EasyVision family of products

Examination rooms

R/F

Vascular

phase 1: 1992

print

store

export

Examination rooms

MR

CT

phase 2: 1994

MPR

MPR

print

store

export

phase 3: 1996

print

view

clinical focus

research

archive
Product types:

• Modality productivity enhancers:
  + Easyvision R/F
  + Easyvision RAD
  + Easyvision CT/MR

street price ca 50 k$, high added clinical value; sales directly related to modality sales

• Clinical Focus:
  + Neurovision
  + Image Guided Surgery

street price ca 100 k$, very high added clinical value; sales limited to specialist areas

• “PACS” workstations
  + Teleradiology Workstation
  + Critical Care Workstation
  + Multi modality review station

street price ca 25 k$, low added value, low margin; sales potentially very high
Extrapolation CDS SW.

<table>
<thead>
<tr>
<th>Year</th>
<th># appl</th>
<th>Mega lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>1991</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>1992</td>
<td>2</td>
<td>0.35</td>
</tr>
<tr>
<td>1994</td>
<td>8</td>
<td>0.6</td>
</tr>
<tr>
<td>1996</td>
<td>32</td>
<td>1.5</td>
</tr>
<tr>
<td>1999</td>
<td>100</td>
<td>5</td>
</tr>
</tbody>
</table>
### Table 1: Efficiency through re-use

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>number applications</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>applications</td>
<td>1</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>inputs, a.o. modalities</td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td><strong>people</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>infrastructure</td>
<td>20+15</td>
<td>21+16</td>
<td>22+16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>application</td>
<td>27</td>
<td>35</td>
<td>41</td>
<td></td>
<td></td>
</tr>
<tr>
<td>total</td>
<td>52</td>
<td>62</td>
<td>72</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td><strong>efficiency</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>people per application</td>
<td>13</td>
<td>8</td>
<td>5</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>
To OO or not to OO

Characteristics of the Easyvision application are:

• Large variety in input images
  + $256^2$, $480^2$, $512^2$, $1024^2$, non square, etc.
  + 8, 10, 12 bits
  + CT, MR, X-ray Image Intensifier

• Large variety in application requirements

• Large variety in use

Easyvision is impossible without OO
Method

Easyvision development method:

- prototype
- evaluate
- engineering

No formal analysis/design/documentation method!

Formal methods:

- work for small projects only
- playground for academics :-)
Simplified layers

<table>
<thead>
<tr>
<th>applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>services and common appl</td>
</tr>
<tr>
<td>toolboxes</td>
</tr>
<tr>
<td>CDSpack</td>
</tr>
<tr>
<td>HW + OS</td>
</tr>
</tbody>
</table>
Adding an application

applications

services and common appl

toolboxes

CDSpack

HW + OS

new

modified

extended
Interfaces

Internal:

• Information model

• Communication mechanism(s):
  + Database storage and notification support
  + Connection
  + In and out streaming support

• API’s to common applications, toolboxes and CDSpack:
  + Objective-C classes and methods
  + properties

External:

• Dicom + PMS + ICS information model

• DICOM services and mechanisms

• PMS and ICS services
Information model

Applications

segments

ICS/WS

PMG

PMS

ARC

DICOM

ACR/NEMA

High innovation rate

stimulate

stimulate

promote

Total transition takes more than 5 years

High interoperability
The platform as deliverable

<table>
<thead>
<tr>
<th>Development process</th>
<th>Source</th>
<th>Software</th>
<th>shared and product</th>
<th>generation recipes</th>
<th>review metrics</th>
<th>admin highlights</th>
</tr>
</thead>
<tbody>
<tr>
<td>test programs, test definition</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(in, out, criterium)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>test programs, test definition</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(in, out, criterium)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>target OS, other purchased items</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>compile, link etc tools</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cluster OS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>customization, support and control of dev. process</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>configuration management (code, PR’s, documentation)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>infrastructure conditions, HW+OS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>actual infrastructure and resources</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>mapping on infrastructure and resources</th>
<th>specifications</th>
<th>review metrics</th>
<th>admin highlights</th>
</tr>
</thead>
<tbody>
<tr>
<td>test images</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>clinical images</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>documentation tools</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The Buy myth.

