ASSESSING EMERGENT TECHNOLOGY PERFORMANCE

Assessing IT risk and opportunity

Overview

I. Introduction: Emergent Technology Issues

II. Explanation: Web of System Performance

III. Validation: Evaluating WOSP

IV. Application: Applying the model

V. Conclusions: Key Point Summary

Part I: Introduction

Issues in Emergent Technology

Separating Progress and Hype

Emergent Technology

- ***Unproven*** technology that ***could*** significantly change things:
  - Some ETs emerge *strongly:* e.g. e-mail, cell phones
  - Some ETs emerge *slowly:* e.g. the Internet
  - Some ETs emerge *briefly:* e.g. microfiche
  - Some ETs emerge *weakly:* e.g. video-phones
- ET is potentially lucrative and risky
  - It pays to predict the future
  - It is expensive to buy every new thing

False predictions

- *Past false predictions:*
  - *Leisure society* - Many now work a 6 day week!
  - *Paperless office* (Toffler) – Now use more paper than ever!
  - *AI will replace people* – still have human telephone help!
  - *Programmerless programming* (Martin)- demand still rising!
- **Errors can cost:**
  - *Video-conferencing*– despite millions $ invested, it has not yet happened – investors lost money
  - *A multi-media Internet* – billions invested in bandwidth for an Internet that is still mainly text/simple graphics. We download MM files but use them offline. People lost money.
  - *E-commerce* – The DotCom bust of 97-01 was when investors assumed e-trade would dominate. It is still < 10% of all trade.

Leading Edge Practice

- **YET** "unpredicted" products succeed:
  - *E-mail*: Assessed as a simple "lean" text messaging system
  - *Chat rooms*: Assessed as a simple non multi-media product
  - *WWW:* Assessing as unworkable without centralized control
  - *HTML*: Experts assessed it as a too-simple tag language
  - *Cell phones*: Originally thought to be a yuppy toy
  - *Google*: A "simple" search engine (unlike multi-mediaYahoo)
  - *E-Bay*: Completely off the the IS/IT theory radar

Those who predict emergent technology make money

Bleeding Edge Theory

- ***Media Richness Theory*** (Daft & Lengel, 1986)
  - Proposed 20 years ago that "rich" media create rich meaning
  - Billions invested in *multi-media* technology
  - YET, many last decade successes were not multi-media, e.g. *blogs*, *e-mail*, *chat*, *text-messaging*, *wikis*, *reputation systems*
- ***Technology Acceptance Model (TAM)*** (Davis, 1989)
  - Proposed 15 years ago that *Ease of Use* was as important as *Usefulness* in end-user technology acceptance, YET,
  - *Mr Clippy*, Word's friendly Assistant, was intensely disliked
  - Microsoft spent millions making Windows more reliable not more usable

Old theories are incomplete

The Challenge

- For today's emerging technologies:
  - ubiquitous computing and smartware, online learning and grading, online groups/project teams, online spaces/places, online agents/avatars with personalities, genetic programming, AI text readers/translators, e-paper, e-books, e-noticeboards, AI chat-bots or stock crawlers, etc, etc
- Companies face bewildering choices, some products:
  - Succeed despite resistance
  - Fail despite marketing hype
  - Sit 10-20 years in limbo, then flourish
  - Never emerge

Which is which?

Two Options

1. *Ignore Theory:* Why bother with useless theory?
   - *If the less one knew of rocket theory the better one's rocket flew, who needs theory?*
   - Use common sense and trial and error
   - *YET w*ould one run a nuclear or space program by trial and error? Complex system errors are costly!
   - Should we develop the technical infrastructure of a ***Future World Society*** by trial and error?
2. *Improve Theory*: *Theory distills experience*, avoids costly mistakes, and prevents predictable errors

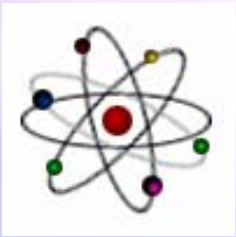
IS theories need improvement!

A New Approach

- **What is a *system?***
  - *General Systems Theory* approach (Bertalanffy, 1968)
  - Physical systems, idea systems, chemical systems, social systems, etc, are all systems, so common principles apply
- **What is *performance?***
  - Alexander's "*Notes on the Synthesis of Form*" (Alexander, 1964)
  - Performance is multi-dimensional, so systems progress in many ways, e.g. laptops are less powerful but more mobile

WOSP is a multi-dimensional general systems theory model for IS/IT

What is a system?



