Failures associated with HCI

But its not the users fault!

Brendan Murphy

MSR Cambridge

Talk contents

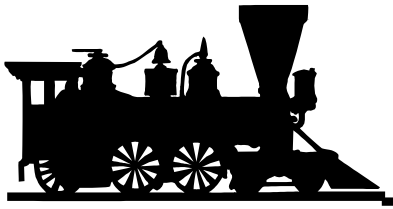
- How do other technologies address this issue.
- Computer behaviour influenced by humans or technology?
- Why HCI is so difficult.
- Summary.

Complex Systems Railways

- Liverpool & Manchester Railway opened 15th Sep 1830 (Stevenson's Rocket)
- Rapid development, no controls, many deaths.
  - E.g. Brunel and Babbage trains narrowly escaped crashing into each other
- Railway Inspectors set up 1840's

Technology Challenges

Train movements based on Time



Leaves Bristol
12 noon

Single Track
London-Bristol 2 hours

Leaves London
2:10pm

What's the problem?

Solution: Change Time

November 1840 Great Western moved to GMT
Sep 1847 all Railways recommended to use GMT
UK set clocks to GMT 1880
US Standardized time in November 1883

Improvements in railway safety

- Improving the quality of components that make up the system
  - trains, carriages, signalling, tracks etc.
- Focus on human element
  - Training for all personnel.
  - Training certification
- Railway inspectors review total system after a crash
  - Analysing the interaction of components
  - Analysing whether external factors impact accident.
  - Make mandatory recommendations.

Current status of trains

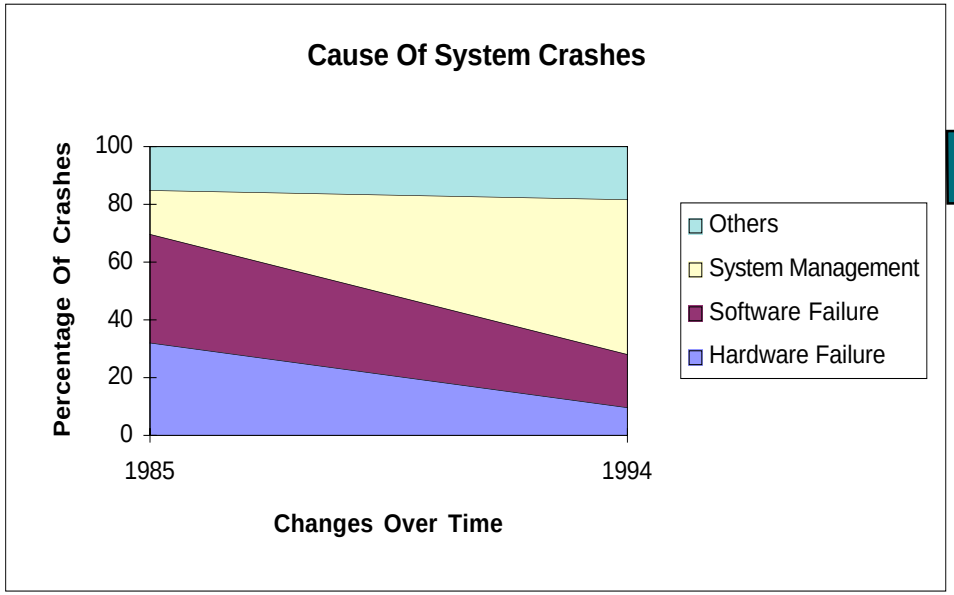
- Trains are now very safe
  - Death Rates in UK >0.5 deaths per 1 billion miles.
- But are they inflexible and expensive to build?
- Movement from trains to cars
  - Death Rates in UK 3,500 deaths per 1 billion miles.

