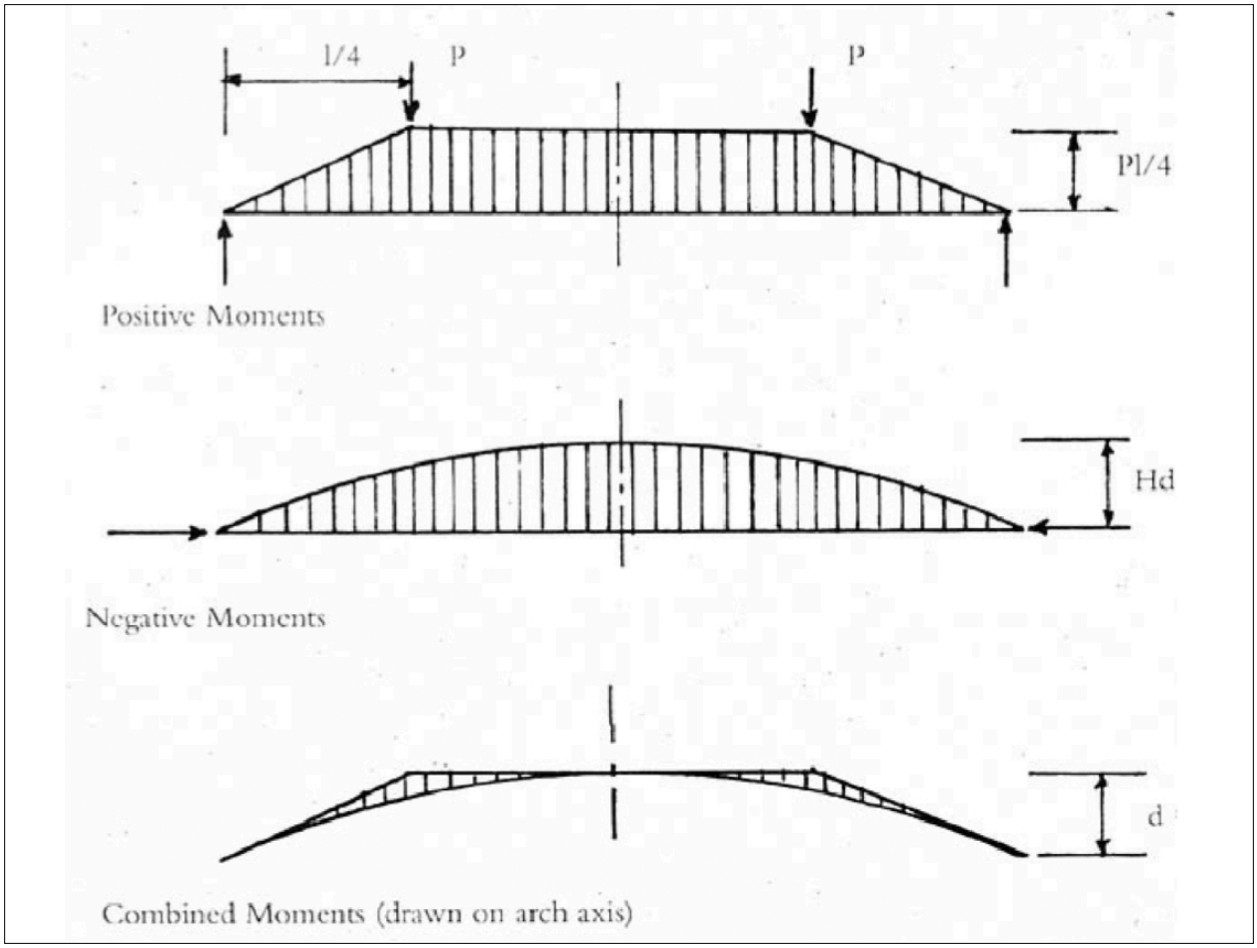


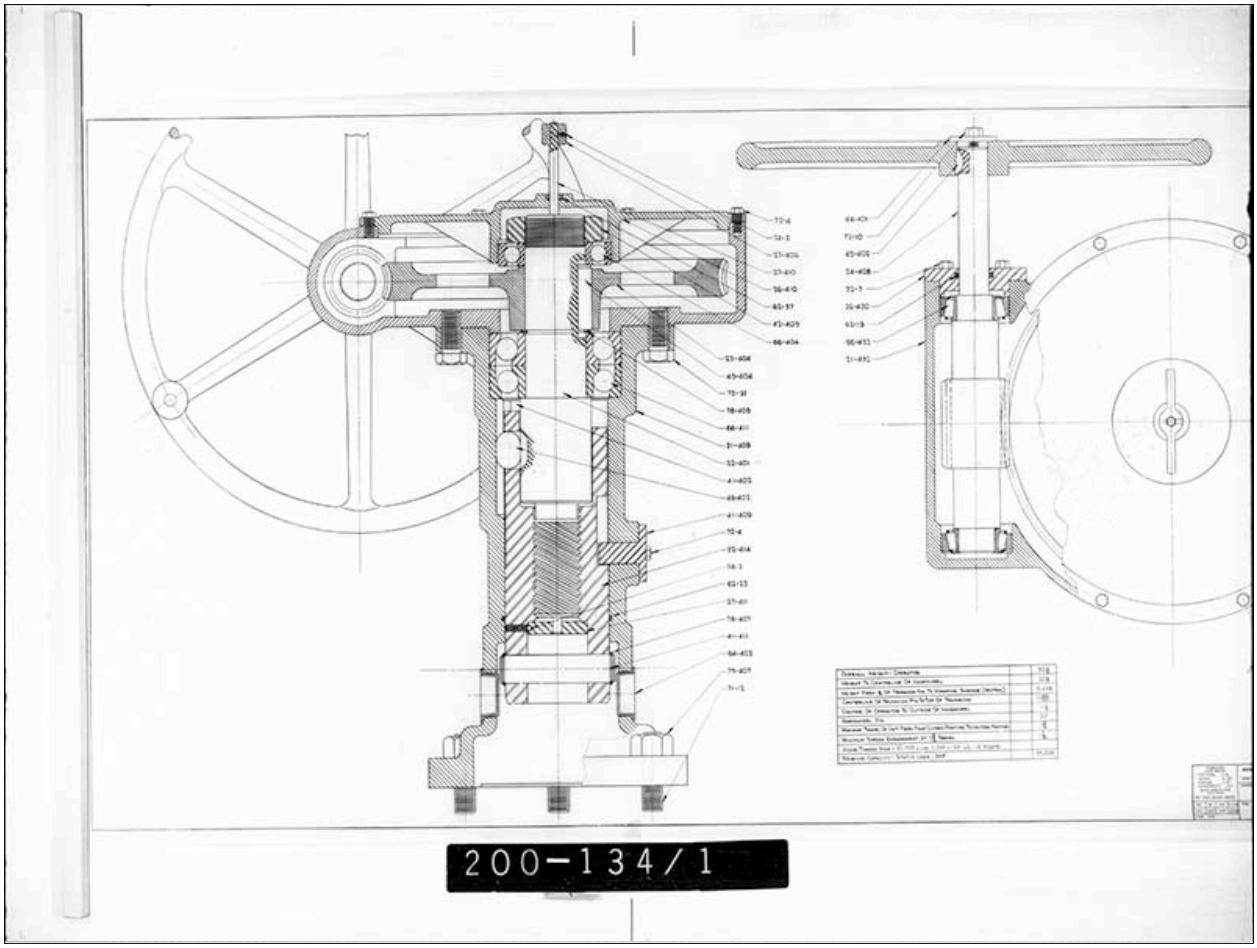
A = 4.247 m^2, I_z = 3.658 m^4, h = 2.50 m