Four levels of IS/IT "system" **(Alter, Grudin, Kuutti, 1996)**

4. ***Social*:** norms, culture, laws, zeitgeist, sanctions, roles (*Sociology*)
3. ***Personal*:** semantics, attitudes, beliefs, opinions, ideas (*Psychology*)
2. ***Software*:** programs, data, bandwidth, memory, processing (*Computing*)
1. ***Hardware*:** computers, wires, printer, keyboard, mouse (*Engineering*)

Performance "emerges"

Each system level "emerges" from the previous:

1 → 2. Information emerges from Mechanics
Physical acts create information (Shannon & Weaver)

2 → 3. Meaning emerges from Information:
Information + cognitive processing creates human meaning

3 → 4. Society emerges from Meaning: A society's culture is common personal meanings that continue over time

- Levels are different <u>views</u> not different systems!
- Higher levels have:
  - *Higher requirements*
  - *Higher benefits*

–e.g. a *just* society (requirement) is more *productive* (benefit)

Levels combine

Social

Personal

Informational

Mechanical

Hardware system

Software system

Human-computer interaction (HCI) system

Socio-technical system (STS)

Social-Technical Systems

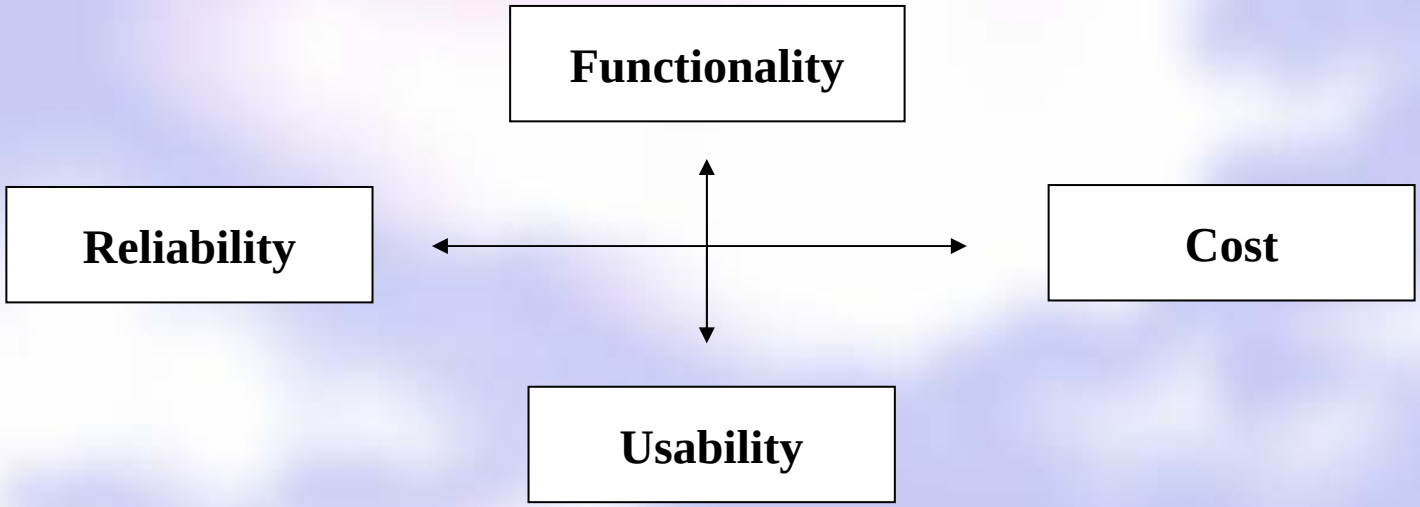
STS = social system built on a technology system

1. *Hardware system* = physical exchanges
2. *Technology system*= H/W + information exchanges
3. *Human Computer Interaction System*
   HCI = Technology + cognitive exchanges (1:1; 1:M)
4. *Social-technical System*
   STS = HCI + group normative exchanges (M:N)

The WOSP model applies to any system level

What is performance?

- Alexander's "design tension" ideas are now used in "Pattern Theory".
  - For example: A vacuum cleaner is a design "form" in the following *problem design space*:



WOSP defines the IS/IT performance space

Non-Linear Evolution

- Successful biological systems range from simple viruses to powerful predators
  - Not just the strong are “fit”
  - Systems balance performance dimensions differently
  - One sided “excellence” can cause extinction!
- Information systems have a virtual “evolution”
  - Mobile laptops vs powerful desktops
  - “Light” background utilities vs “heavy” mainline suites
  - Open source public systems vs proprietary private systems

WOSP implies non-linear system evolution

System performance is?