Traditional Engineering approach to safety

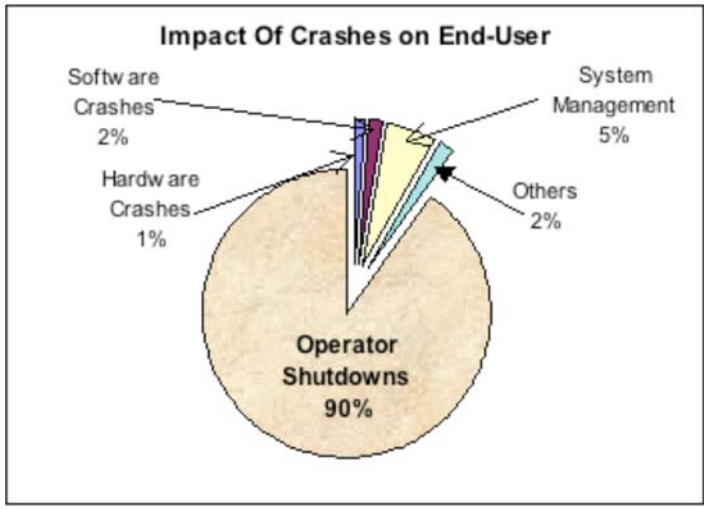
- Use of models based on theory verified through historical validation.
- Standardization & Certification.
- Making the problem fit the solution.
- Very conservative, reluctant to accept new ideas/implementations.