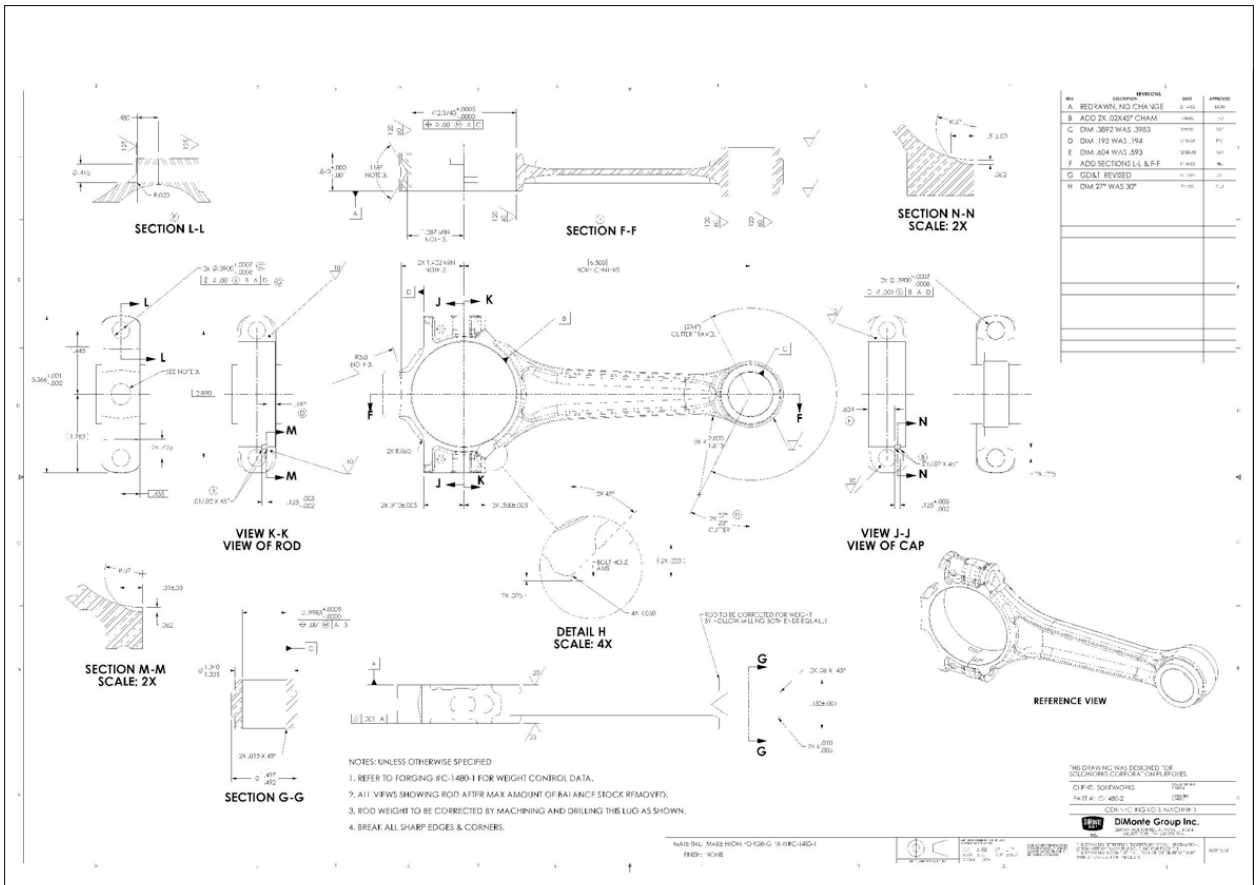
Cross-section at arc

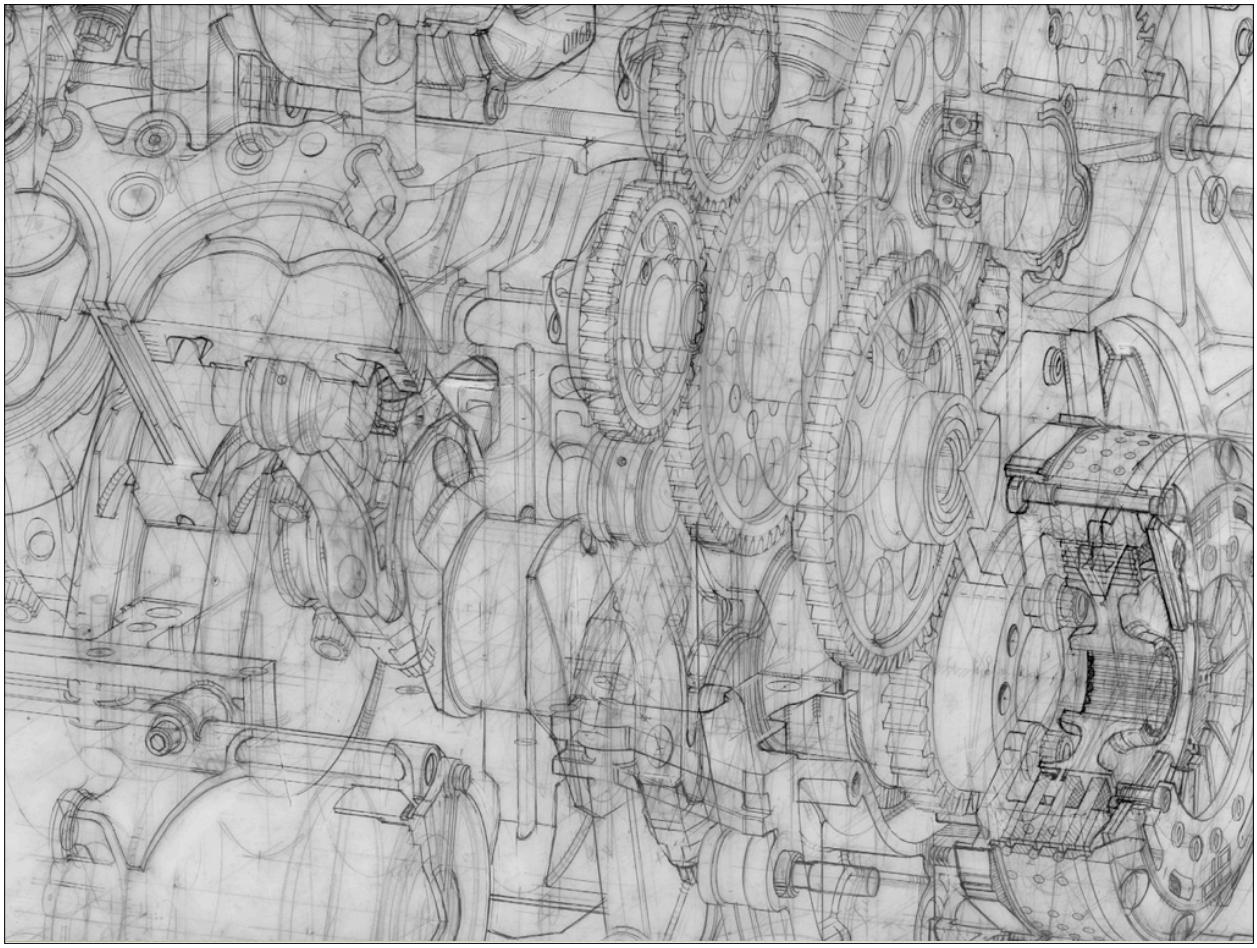


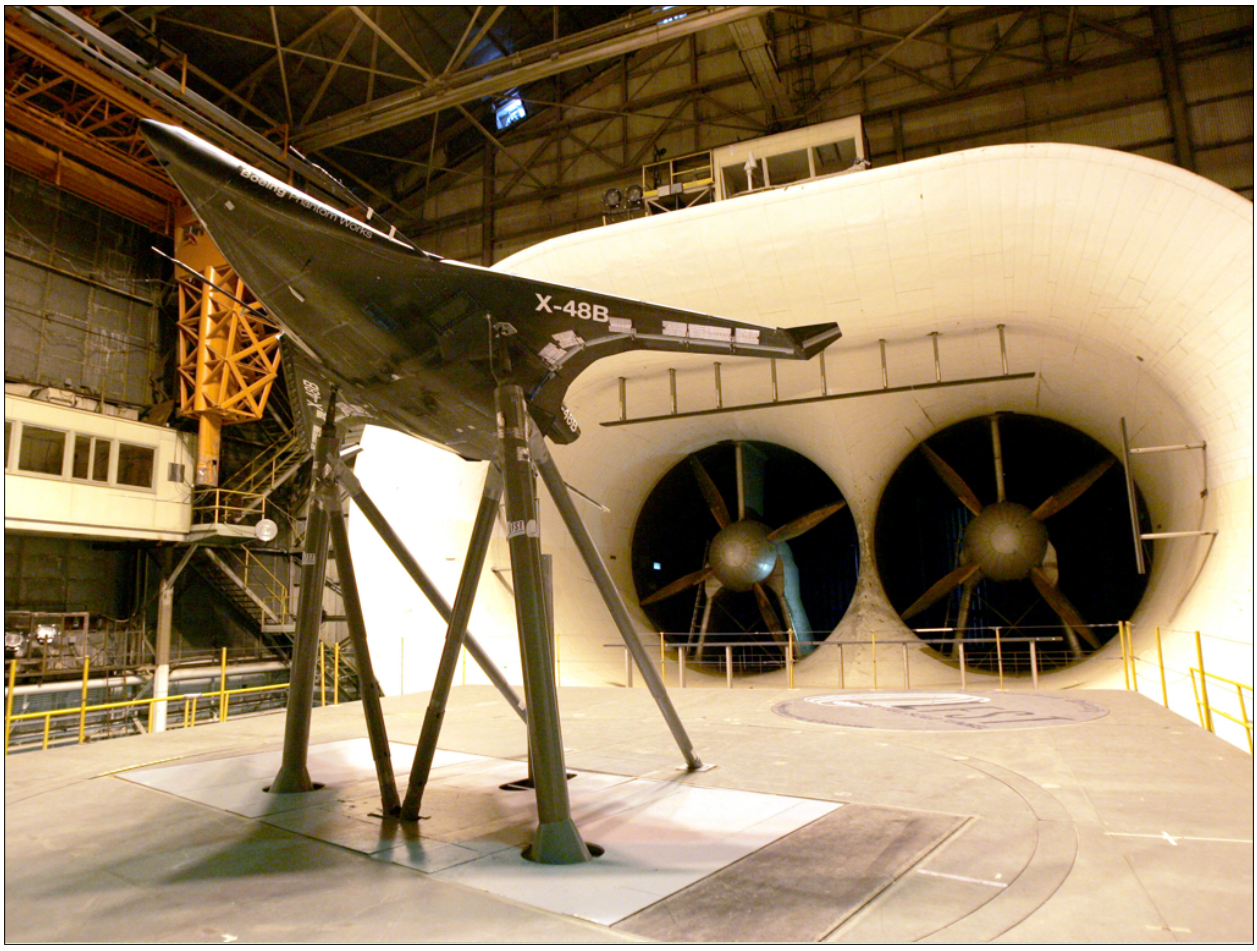
A = 4.836 m^2, I_z = 5.951 m^4, h = 3.00 m


l/4
P
P
Pl/4
Positive Moments
Hd
Negative Moments
d
Combined Moments (drawn on arch axis)


200-134/1


SECTION L-L
SECTION F-F
SECTION N-N
SCALE: 2X
VIEW K-K
VIEW OF ROD
VIEW J-J
VIEW OF CAP
SECTION M-M
SCALE: 2X
DETAIL H
SCALE: 4X
REFERENCE VIEW
SECTION G-G
NOTES: UNLESS OTHERWISE SPECIFIED
1. REFER TO FORGING #C-1480-1 FOR WEIGHT CONTROL DATA.
2. ALL VIEWS SHOWING ROD AFTER MAX AMOUNT OF BALANCE STOCK REMOVED.
3. ROD WEIGHT TO BE CORRECTED BY MACHINING AND DRILLING THIS LUG AS SHOWN.
4. BREAK ALL SHARP EDGES & CORNERS.
DiMonte Group Inc.