How well a system interacts with its environment

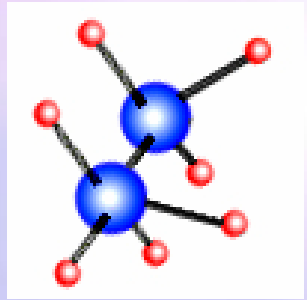
- An environment offers a system:
  - **Opportunity** – for benefits or gains
  - **Risk** – of damage or loss
- All performance is relative to the current local environment

Part II: Explanation

The Web of System Performance (WOSP) Model

Performance is excellence plus balance

System elements



- Assume four general system elements:
  - *Boundary:* monitors system entry/exit
  - *Internal structure:* supports and controls
  - *Effectors:* generate output effects
  - *Receptors:* process input signals
- Examples:
  - *People*: skin, brain and organs, muscles & senses
  - *Computers*: case, mother-board architecture, peripheral output and input devices

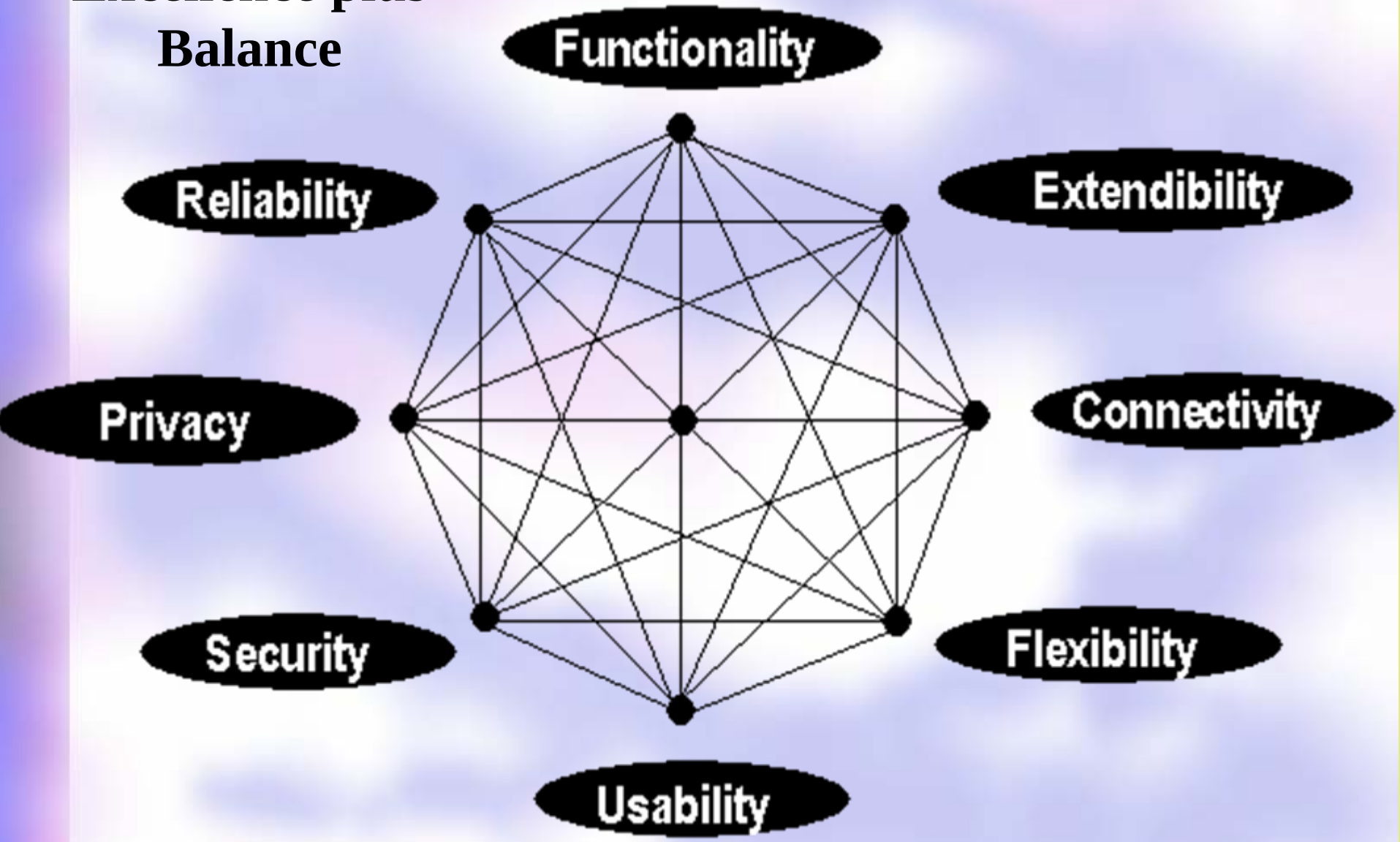
Environment Responses

- Assume two natural system *goals*:
  - *To gain opportunity benefits*
  - *To avoid risk and system loss*
- Business *opportunities* are “intangibles”, as their loss is unknown
- Intangibles are a cause of IS/IT failure,
  - e.g. Visicalc lost its spreadsheet supremacy by opportunities lost rather than errors made