Product Behaviour

VAX Behavioural Drivers

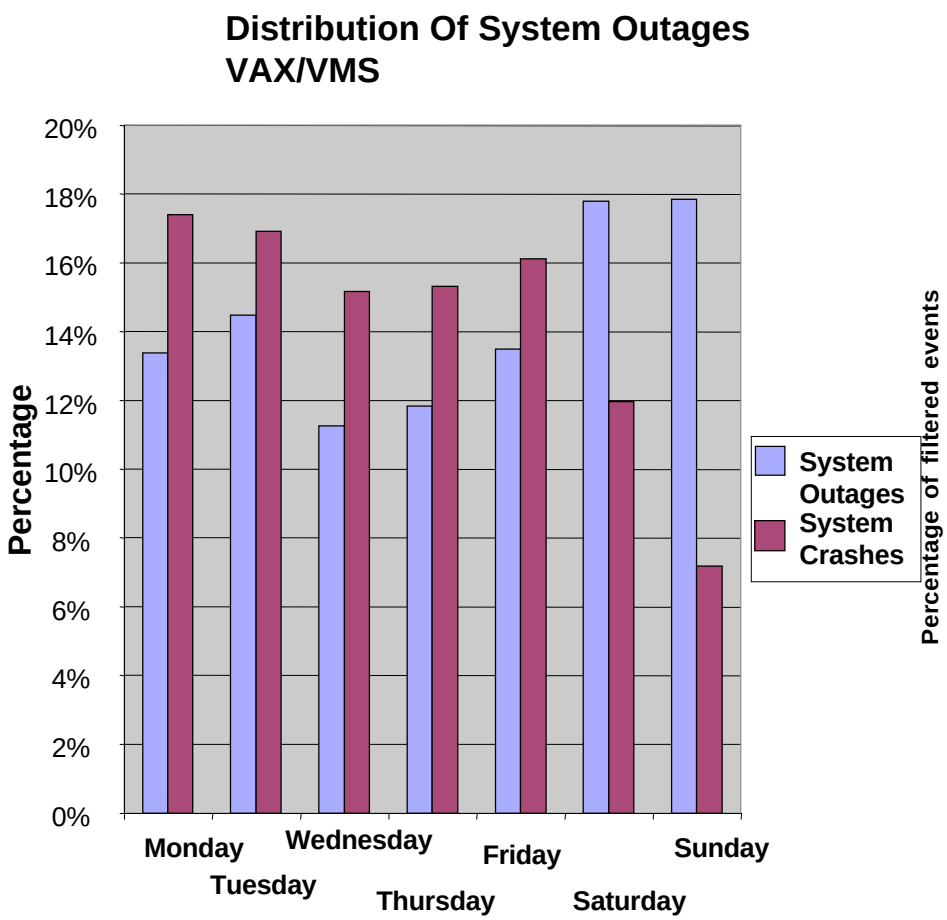


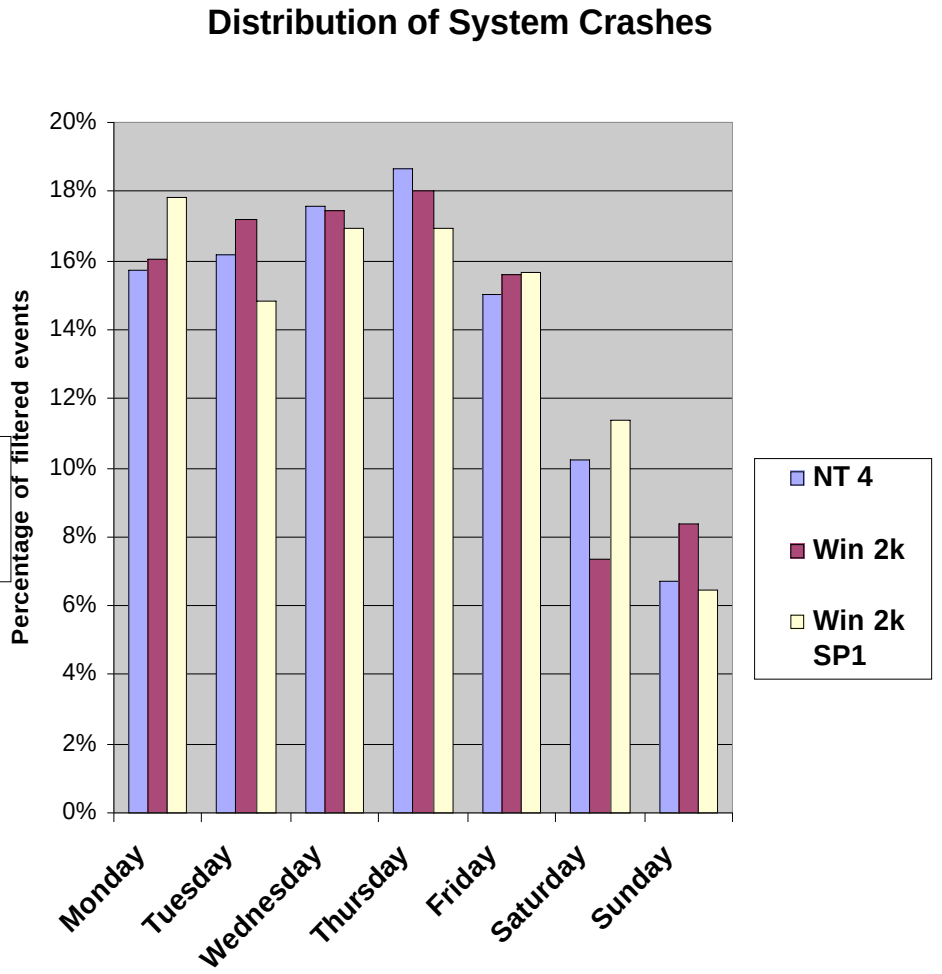
Impact on Availability?



Reliability vs. Maintenance Events

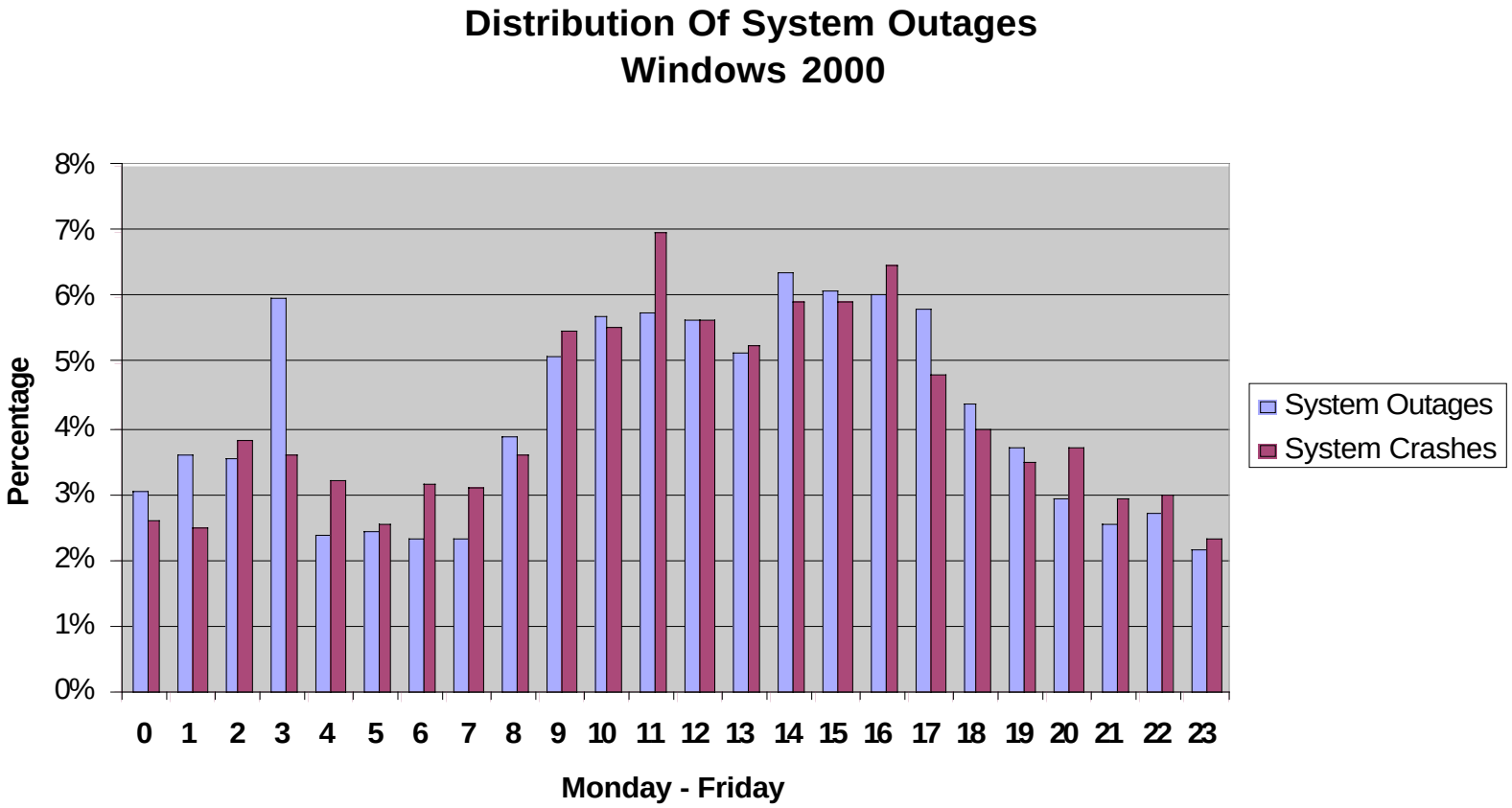
Differentiation by Day?