X-48B

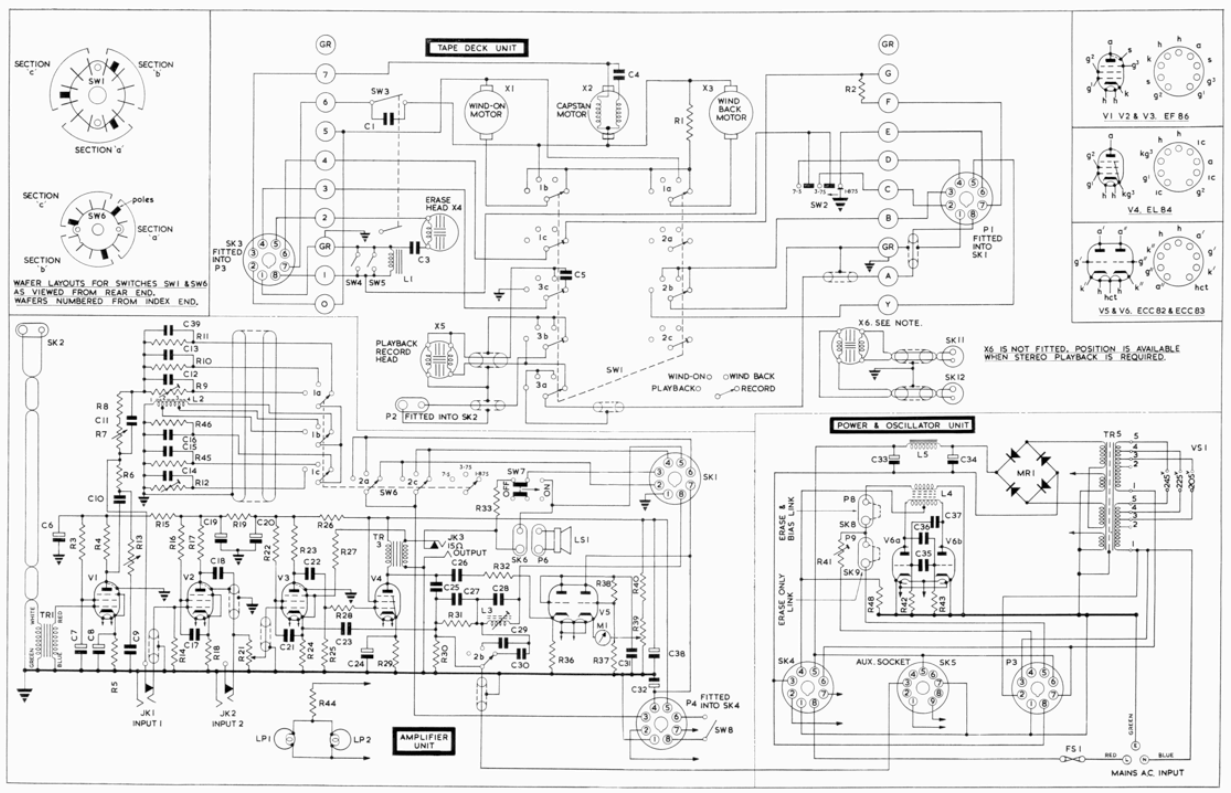
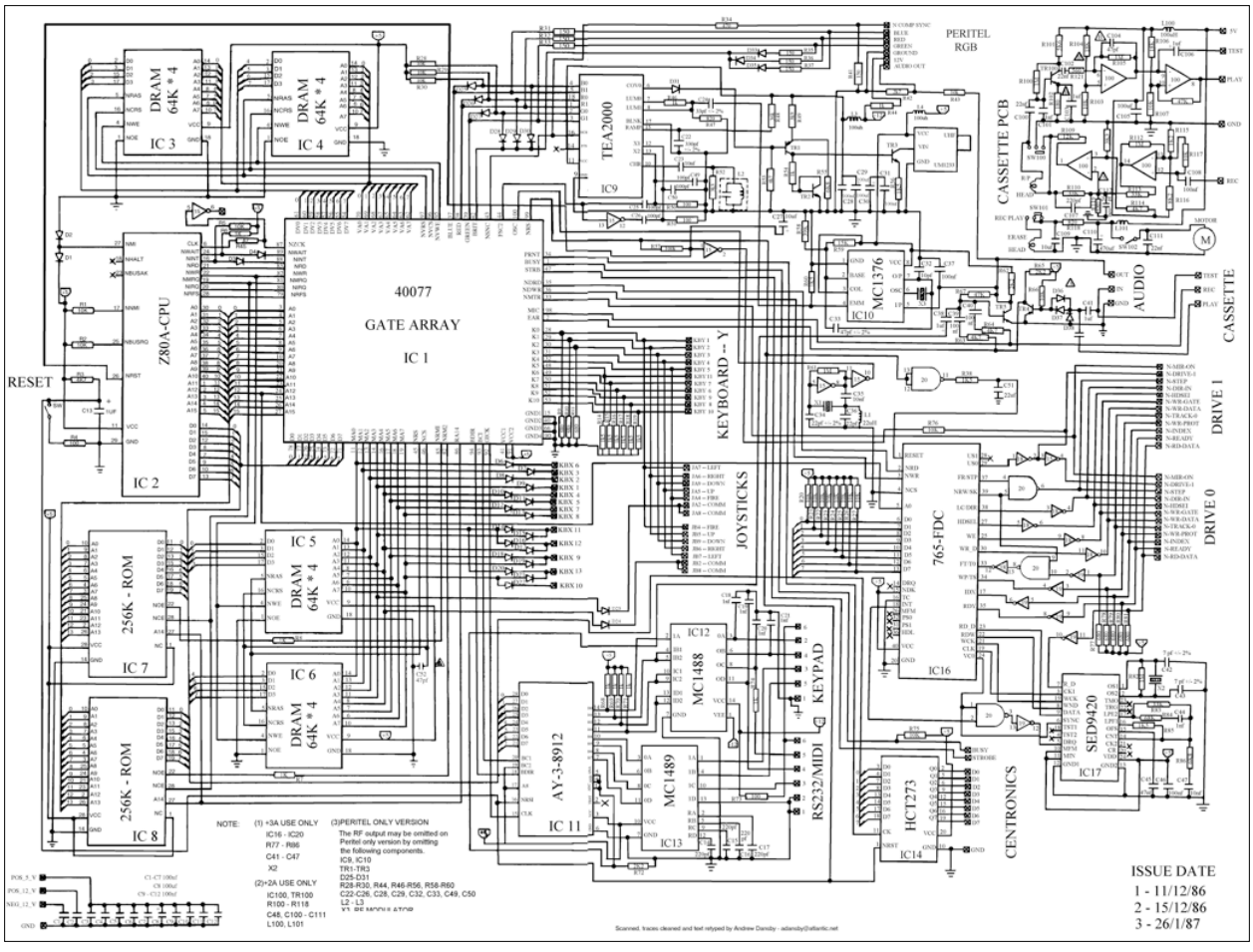