WOSP performance goals

- System elements can aim to gain value or avoid loss:
  - *Boundary*
    - To enable useful entry (*extendibility*).
    - To deny harmful entry (*security*).
  - *Internal structure*
    - To accommodate external change (*flexibility*).
    - To accommodate internal change (*reliability*).
  - *Effector*
    - To maximize external effects (*functionality*).
    - To minimize internal effort (*usability*).
  - *Receptor*
    - To enable meaning exchange (*connectivity*).
    - To limit meaning exchange (*privacy*)

Excellence plus Balance



Functionality

- *Aim:* To act directly on the environment to produce a desired change
- *Also called*: Capability, effectualness, usefulness, effectiveness, power, utility
- *What tasks* can the system do?
- *Pro:* All purpose, can always "get the job done"(e.g. Office Suites)
- *Con:* Feature creep creates "bloatware" with large complicated menus. Most people can use only 50% or less of its potential

Usability

- *Aim:* To minimize relative resource action costs
- *Also called*: Ease of use, simplicity, user friendliness, efficiency, accessibility
- *How easy* is it to operate/run?
- *Pro:* Intuitive, so needs less training, help & documentation - easier to distribute
- *Con:* May lack power at a critical moment.

Reliability

- *Aim:* To continue operating despite internal changes like part failure
- *Also called*: Stability, dependability, robustness, ruggedness, durability, availability, maintainability
- *How often* can it perform?
- *Pro:* Long lasting (lifetime warrantees are economic) predictable (users can plan), fails slowly (degrades) and recovers quickly (undo/back operations reverse errors)
- *Con:* Redundant parts make it hard to change (have to change every part) and recovery can undo changes

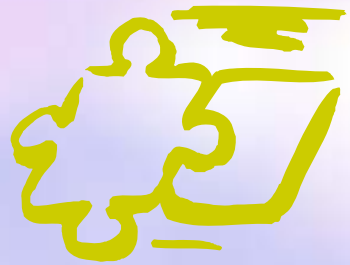
Flexibility

- *Aim:* To still work in (adapt to) new environments
- *Also called*: Adaptability, portability, customizability, plasticity, agility, modifiability
- *Where* can it perform?
- *Pro:*
  - *Mobile* computing - people use software anywhere,
  - Can *learn* usage trends (e.g. cache prediction algorithms)
  - *Customize* to individuals (e.g. control panel),
  - *Demand driven* (e.g. CSMA/CD Ethernet vs polling networks)
- *Con:* One change can affect interconnected, holistic systems catastrophically!

Security

- *Aim: To prevent unauthorized entry, misuse or takeover*
- *Also called*: Defense, protection, safety, threat resistance, survivability
- *Who* controls the system?
- *Pro:* Resistant to external attack/entry. Secure software/data resists hackers/viruses
- *Con:* Central control & strong boundary checks reduce opportunities from openness, usability, connectivity and flexibility (e.g. many logons)

Extendability



- *Aim: To use outside elements*
- *Also called*: Openness, interoperability, permeability, compatibility, scalability, tool use
- *What* can the system use?
- *Pro:* Open source code allows third-party "plug-ins"
  - Open IBM PC vs closed McIntosh. Early McIntosh was more reliable/usable, but the PC was more extendable!
  - General hardware port *standards* allow extendibility, e.g. USB
  - Clip and paste lets one application use another's data
- *Con:* Allowing boundary flows/connections by foreign entities can negatively impact the system

Connectivity

- *Aim:* To communicate socially with similar systems
- *Also called*: Networkability, communicativeness, interactivity, sociability
- *Who* can we communicate with?
- *Pro:* Connected systems can be:
  - Informed by other systems - “group aware”
  - Up to date by download means lower “cycle times” (user complaint cycles, software patch cycles etc faster)
  - Connect by many channels (multi-media), two-way channels (reciprocal), or high linkage (many-to-many)
- *Con:* Information overload/distraction

Privacy

- *Aim:* To control the release of self information
  - If we own ourselves (freedom) we own our data (privacy)
- *Also called*: Confidentiality, secrecy, camouflage, stealth, social rights, information ownership
- *Who sees us*?
- *Pro:* Protect credit card data misuse by encryption.
  - Privacy apps are popular downloads, e.g. Zone Alarm
  - Need privacy to regenerate (e.g. sleeping, "packing" a file)
  - *Social rights* to information (e.g. digital signatures copy protection)
- *Con:* Reduces information flows