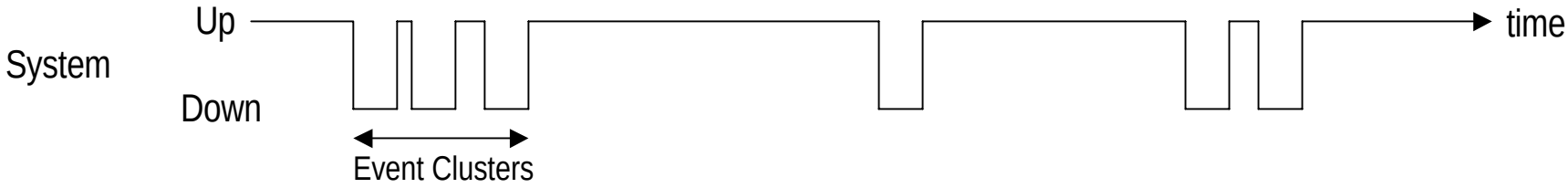
Reliability vs. Maintenance Events

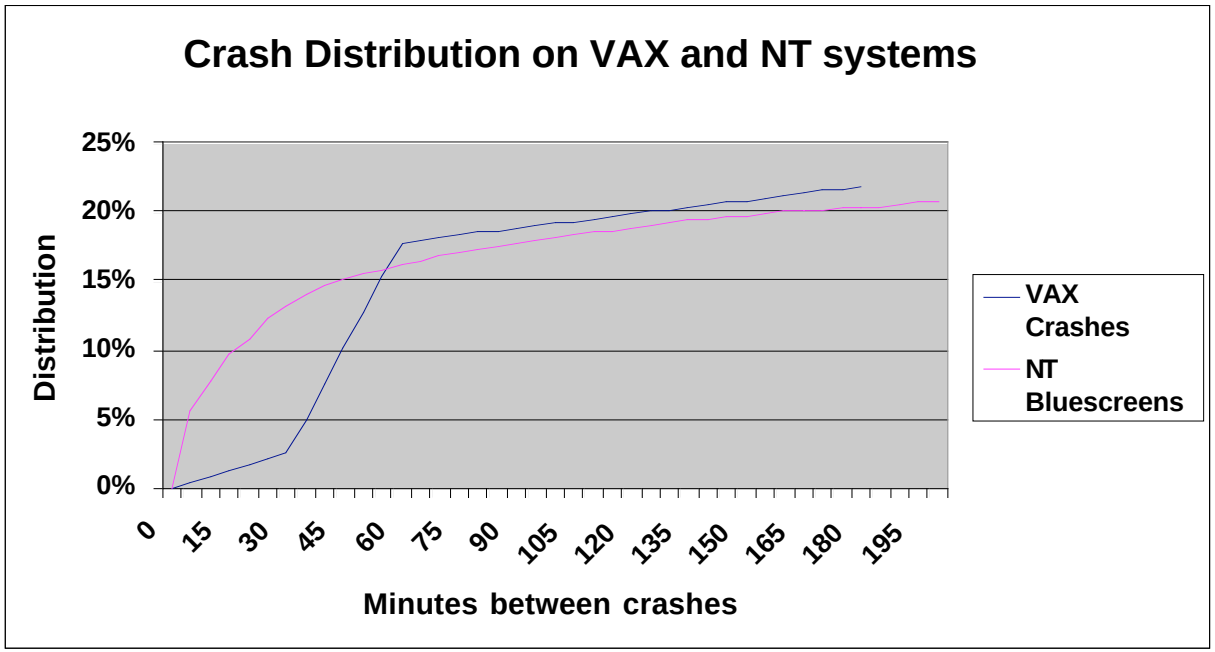
Differentiation by Day?



