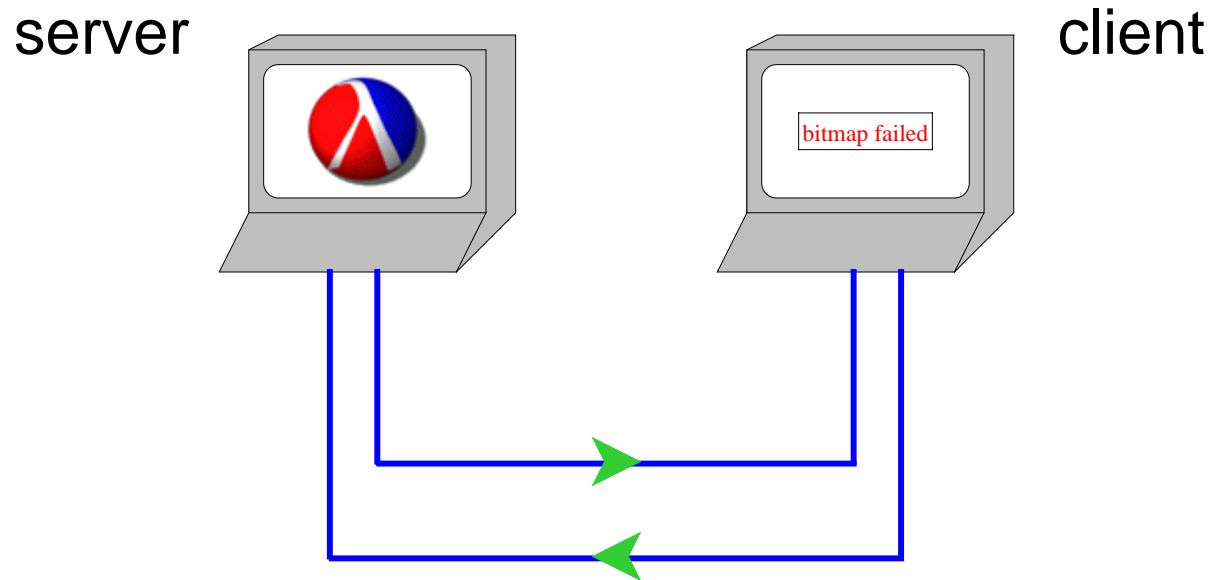


## "The Future of Software Bugs"

*Computerworld* (10/27/03) Vol. 31, No. 49, P. 32; Thibodeau, Patrick

The continuing threat of software bugs stems from a variety of factors, including software vendors and in-house development teams that rush testing and sacrifice quality so they can rapidly move products to market; **academic computer science programs that place more emphasis on development than testing**; and legislation that absolves developers of the blame for damages users suffer as a result of faulty products.

# Web Server Example



Connecting:

Server: `(define l (tcp-listen 4000))`

Client: `(tcp-connect "127.0.0.1" 4000)`  
→ `#<input-port> #<output-port>`

Server: `(tcp-accept l)`  
→ `#<input-port> #<output-port>`

*Examples in DrScheme...*

# Web Page Encoding

A web page is more than plain characters:

<b>CS 2010</b>		
<b>Date</b>	<b>Topic</b>	<b>Notes</b>
Nov 3	Java	<a href="#"><u>slides</u></a>
Nov 5	<i>Mid Term 2</i>	

To encode fonts, color, table layout, links, etc., web servers and clients communicate using ***XML*** ... roughly

```
<html><p align="center"><font size="+2">CS 2010</font></p>
  <table><tr><td><b>Date</b></td>
    <td><b>Topic</b></td>
    <td><b>Notes</b></td></tr>
  <tr><td>Nov 3</td> ...</tr>
  ...</table></html>
```

*Examples in DrScheme...*

## Generating XML

Since XML has an S-expression like structure, and since we're using Scheme, it makes sense to generate S-expressions and convert them to XML

```
(xexpr->string '(html () "Hello"))  
"should be" "<html>Hello</html>"
```

```
(xexpr->string '(html () "0 < 1"))  
"should be" "<html>0 &lt; 1</html>"
```

```
(xexpr->string '(html ()  
                (font ((size "+2"))  
                      "Hello")))  
"should be"  
"<html><font size=+2>Hello</font></html>"
```

If you're using Java, then you'll generate object trees instead of S-expressions, but it's the same idea

*Family tree server in DrScheme...*