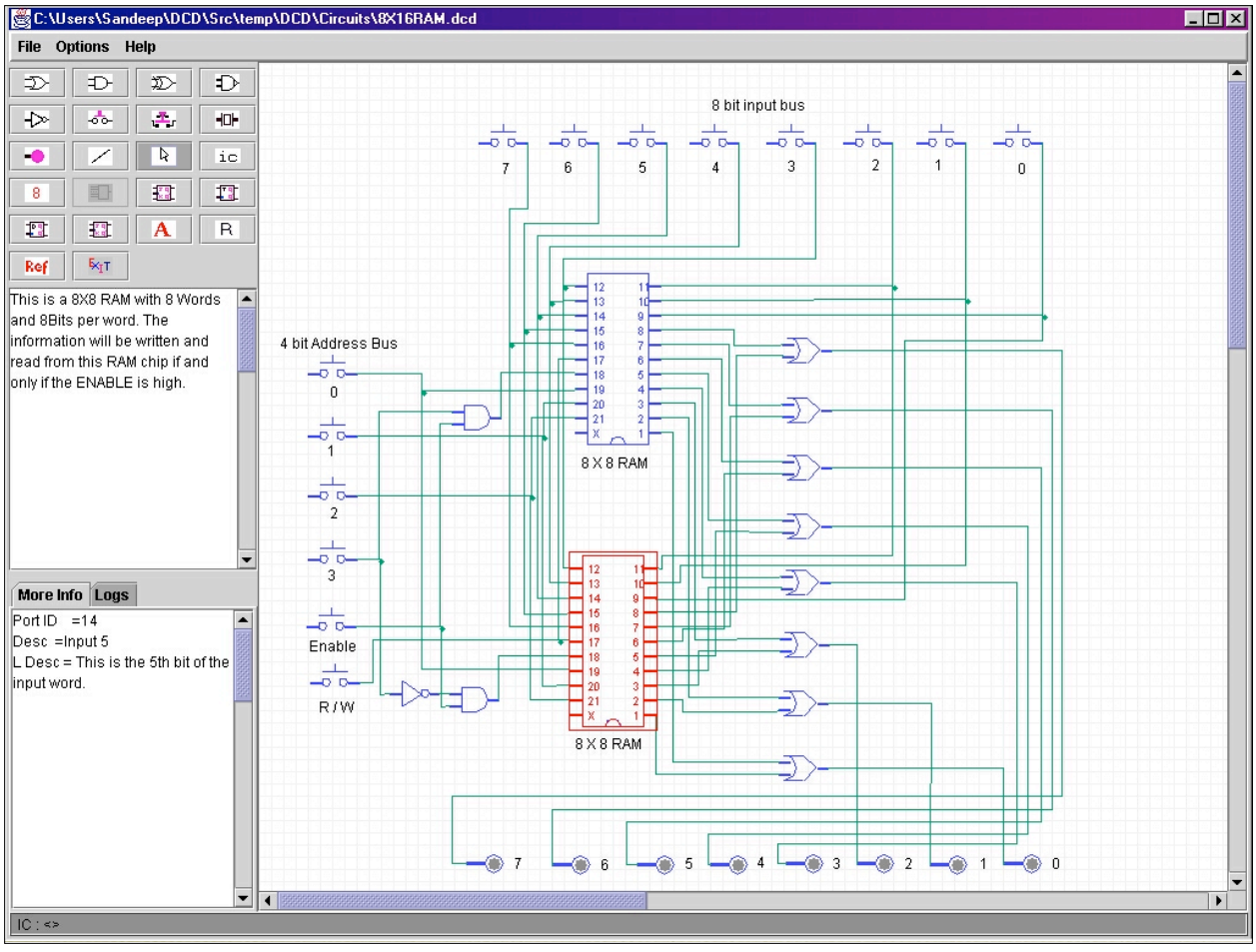
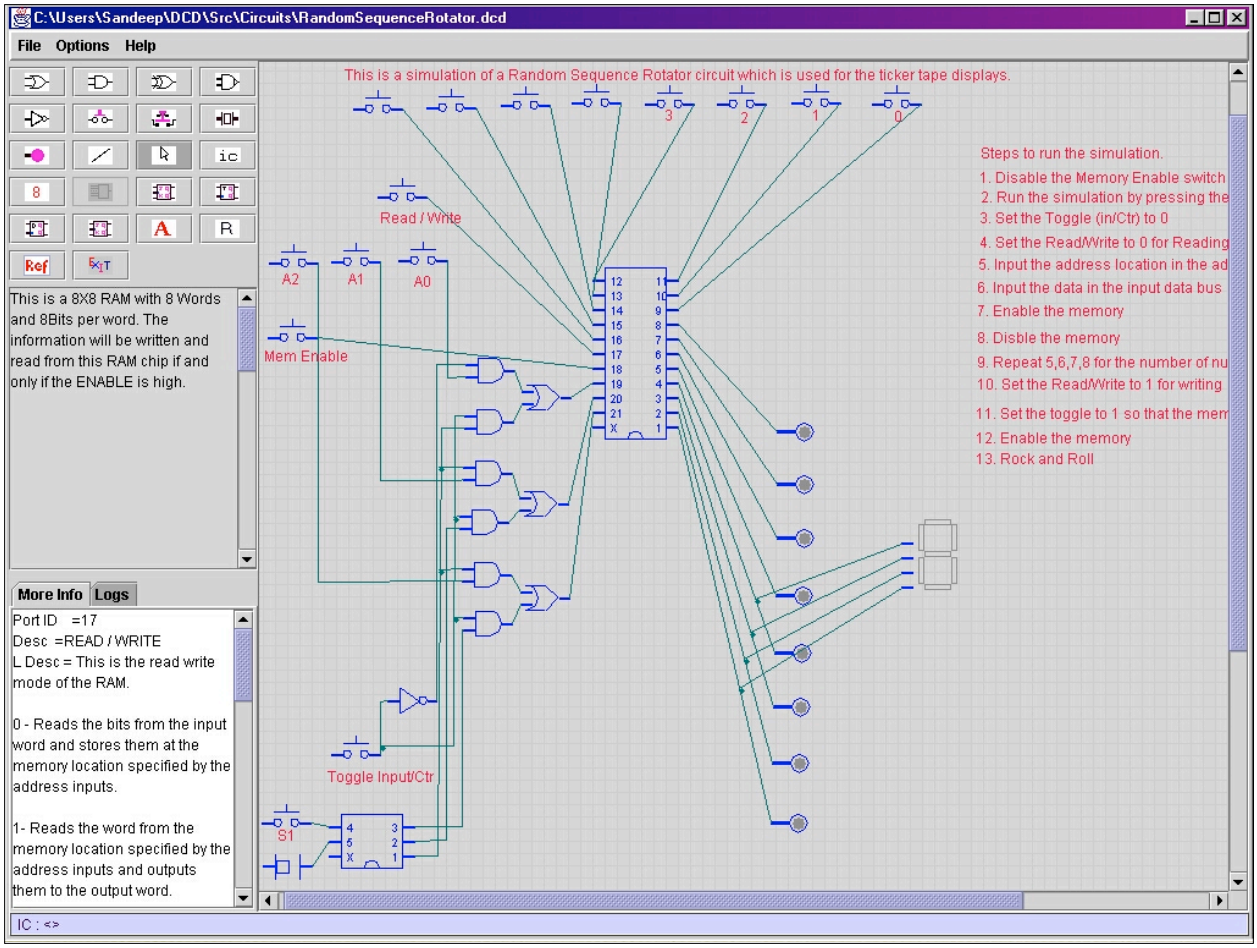