System Instability on single servers

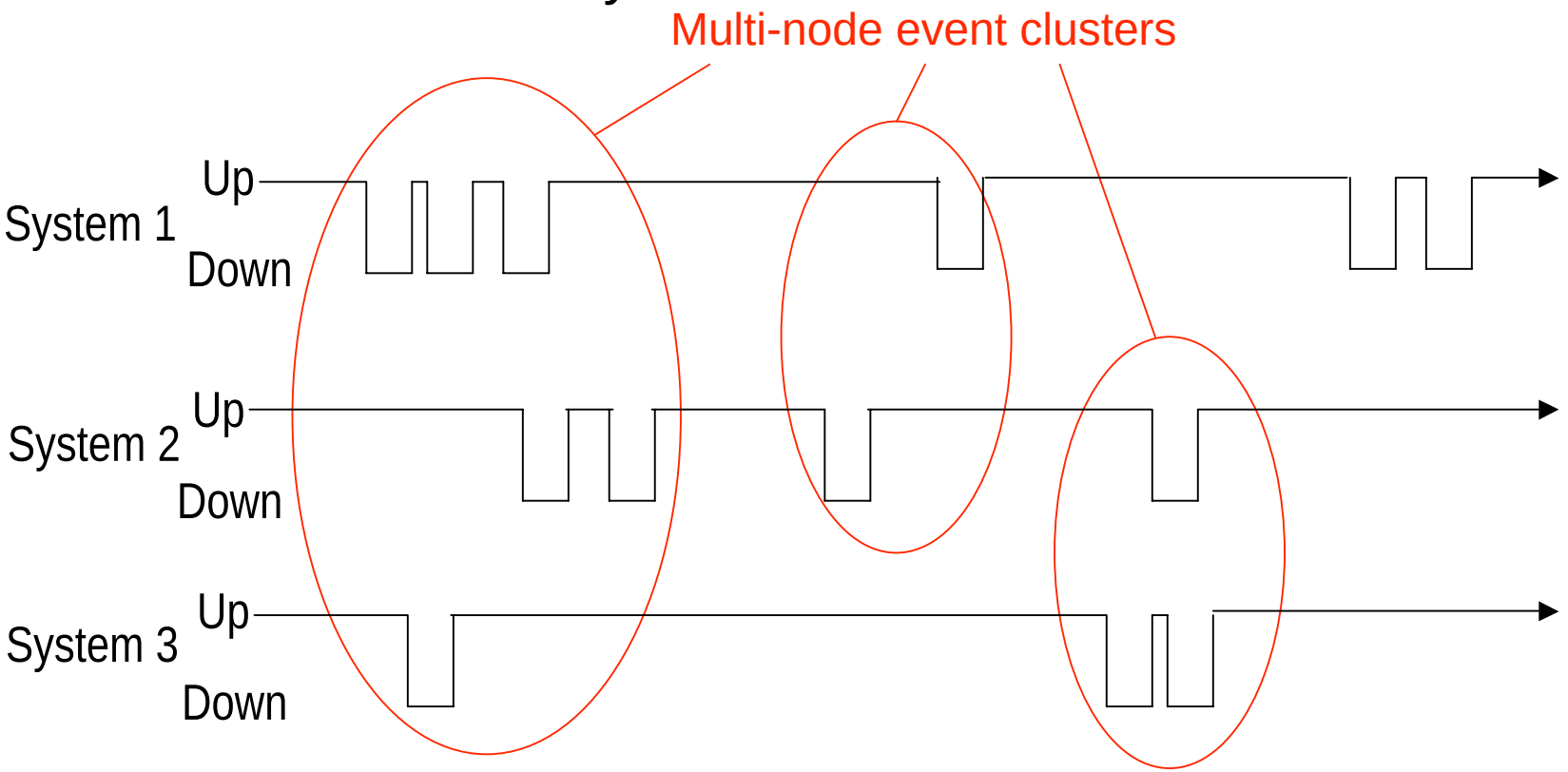
- A system crash may induce subsequent crashes.
- Behaviour first characterized in the early 1980s.