Summary

An application is not “high performance” if it is:

1. *Ineffectual* – it cannot do the job
2. *Unusable* – you cannot make it work
3. *Unreliable* – it breaks down often
4. *Insecure* – it succumbs to viruses
5. *Inflexible* – it fails when things change
6. *Incompatible* – it cannot use standard plug-ins or data
7. *Disconnected* – it cannot communicate
8. *Indiscreet* – it reveals private information

Part III: Validation

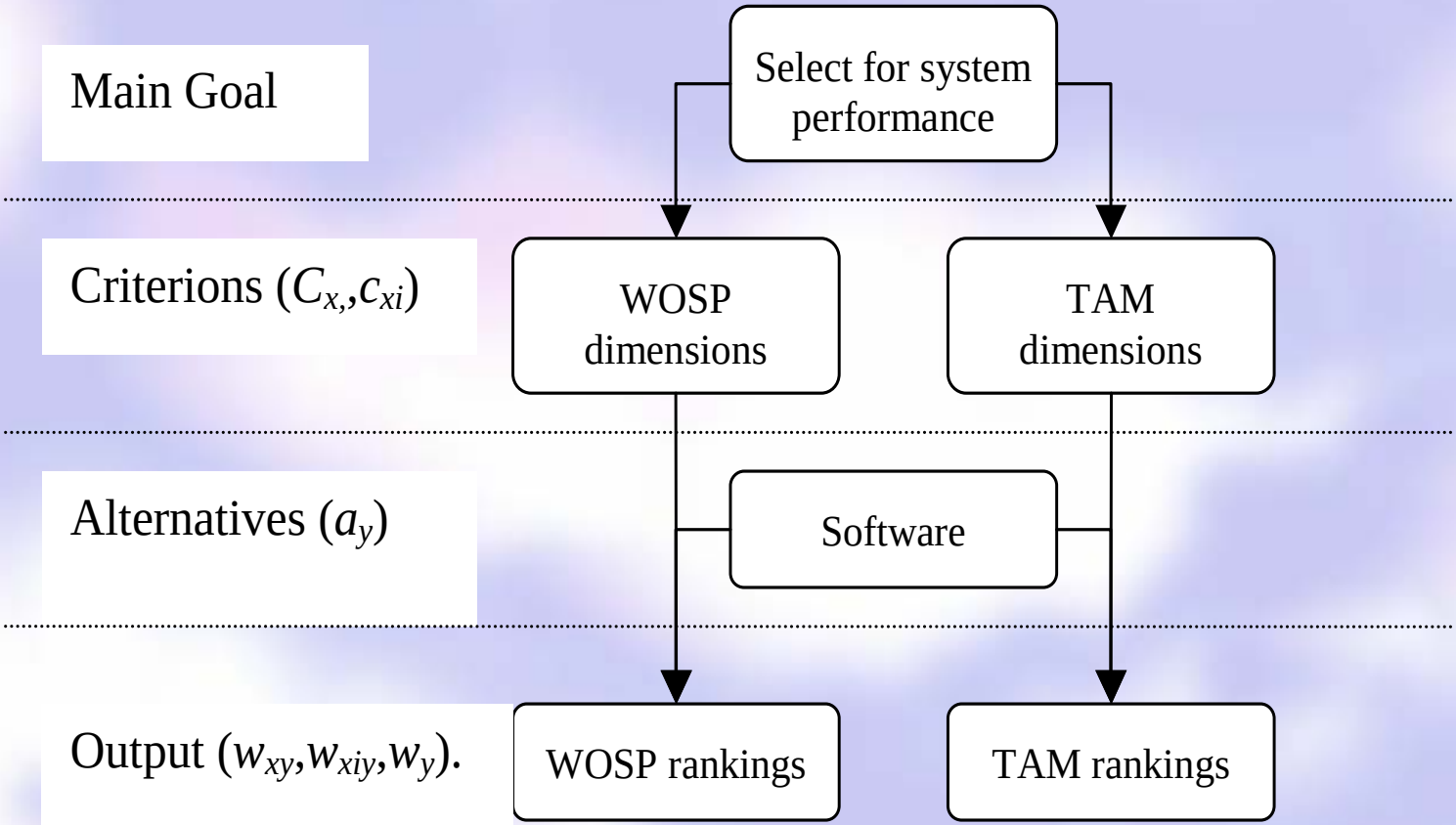
Evaluating WOSP

IT evaluation is more than functionality + usability

Validation Studies

- **AHP Study**: Subjects used Analytical Hierarchical Processing to rate 3 browsers via two TAM criteria and eight WOSP criteria, then compared the two methods
- **Conjoint Analysis Study**: Subjects were asked to rate and rank 30 browsers for their company, given different WOSP ratings for each