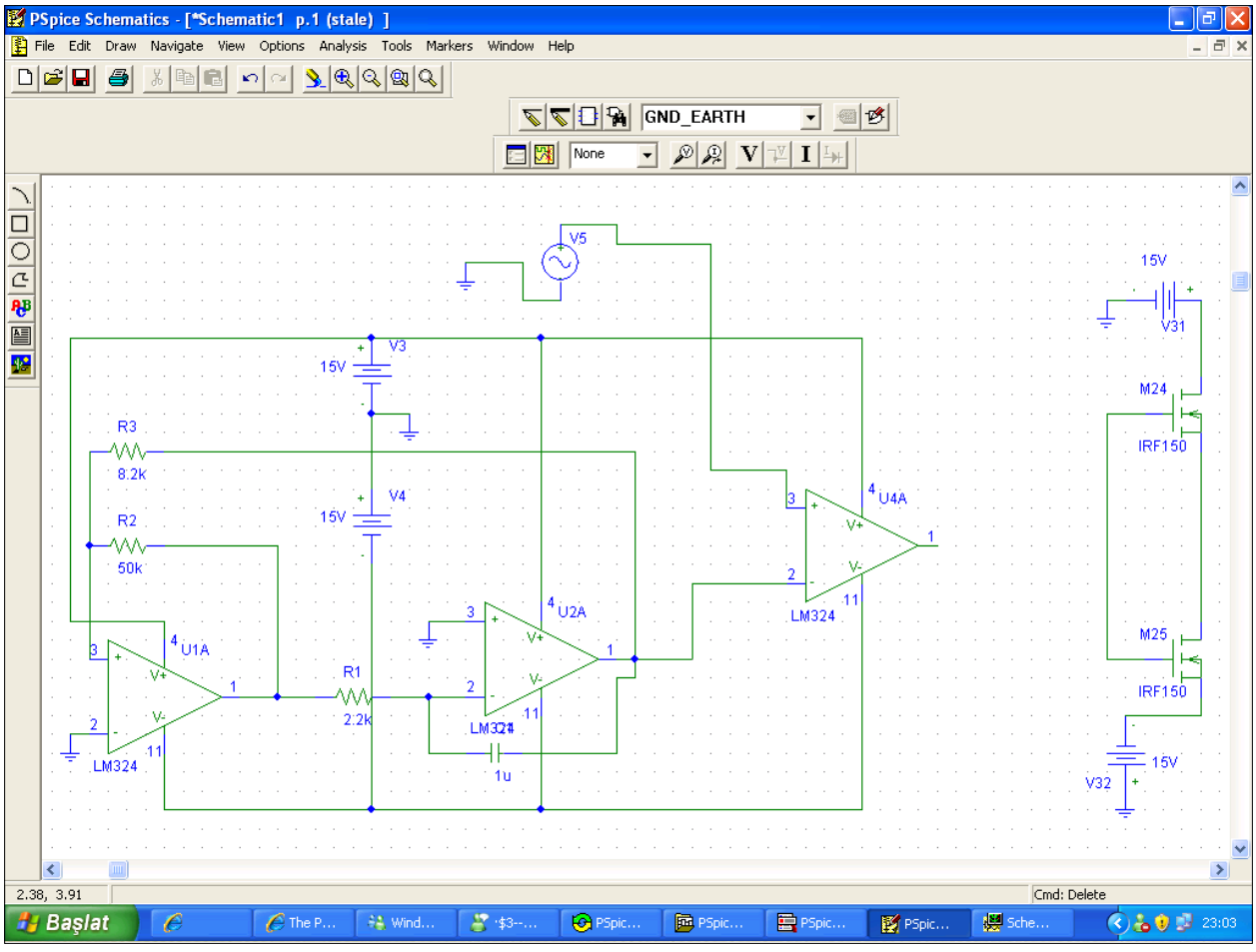
FIG. 20 CIRCUIT DIAGRAM.