System Instability on distributed systems

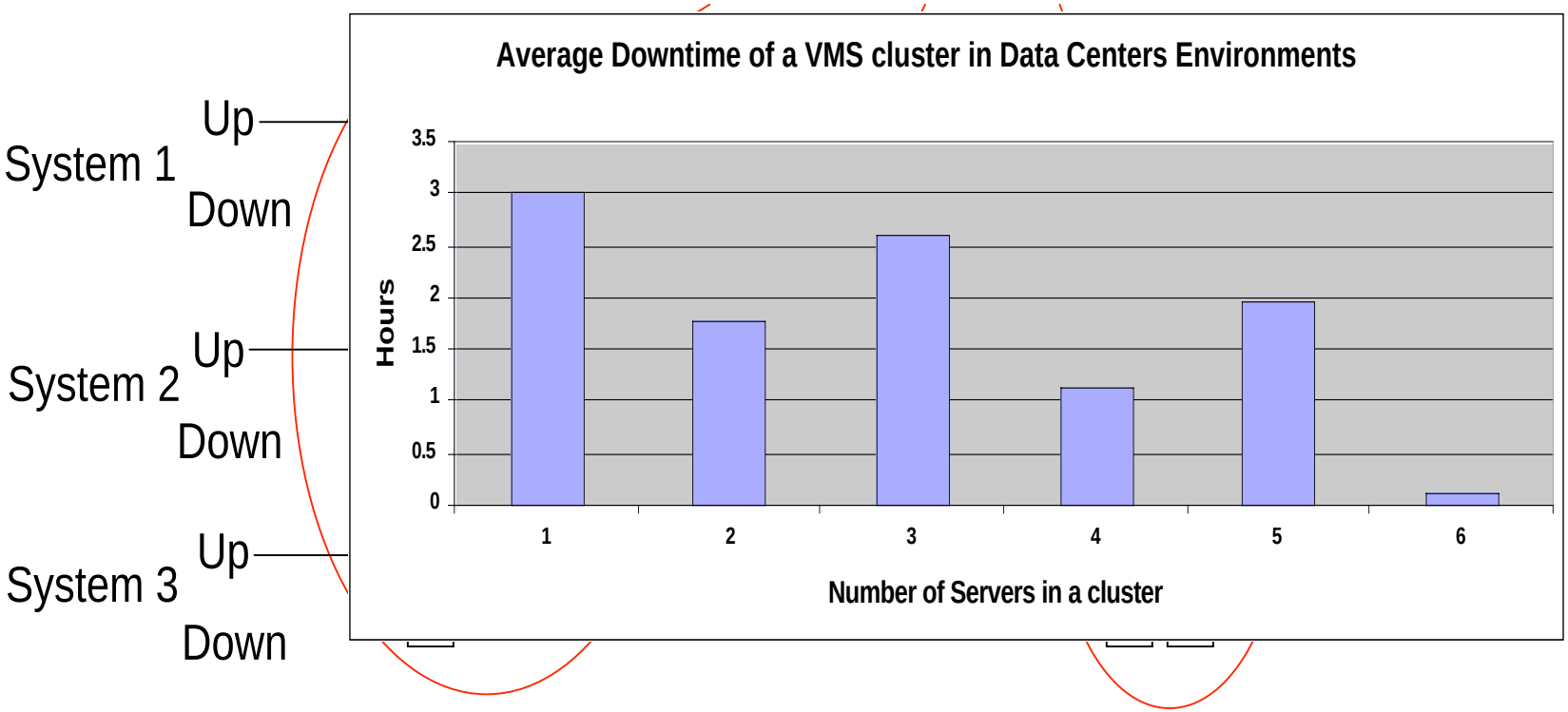
- Configuration impacts dependability.
  - Affected by technical and human factors.