AHP Design



AHP TAM vs WOSP

Effect	N	TAM	WOSP	df	MS	F	p
Confidence	20	3.7	2.4	1	16.9	12.8	.002*
Accuracy	20	3.6	2.45	1	13.2	12.4	.002*
Completeness	19	4.21	2.42	1	30.4	16.8	.001*
Outcome Satisfaction	18	3.14	2.35	1	5.64	9.63	.006*

Subjects were more satisfied with the WOSP decision, felt more confident using WOSP, and felt WOSP was more accurate and complete

CA & AHP Criteria Ranks

For a Browser

CRITERIA	AHP WEIGHTS		RANK	
	LOCAL	GLOBAL	AHP	CA
Security	0.79	0.17	2	1
Privacy	0.64	0.20	1	2
Usability	0.58	0.09	3	3
Functionality	0.42	0.12	6	4
Reliability	0.57	0.12	4	5
Connectivity	0.36	0.11	5	6
Flexibility	0.43	0.09	7	8
Extendibility	0.21	0.04	8	7

Security & Privacy ranked before Usability & Functionality!

Part III: Application

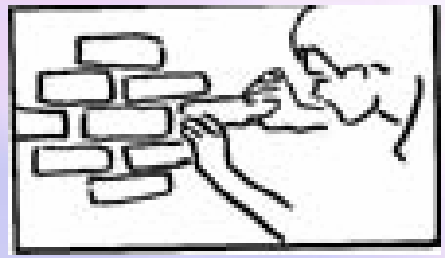
Applying the model

The Devil is in the Details

Performance

- WOSP *area* estimates overall performance
- WOSP does not distinguish functional requirements and non-functional requirements (NFRs)
- If a system fails because:
  - –It cant do the job (poor functionality)
  - –Users cant work it or don't like it (poor usability)
  - –It breaks down (poor reliability)
  - –It is destroyed by a virus (poor security)
- The end result is the same- *the system does not run*!

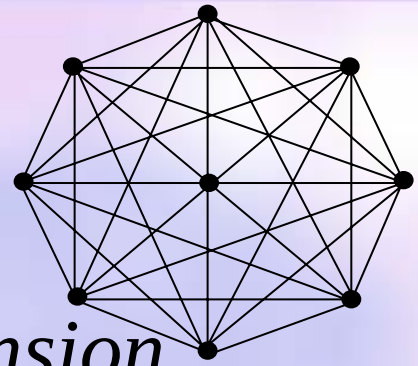
Modularity



- The WOSP goals are conceptually *modular*
- Independent definitions allow any combination:
  - A bullet-proof plexi-glass room is secure but not private
  - Perfect encryption allows data theft (insecurity) without privacy loss
  - Reliability engineers are *service providers* while police security are *service deniers* [Jonsson, 1998]
- AHP study correlations support this:
  - Privacy/Security correlation = +0.01
  - Extendibility/Connectivity correlation = –0.29

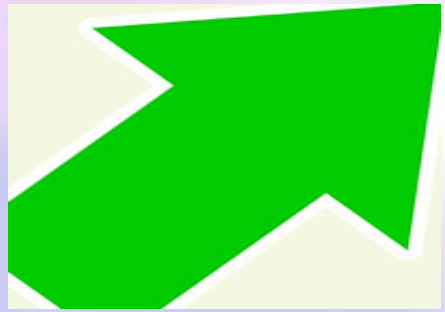
    Usability/Reliability correlation = –0.16

Tensions



- *Each WOSP line is a cross-cutting tension*
- Each has a different tension value
- Increasing one performance aspect can make others "bite back" (Edward Tenner)
  - More functions means more menu options and more for users to learn
  - More "open" wireless systems are less secure
  - More border security can reduce positive entry
  - A flexible control panel can reduce reliability, e.g. when users change screen settings

Progress



- Experts predict based on the past
- Past “strong” dimensions have greater tensions with other factors
- *One-dimensional progress gives diminishing returns*
- IS progress moves “unexpectedly” to performance dimensions with less tension:
  - A decade ago gaming went multi-media
  - Last decade developed not more MM but *social gaming*, e.g. The Sims and MMORPGs (massively multiplayer online role-playing games)

Progress is like a train on many tracks, not one track

Innovation

- Innovation can *reconcile performance goal tensions*
  - Clip/paste **text** and clip/paste **graphics** are different functions, but learning one gives both
  - Interface consistency reduces the functional/usability conflict
- Innovation gives two+ dimensions for the price of one!
- Hence we admire :
  - Mathematical *Elegance* = Functionality + Simplicity
  - Government *Discrimination* = Security + Openness
  - *Legitimate* social rights = Connectivity + Privacy