The nineties MBA course teaches:

Thou shall buy....

The poor heathen suffers from NIH (Not Invented Here) syndrome.

Reality is somewhat more complicated!
Buy, potential components:

- Operating system
- Communication
- Database engine
- User interface and related utilities
- Graphics and related utilities
- Image processing
- 3D rendering
- Foundation classes
- Installation
- Licensing, SW keys
- Security, a.o. encryption
- Multi media, virtual reality peripheral support
- etc.
Embedding

- Installation

- Configuration

- Customization

- Start up, shutdown

- Specifications:
  + functional
  + system design
  + sw design

- Interface to application SW:
  + add semantics level
  + use of appropriate low level mechanisms
  + match to high level mechanisms:
    - notification, scheduling
    - job requests, subscriptions
Embedding (continued)

- Exception handling:
  + System monitor
  + Error propagation
  + Logging

- Resource allocation and monitoring provisions
  + CPU
  + Memory
  + Disk

- Resource tuning, see above

- Safety design

- Security design
innovation from outside
concentration on key technology
initial cost reduction
low level interoperability
cohabitation

transition cost
know how required
release propagation
integration effort
embedding
flexibility
resource use
performance
license costs

BUY
MAKE
Release propagation delay

- Layer 5
- Layer 4
- Layer 3
- Layer 2
- Layer 1

Product with renewed layer 1 deliverable

Layer 1 release

Layer 2 integration alfa, beta

Layer 3 integration alfa, beta

Layer 4 integration alfa, beta

Layer 5 integration alfa, beta

Time

1 year

2 years
The Buy myth converted in common sense:

The right questions to ask are:

- When to buy?

- How can the design enable buy?

- Which process is needed for buy?
The Re-use myth put into perspective:

The right questions to ask are:

- When to re-use (cost vs. benefit)?
- How can the design enable re-use?
- Which process is needed for re-use?
Pro Re-use

- development cost sharing
- verification cost sharing
- same look and feel
- application developers focus on application
- increased quality, due to repeated use

Contra Re-use

- cost of generalization
- overhead cost
- increase of total complexity
- coupling of lifecycles, products, schedules
- vulnerability (Biological evolution is based on diversity...)

See make vs. buy

Re-use is means not goal.
A look into the future

From box to function:

- customer wants any function on any location/time, not limited by “random” product or box boundaries

In parallel with:

- large number of clinical applications
- integration of health care function
- break down in manageable projects / teams, lifecycle independency
- finite number of skilled development personnel
**Table 2:**

<table>
<thead>
<tr>
<th></th>
<th>1980</th>
<th>1995</th>
<th>2005</th>
</tr>
</thead>
<tbody>
<tr>
<td>integration level</td>
<td>generator</td>
<td>department</td>
<td>health</td>
</tr>
<tr>
<td></td>
<td>stand</td>
<td>care</td>
<td>care</td>
</tr>
<tr>
<td>time to market</td>
<td>2-5 year</td>
<td>1-2 year</td>
<td>0.5 year</td>
</tr>
<tr>
<td>code size complete</td>
<td>$10^4-10^5$</td>
<td>$10^6$</td>
<td>$10^7-10^8$</td>
</tr>
<tr>
<td>product</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>memory size</td>
<td>96 kB</td>
<td>96 MB</td>
<td>? GB</td>
</tr>
<tr>
<td>CPU power</td>
<td>0.1 MIPS</td>
<td>100 MIPS</td>
<td>? GIPS</td>
</tr>
<tr>
<td>dev group size</td>
<td>10-50</td>
<td>50-200</td>
<td>?</td>
</tr>
<tr>
<td>of which ASW</td>
<td>2-10</td>
<td>20-60</td>
<td>?</td>
</tr>
</tbody>
</table>
Skills

- increased integration
- increased complexity
- increased abstraction
- increased focus on application
- increased (time) pressure

Increased skills required
Profile of 1999 developer

- application oriented
- development process aware
- multi disciplinary
- fluent in software engineering lingo

These people are rare!