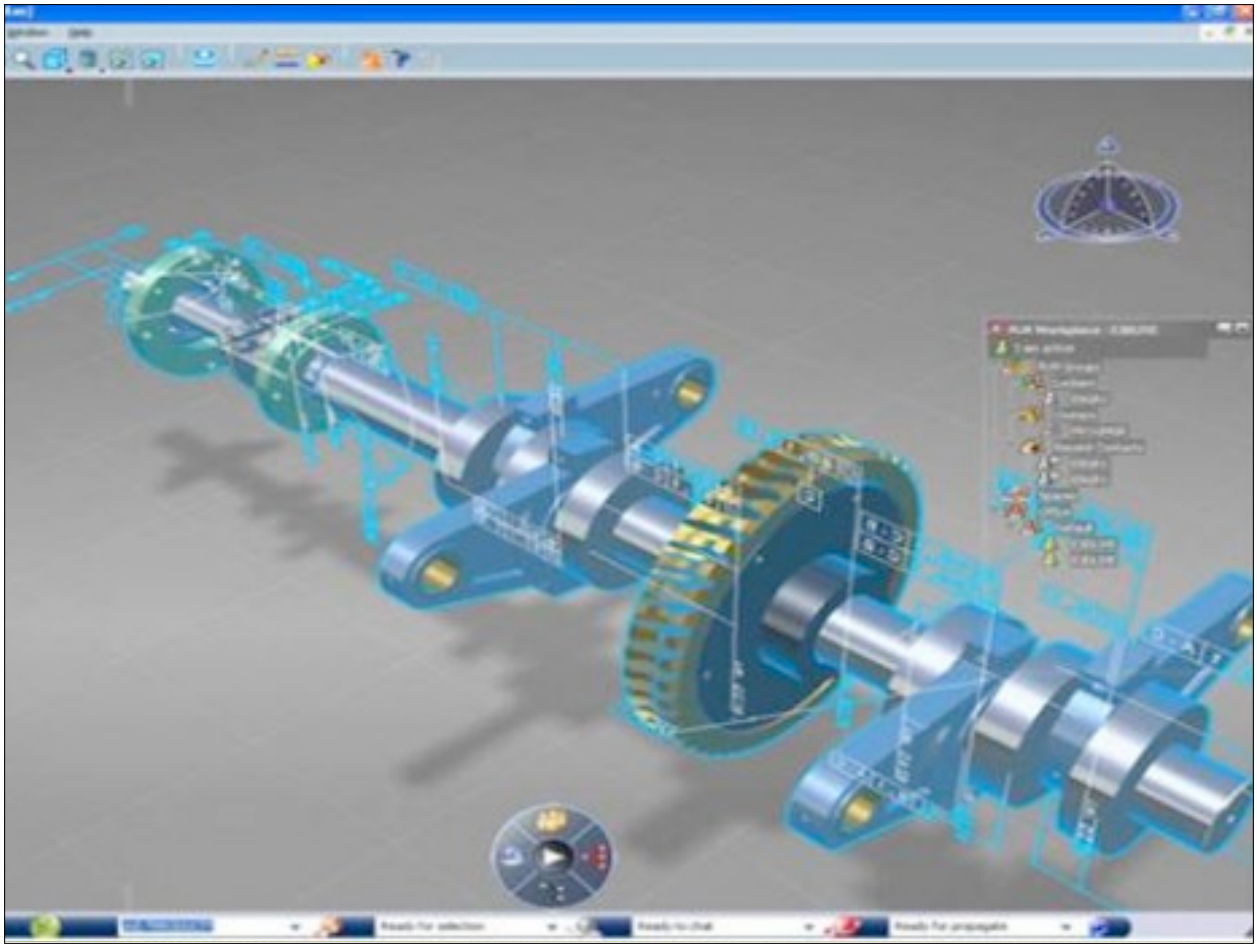


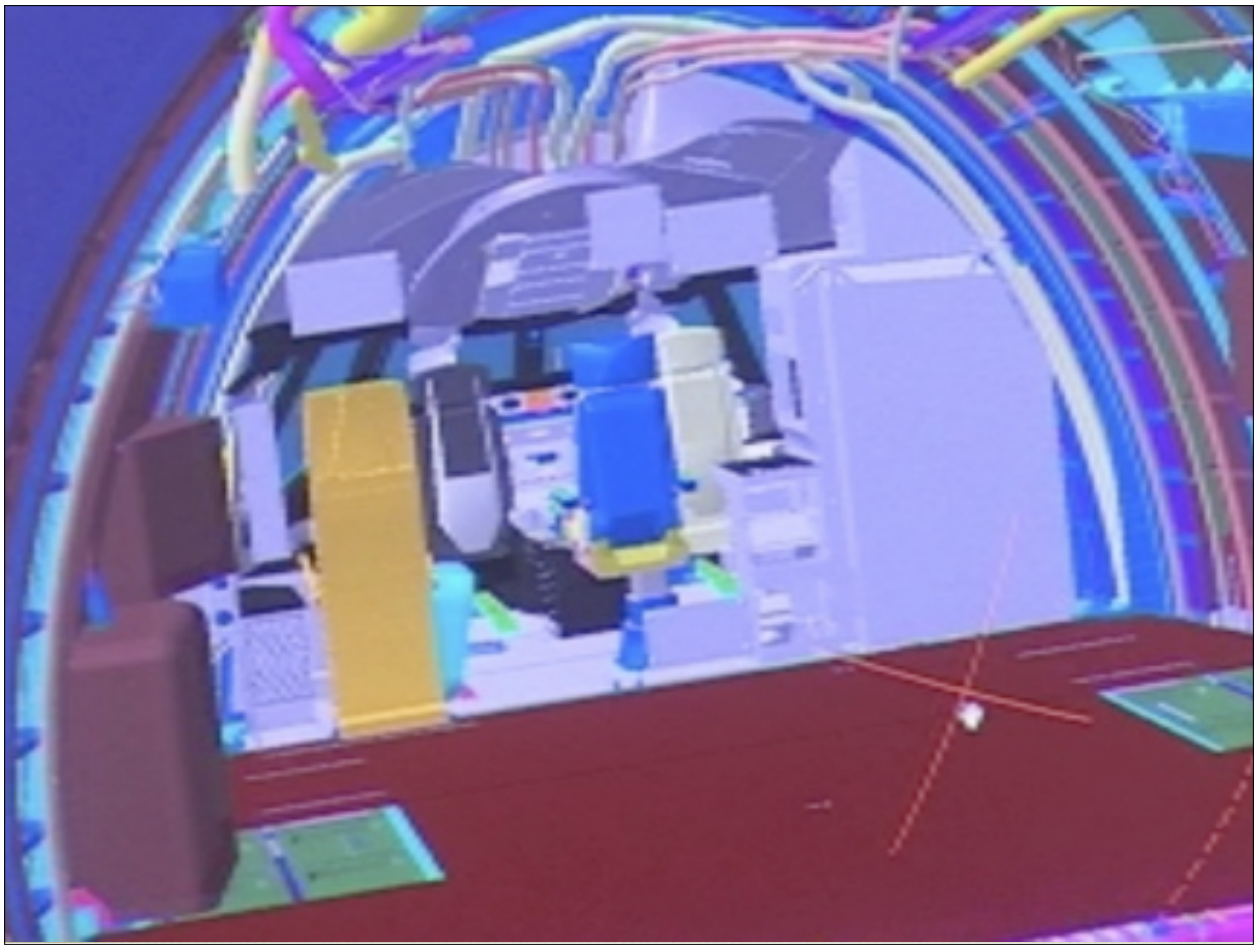
```ruby
module RSpec::Core
  class Reporter
    def initialize(*formatters)
      @formatters = formatters
      @example_count = @failure_count = @pending_count = 0
      @duration = @start = nil
    end

    def report(count)
      start(count)
      begin
        yield self
      ensure
        conclude
      end
    end

    def conclude
      begin
        stop
        notify :start_dump
        notify :dump_pending
        notify :dump_failures
        notify :dump_summary, @duration, @example_count, @failure_count, @pending_count
      ensure
        notify :close
      end
    end
```

