“A BRIEF HISTORY OF SOFTWARE”

David L. Hayes
Fenwick & West LLP
April 14, 2016
“A BRIEF HISTORY OF SOFTWARE”

David L. Hayes
Fenwick & West LLP
April 15, 2016
“A BRIEF HISTORY OF SOFTWARE”

David L. Hayes
Fenwick & West LLP
April 15, 2016
“A BRIEF HISTORY OF SOFTWARE”
From Mainframes to Mobile

David L. Hayes
Fenwick & West LLP
April 15, 2016
100 BILLIONTH DOWNLOAD FROM APP STORE
PATENTS

- Benson 1972
- Flook 1978
- Diehr 1981

PATENTS

Bilski 2008

State Street Bank 1998


60  70  80  90  00  10  20
PATENTS

Mortgage Grader 2016

Mainframes Minicomputers PCs World Wide Web Cloud Computing & Open Source

Mobile Computing

60 70 80 90 00 10 11 12 13 14 15 16 17 18 19 20
11 Tests for Patentability
COPYRIGHTS

Standardization on Windows & Mac OS and web browsers
2000

COPYRIGHTS

Oracle v Google 2014

Mass market
Uncertainty of SW patents
Open source
Cloud computing

COPYRIGHTS

TRADE SECRETS

Beginning of PC era
TRADE SECRETS

Look & feel

Salesforce 1999

A BRIEF HISTORY of SOFTWARE

- Mainframes
- Minicomputers
- PCs
- World Wide Web
- Cloud Computing & Open Source
- Mobile Computing
How to Tie a Tie

1.

2.

CartoonChurch.com
A BRIEF HISTORY of SOFTWARE
Why is software so hard to protect?
“Software” is not a monolithic thing
7 key characteristics
1. Software is inherently functional
1. Software is inherently functional

Requires creativity, but for functional outcomes
1. Software is inherently functional

Interoperability
2. Software embodies multiple types of creativity
2. Software embodies multiple types of creativity

Traditional expression (games)
2. Software embodies multiple types of creativity

Line by line code
2. Software embodies multiple types of creativity

Non-literal internal elements (SSO, data formats, command sets)
2. Software embodies multiple types of creativity

External interfaces (APIs, input formats, GUI)
3. Software evolution is often incremental
3. Software evolution is often incremental

Limited record of prior art (it’s often just in the code)
3. Software evolution is often incremental

Protection of incremental changes may not be very important
3. Software evolution is often incremental

Object oriented programming and reuse of code
4. Software is increasingly short lived
4. Software is increasingly short lived

UNIX
50 yrs (1969)
4. Software is increasingly short lived

- **UNIX**
  - 50 yrs (1969)

- **Mainframe**
  - 10-20 yrs
4. Software is increasingly short lived

- UNIX: 50 yrs (1969)
- Mainframe: 10-20 yrs
- PC software: 6 mos - 2 yrs
4. Software is increasingly short lived

<table>
<thead>
<tr>
<th>Technology</th>
<th>Lifespan</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIX</td>
<td>50 yrs (1969)</td>
</tr>
<tr>
<td>Mainframe</td>
<td>10-20 yrs</td>
</tr>
<tr>
<td>PC software</td>
<td>6 mos - 2 yrs</td>
</tr>
<tr>
<td>Mobile app</td>
<td>A few weeks</td>
</tr>
</tbody>
</table>
5. Software development methodology has evolved
5. Software development methodology has evolved

Small development teams working over a long time (mainframes)
5. Software development methodology has evolved

Large enterprise development teams using top-down methods
5. Software development methodology has evolved

Inter-company joint development
5. Software development methodology has evolved

Agile development
5. Software development methodology has evolved

Open source
6. Software exists in many different markets
6. Software exists in many different markets

Discrete, small markets for mainframe software
6. Software exists in many different markets

Medium size markets for enterprise software
6. Software exists in many different markets

Mass markets for consumer software
6. Software exists in many different markets

Essentially infinite markets for mobile apps
7. Software has many different distribution and use architectures
7. Software has many different distribution and use architectures

On premises
7. Software has many different distribution and use architectures

Client/server
7. Software has many different distribution and use architectures

Cloud computing (public and private)
7. Software has many different distribution and use architectures

SaaS/ASP/On-demand
7. Software has many different distribution and use architectures

Grid computing & other forms of distributed computing
7. Software has many different distribution and use architectures

Peer-to-peer
7. Software has many different distribution and use architectures

Mobile computing
A BRIEF HISTORY of SOFTWARE

“There you have it”