Our industry will become skill limited,
instead of initial cost limited
specification | design | make | test

covered by one person
Technological changes, opportunities

- Corba, SOM, OLE, ...
- Java, ...
- Windows NT, Windows 95, OS 2
- Taligent, Spring, ...
- SW only products
- Multi media (HW+SW)
EasyVision in 2000

• More than 100 independent applications
• Interoperating fluently with other EV applications
• Interoperating fluently with other vendors
• Interoperating fluently with other health care applications (Information systems, etc.)
• SW only
• Running on at least UNIX and NT platforms
• Distributed development process
• Consolidation and cross fertilization process
• Platform for innovative applications in image handling, analysis, clinical focus.
Re-use levels

• Concepts
• Development Process
• Interoperability architecture
• Functional specifications
• User interface
• Algorithms
• Design
• Verification (test suite, spec)
• Skills
• Copy implementation, code
• Implementation, code
• Application modules
september 1991

View, test vehicle only

image | gfx | UI | DB

SunOS, SunView

Standard Sun workstation
september 1992

R/F application

<table>
<thead>
<tr>
<th>Print</th>
<th>Store</th>
<th>View</th>
<th>Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>spool</td>
<td>HCU</td>
<td>image</td>
<td>gfx</td>
</tr>
<tr>
<td>RC driver</td>
<td>HC driver</td>
<td>DOR driver</td>
<td>NIX</td>
</tr>
<tr>
<td>SunOS</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Print Store View Cluster

RC interf HC interf DOR Standard IPX workstation

Desk, cabinet, cables, etc.

start up

RC 3M

DSI
june 1994

<table>
<thead>
<tr>
<th>EasyVision CT/MR</th>
<th>EasyVision R/F</th>
<th>EV RAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>specialized appl. (dental, etc.)</td>
<td>specialized appl.</td>
<td></td>
</tr>
<tr>
<td>MR</td>
<td>CT</td>
<td>RF</td>
</tr>
<tr>
<td>Compose</td>
<td>Print</td>
<td>Store</td>
</tr>
<tr>
<td>spool</td>
<td>HCU</td>
<td>store</td>
</tr>
</tbody>
</table>

CDSpack

**Solaris**

**RC inter**

**HC inter**

**DOR**

Standard IPX or SS5 workstation

Desk, cabinet, cables, etc.

---

PMS-ICS-workstations

31 of 33

Gerrit Muller
june 1994

<table>
<thead>
<tr>
<th>EasyVision</th>
<th>EV PCR</th>
</tr>
</thead>
<tbody>
<tr>
<td>specialized appl. (dental.)</td>
<td>specialized appl.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MR</th>
<th>CT</th>
<th>RF</th>
<th>Vascular</th>
<th>Cardio</th>
<th>PCR</th>
</tr>
</thead>
</table>

Compose | Print | Store | MPR | View | Export | Cluster |

spool | HCU | store | image | gfx | UI | DB | PMS net in | PMS net out |

CDSpack

RC driver | HC driver | DOR driver |

Solaris

RC interf | HC interf | DOR | Standard IPX or LX+ workstation |

Desk, cabinet, cables

<table>
<thead>
<tr>
<th>MR</th>
<th>CT</th>
<th>DSI</th>
<th>DCAS</th>
<th>PCR</th>
</tr>
</thead>
</table>

new HCU

Remote access customization

Service mode

SW keys

Config

Install

Start up