Innovation reconciles apparent performance "opposites"

Example

- In 1992, Apple CEO John Sculley introduced the hand held Newton, saying portable flexibility was the wave of the future – **he was right**
- But the Newton's portability reduced data input usability, and in 1998 Apple dropped the line due to poor performance
- When Palm's Graffiti language improved hand-writing recognition, PDA market took off again

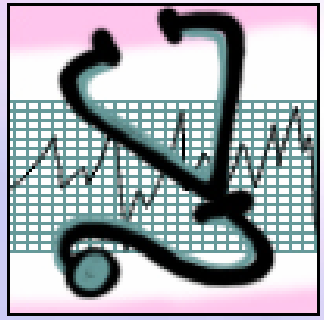
Innovations may need combination advances

Killer applications

- The basics of most “killer applications” can be coded and tested in less than a week:
  - Email, browser, chat, reputation systems, wikis …
- Why are killer apps so functionally simple?
- *Functional simplicity allows WOSP expansion*
- If initially the web is “slack”, it can expand
- Powerful functions like video e-mail are more difficult to support all-round

Balanced performance wins

Evaluation Method



Evaluating emerging technology :

1. Assess environment
2. Create WOSP profile
3. Define levels
4. Evaluate choices:
   a. General → Short List
   b. Detailed → Priorities
5. Make a business decision

1. Assess environment

- Different environments favor different profiles, so performance has no "perfect" form
  - *Opportunistic* environments give benefits to systems that can reap them: functionality reliability extendibility & connectivity
  - *Hazardous* environments harm systems that cannot handle danger: security, privacy, usability, & flexibility
  - *Dynamic* environments change risk and opportunity quickly: flexibility, usability, & connectivity
- Note: Can be opportunistic and hazardous

WOSP profile should "fit" the environment

2. WOSP Profile

Dimension	Detail	Value %
Functionality	Must do a complex job or be very accurate?	
Usability	Must be easy to learn and use ?	
Security	Must resist outside attack/take-over?	
Extendibility	Uses outside components/data?	
Reliability	Must avoid/recover from failure?	
Flexibility	Must be mobile or adapt to new contexts?	
Connectivity	Must communicate or download data?	
Privacy	Must protect personal/corporate data?	
Performance	***To interact successfully with environment.***	**100%**

Not all WOSP criteria always apply: Default= 12.5%

Example 1

Criteria	Value
Functionality	30%
Usability	40%
Flexibility	0
Reliability	30%
Security	0
Extendiblity	0
Connectivity	0
Privacy	0
Total	**100%**

Fast-food POS S/W:

- Needs easy to learn software as staff come and go quickly
- May trade functionality for usability
- Must not break down

Don't pay for performance you don't need

Example 2

Criteria	Value
Functionality	12.5%
Usability	10%
Flexibility	20%
Reliability	20%
Security	17.5 %
Extendiblity	0
Connectivity	0
Privacy	20%
Total	**100%**

Medical records system

- Must be mobile, as doctors visit patients
- Medical data must be private and secure
- Highly skilled doctors do not need very "friendly" systems

Mobility and privacy are hard to combine

3. Define levels

- For example, reliability:
  - *Hardware reliability*- part failure
  - *Software reliability* – program hangs and reboots
  - *HCI reliability* – System operator makes errors
  - *Social reliability* – Warrantees and guarantees of replacement and support (e.g. Dell)

WOSP LEVEL EXAMPLES			
Dimension	**Hardware**	**Software**	**HCI/CHI**
Functionality	Chip capacity, memory	Output change rate, frames/sec	Task ability eg to change documents
Usability	Heat, power consumption,	"Lite" s/w, less cpu resources to run background	Intuitive s/w, need no manual/training
Security	Sealed, secure, insulated	Firewall, virus checks	UserID/Password, bio-id
Extendibility	Standard plugs & connections,	S/w compatible with other s/w	User plug –ins, extensions
Reliability	Uptime, easy to repair	Error recovery	Reduce operator errors
Flexibility	Switchable, e.g. 110-240 volts	Platform independence	Control panel, set language
Connectivity	Network card, comms outputs	Bandwidth, no of connections	Can exchange meaning (email)
Privacy	Shielded, tempest proof	Encrypt PINS in online buying	Anonymous web surfing

Example

- An “excellent” network security system can:
  - *Reduce usability*: If it cancels passwords not changed every three months, increasing help desk load as staff call in with failed passwords
  - *Reduce connectivity*: If it limits access to/from outside the network, e.g. cannot logon from home
  - *Reduce functionality*: If you can only logon via the physical PC in your office, staff cannot do classroom demos
- “Excellent” security can reduce performance!