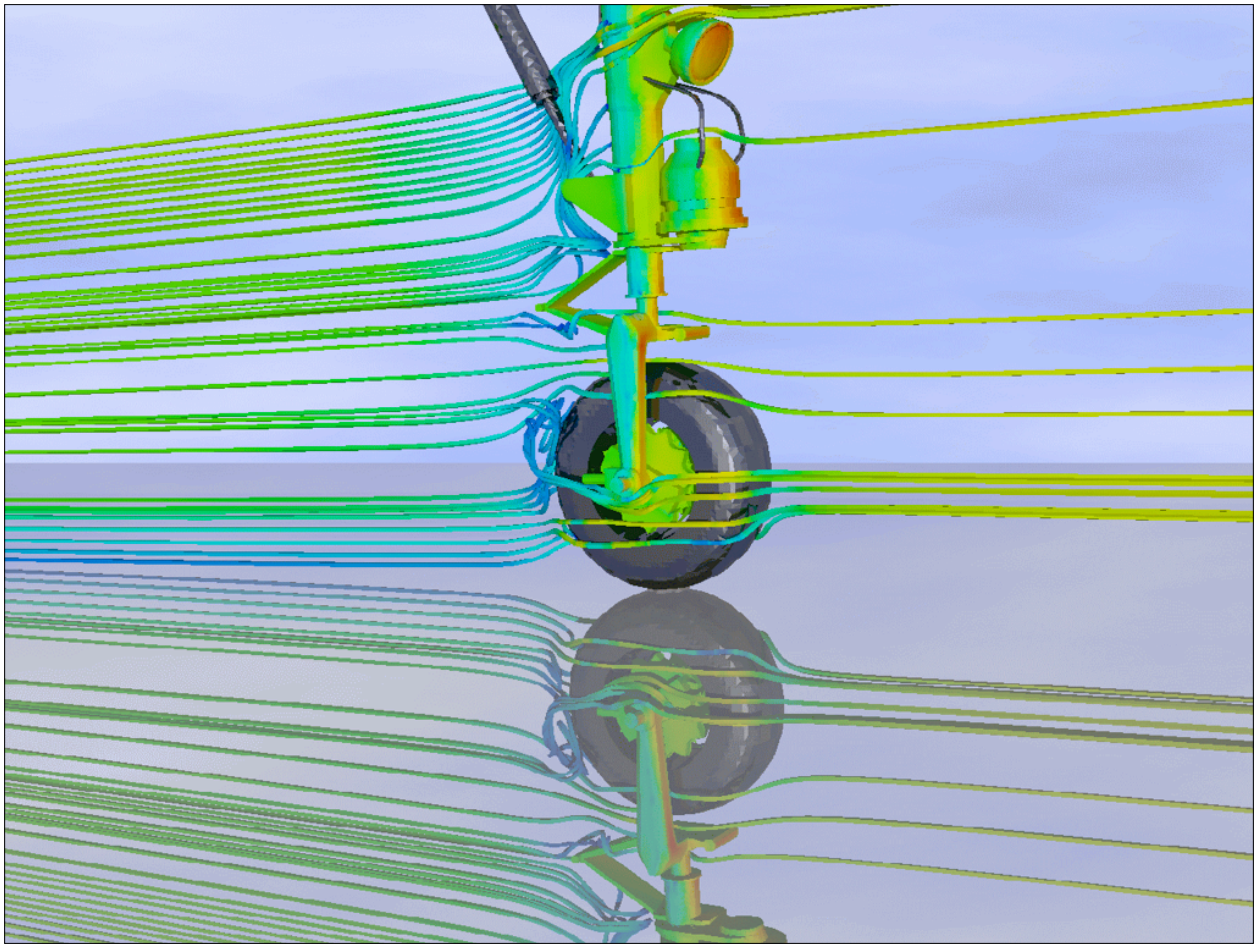


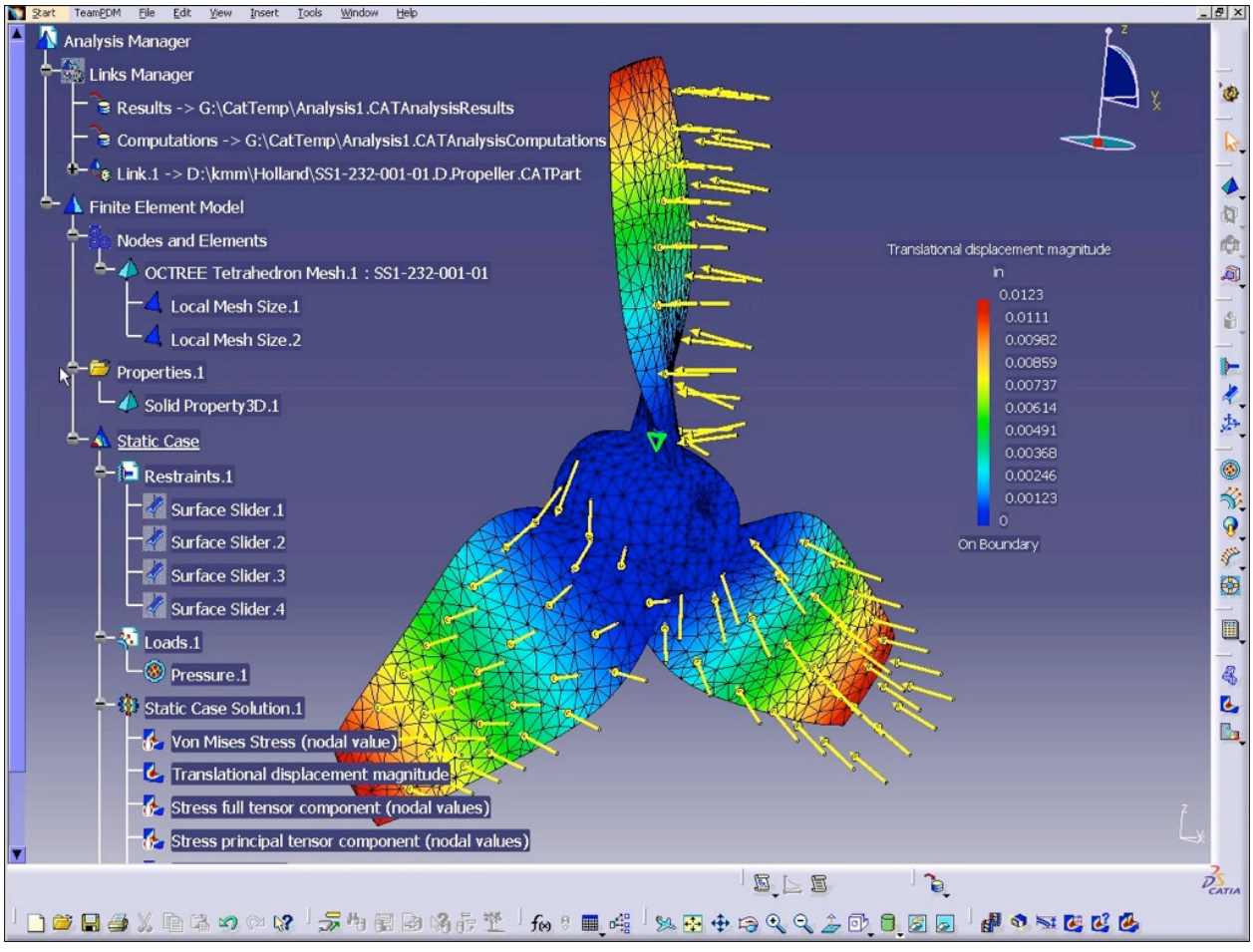

C:\Users\Sandeep\DCD\Src\Circuits\RandomSequenceRotator.dcd
File Options Help
This is a simulation of a Random Sequence Rotator circuit which is used for the ticker tape displays.
Steps to run the simulation.
1. Disable the Memory Enable switch
2. Run the simulation by pressing the
3. Set the Toggle (in/Ctr) to 0
4. Set the Read/Write to 0 for Reading
5. Input the address location in the ad
6. Input the data in the input data bus
7. Enable the memory
8. Disble the memory
9. Repeat 5,6,7,8 for the number of nu
10. Set the Read/Write to 1 for writing
11. Set the toggle to 1 so that the mem
12. Enable the memory
13. Rock and Roll
Read / Write
A2
A1
A0
Mem Enable
Toggle Input/Ctr
S1
This is a 8X8 RAM with 8 Words and 8Bits per word. The information will be written and read from this RAM chip if and only if the ENABLE is high.
More Info
Logs
Port ID =17
Desc =READ / WRITE
L Desc = This is the read write mode of the RAM.
0 - Reads the bits from the input word and stores them at the memory location specified by the address inputs.
1- Reads the word from the memory location specified by the address inputs and outputs them to the output word.
IC : <>