System Instability on distributed systems

- Configuration impacts dependability.
  - Affected by technical and human factors.

Multi-node event clusters



Techniques for improving Software Quality

- Formal Specifications.
- Use of strong type checking languages.
- Formal development processes (e.g. SEI 5).
- Use of software verification tools.
- User training.

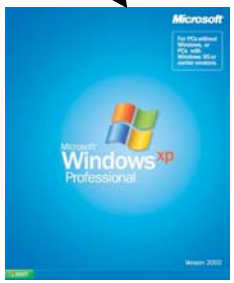
Techniques for avoiding HCI errors

- Fully understand the users specifications
- Bound the product based on those specifications.
- Develop the UI based on User Modelling, Intelligent User Interfaces, HCI design and visualization, psychological studies.
- Verify acceptability through trials with target groups.

The problem: How to improve the reliability of software.

Well understood
user profile
Hardware frozen

Media Center.
Standardized interface
Multi Functional
Evolving
Hardware limited but
regional variations.



Changing user needs
(home and business)
Infinite hardware options
New usages being
developed

Usage profile unknown
Flexible and adaptable
Rapid development

What Microsoft does well

- Standardized application interface
  - File always top left, help last in list.
  - Print always under file etc.
  - Short cut keys common across applications.
  - Common programmable interface.
- Do not force users to new UI (Classic view).
- Installation of OS and products for the home user.
  - Still searching for the optimum solution for installing products across multiple systems (same as the rest of the industry).

Major causes of HCI induced failures

- Not following correct management procedures.
- Not correctly configuring their systems.
- Not patching their systems.
- Using non compliant hardware/software.
- Viruses due to non patched and wrongly configured systems.

Bug collection and patch distribution system an example of HCI problems.

- Develop process to collected Windows XP failures (i.e. to fix problems we create).
- Base process on Windows 2000 failure rates and characteristics.
- Initial results of deployment
  - Failure rates a factor of 10 greater than forecast (HCI failures become prominent).
  - Failure characteristics different to Windows 2000 (lack of complex software bugs).
  - Most failures not due to Microsoft code.
    - But still viewed as Windows problems.

Bug collection and patch distribution

- Work with peripheral device manufacturers to resolve errors.
- Correct all Microsoft bugs and release patches.
- Develop Windows Update process as a user invoked service for privacy reasons.
- Problems
  - Peripheral device manufacturers assume users know they always need the latest driver.
  - Users blames Microsoft not the device manufacturers.
  - Users had better things to do than invoke Windows Update.
  - Not all patches had to go to all users.
  - The frequency of patch production overwhelmed business users.
  - Patches were too large for users with 56k modems.

Then security issues came along.

- Most security issues identified by security researchers or firms.
  - Provides free publicity to the researcher or firm.
  - Pressure from them to publish.
- Obvious way to resolve these issues.
  - Come up with patch for failure and other risk areas.
  - Test and release patch through Update ASAP.
- Result
  - Hackers reverse engineer patch.
  - Virus only start appearing after patch is released.
  - Users not updating systems.

Solution for Windows XP

- Windows update is a push rather than pull process.
- Patches are as small as possible
- Patches release in monthly batches.
- Not all patches are released
  - Distribute uncommon patches via crash reporting process.
- Develop a process for business for patch distribution.
- Windows XP SP2 will automatically configure the system for security.
  - May break some applications (e.g. Kazaa).
- Architectural changes in Longhorn to assist in patch distribution.

Summary.

- Perfect Reliability of complex systems is probably impossible.
  - Normal accidents, Charles Perrow
- The computer is a component of the complex systems.
  - Becoming smaller all the time.
- The system is the interaction of humans and technology.
  - Wizards and usability have to replace the need for training.
- Computer software rarely has a single objective.
  - Can wizards and usability address near infinite flexibility.
- Reliability is one of many system attributes.
  - The problem should define the relative importance of reliability.
  - Higher reliability inevitably decreases product flexibility.
- Is research focusing on the problem?