4. Evaluate choices

- Having defined a WOSP performance "space" for your business for a given product type
- Evaluate "best-fit" choices in two phases:
  a. **General evaluation,** to give a "short list"
  b. **Detailed evaluation**, to prioritize the short list

a. Caution: Some evaluation methods take so long the choice passes them by

3a. General Evaluation

- Rate on three or more *performance criteria statements*, questionnaire issued widely *e.g.* for a browser:
  - Privacy: It stops web sites reading my email when I browse
  - Security: It detects/prevents spyware from installing
  - Functionality: It is all I need to search and display the Internet
  - Usability: The user interface is consistent and easy to learn
  - Flexibility: It has a "control panel" to change browser settings
  - Reliability: Even if I do many things at once, it still works well
  - Extendibility: It works with third party tools, like real-player & flash
  - Connectivity: If a download fails it can be restarted again later

Scale: 1. Strongly Agree 2. Agree 3. In the middle 4. Disagree 5. Strongly Disagree

Applications may have different performance statements

3b. Detailed Evaluation

- *Detailed evaluation* by specialists, analyze by AHP
  - *Functionality*: Output testing via test-bed *(main module)*
  - *Usability*: Usability testing *(interface, help, wizards)*
  - *Reliability*: Load and error testing *(error & recovery code)*
  - *Flexibility*: Portability testing *(control panel, preferences)*
  - *Extendibility*: Compatibility testing *(plug-in manager, import/export)*
  - *Security*: Threat & penetration testing (*logon/register module*)
  - *Connectivity*: Internet/network testing *(channels and network)*
  - *Privacy*: Social legitimacy testing *(rights control module)*

4. Make business decision

- WOSP assesses the *system* at a *moment in time*
- Also assess:
  - *Creation/purchase costs*: Capital investment, return on investment, purchase options, etc
  - *Existing infrastructure*: Current facilities, s/w platform (Windows vs Sun), staff and procedures
  - *Legal and governmental* requirements
  - *Normative effects*: A weaker application may better if everyone uses it, e.g. VHS vs Beta

WOSP helps make balanced technology decisions

Part IV: Conclusions

Tips and Trends

We want it all, and we want it now
(Queen)

Tips

- *Level*: Consider social levels (HCI, STS)
  - Human/social levels will soon drive technology
  - Internet is now more a social system than a technical one
- *Balance:* Success needs many causes but failure needs only one. WOSP is a *checklist*
- *Profile:* What is your desired performance profile? – your top 3-4 criterion goals are?
- *Statements*: Develop practical criterion statements to assess particular applications quickly
- *Cross-cutting effects:*enhancing one feature can reduce another

Trends

- *Higher levels:* As IS/IT becomes more complex, higher human/social levels will become more critical
  - The last decade was about person-to-person communication
  - The next decade will be about group-to-group communication, e.g. reputation systems, wikis, online voting and democracyetc
- *Integration:* Initially the web is “slack”, so any advance is progress, but as tensions grow, innovation is needed
  - Combination advances not just single advances
  - Each new advance must be integrated not just added
- *Privacy is the sleeper*: New “smart” location devices will tell when you are in the restroom, and which cubicle!

People & Technology?

- TV and Films speculate:
  - *Will machines replace us?*
    the Terminator, IBM's Deep Blue, AI the movie.
  - *Will machines control us?*
    From the industrial dark ages to the Matrix.
  - *Will machines join with us?*
    Star Trek's Borgs, Star War's Darth Vader.
- Are "smart" machines stupid?
  - They require a context, while people can *create contexts*
  - Technology systems have evolved for <100 years
  - Human systems have evolved for >3,000,000 years
  - WE ARE THE SENIOR PARTNER

Technology must support human processes

Focus on the human process

Human Process		Technology
– 1:1 conversation	→	Email
– Many senses	→	Multi-media system
– Trading	→	Ebay type systems
– Foraging	→	Browser design
– Group conversations	→	Chat
– Associative memory	→	Hypertext
– Feedback learning	→	The wonderful back button
– Normative behavior	→	Reputation systems

Maximum efficiency?

A man had a donkey. His tavern friends told him to be more efficient, so every day he fed it one less carrot. Eventually it fell over and died.
Talking later in the tavern he sighed:

"What a pity, given just a bit more time I could have got it used to eating nothing at all"

Conclusion

Excellence requires Balance

See brianwhitworth.com/papers.html for other papers:

1. *The Web of System Performance*
2. *Polite Computing*
3. *Spam and the Social-Technical Gap*
4. *Legitimate by design*
5. *Voting before discussing*