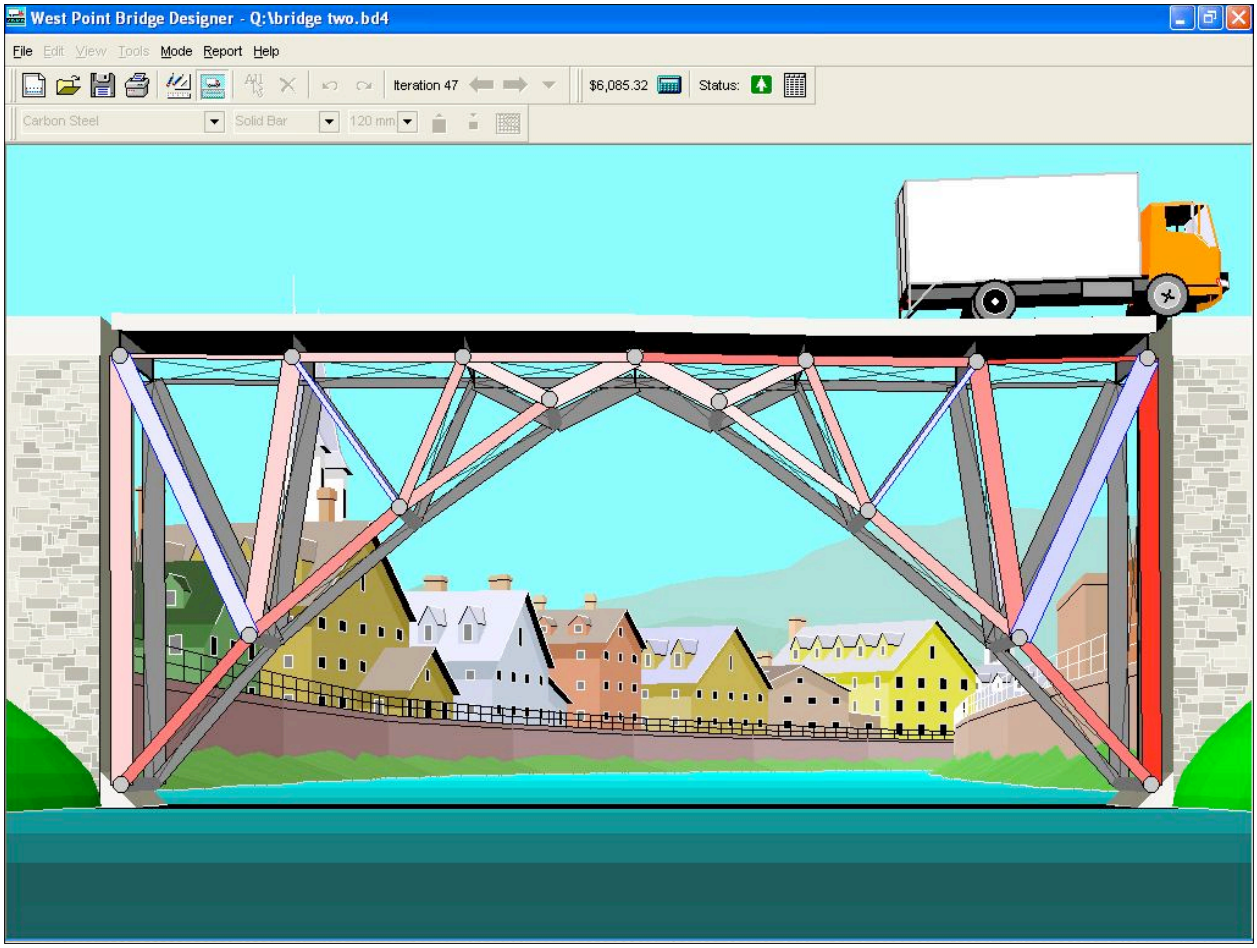

PSpice Schematics - [*Schematic1 p.1 (stale)]
File Edit Draw Navigate View Options Analysis Tools Markers Window Help
GND_EARTH
None
V5
V3
15V
R3
8.2k
V4
15V
R2
50k
U1A
LM324
R1
2.2k
U2A
LM324
1u
U4A
LM324
15V
V31
M24
IRF150
M25
IRF150
15V
V32
2.38, 3.91
Cmd: Delete
Başlat
23:03

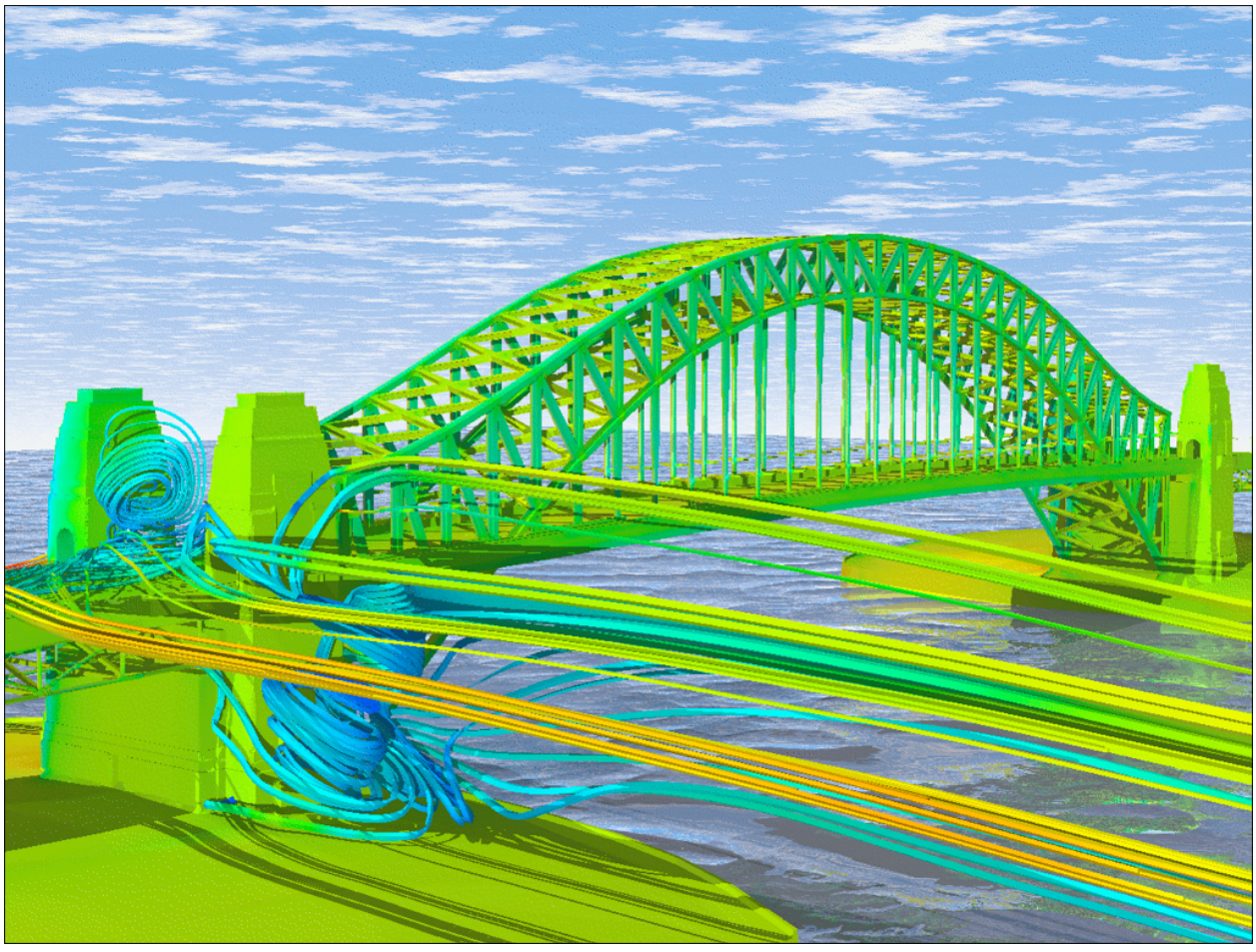







Start
TeamPDM
File
Edit
View
Insert
Tools
Window
Help
Analysis Manager
Links Manager
Results -> G:\CatTemp\Analysis1.CATAnalysisResults
Computations -> G:\CatTemp\Analysis1.CATAnalysisComputations
Link.1 -> D:\kmm\Holland\SS1-232-001-01.D.Propeller.CATPart
Finite Element Model
Nodes and Elements
OCTREE Tetrahedron Mesh.1 : SS1-232-001-01
Local Mesh Size.1
Local Mesh Size.2
Properties.1
Solid Property3D.1
Static Case
Restraints.1
Surface Slider.1
Surface Slider.2
Surface Slider.3
Surface Slider.4
Loads.1
Pressure.1
Static Case Solution.1
Von Mises Stress (nodal value)
Translational displacement magnitude
Stress full tensor component (nodal values)
Stress principal tensor component (nodal values)
Translational displacement magnitude
in
0.0123
0.0111
0.00982
0.00859
0.00737
0.00614
0.00491
0.00368
0.00246
0.00123
0
On Boundary


West Point Bridge Designer - Q:\bridge two.bd4
File Edit View Tools Mode Report Help
Iteration 47
$6,085.32
Status:
Carbon Steel
Solid Bar
120 mm



Software Models (i.e., source code)

- Because the artifact is abstract, model is "concrete".
- Model isomorphic to built artifact.
- *Feels* like working directly with the constructs.
- We are designers *and* builders

Software Models (i.e., source code)

- Furthermore, we are also writers.
- Code must serve two purposes:
  - to be the solution
  - to describe the solution

Programs must be written for people to read, and only incidentally for machines to execute.

—*Harold Abelson and Gerald Jay Sussman*

Software practitioners – especially, ironically, the good ones – often [...] fall in love with the software itself and start thinking of themselves as craftsmen of software.

—*Dan North*

Let us change our traditional attitude
to the construction of programs.
Instead of imagining that our main task
is to instruct a computer what to do,
let us concentrate rather on
explaining to human beings
what we want a computer to do.

—*Donald Knuth*

Internal vs. External

- Abstract vs. concrete
- Hidden vs. public and visible
- Potential effects vs. immediate effects

Programmers have the luxury of being both engineers and craftsmen:

- because we are both designers and makers.
- because we are not insulated from the artifacts we are designing.
- because, like craftsmen, we can *feel* the things we build.

Programmers have the responsibility to be craftsmen, not just engineers:

- because otherwise we inevitably lose sight of the less tangible (but no less important) aspects of our creations.