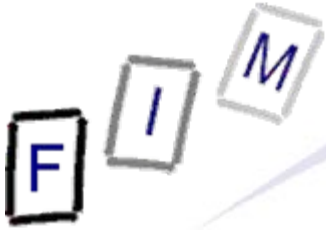
Mag. iur. Dr. techn. Michael Sonntag

Selection

Software, Providers

Institute for Information Processing and
Microprocessor Technology (FIM)
Johannes Kepler University Linz, Austria

E-Mail: sonntag@fim.uni-linz.ac.at
http://www.fim.uni-linz.ac.at/staff/sonntag.htm



Questions?

Please ask immediately!

Introduction

- Selection of operating system
  - → Strategies
- Determining the kind of software to use
  - → Total Cost of Ownership (TCO)
  - → Strategies
- Selecting external providers
  - → Access: ISP
  - → Hosting: Webservers, mail, file, ...
  - → DNS: Where to register
- Each with a small checklist of things to think of when selecting (even though for SMEs not all items will be important or even be answered by suitable providers)

Selecting OS

- Selection of software and especially OS should be as "undogmatic" as possible
  - → "Microsoft bashing" might be fashionable at schools, but for businesses less ideological reasons are much more important
  - → "Linux enthusiasts" could find resistance with users
- We will look at three OS configurations in more detail
  - → Windows: Employing Windows throughout the enterprise
  - → Linux: On desktop, servers, infrastructure
    - » Could be any other FOSS (e.g. BSD); limited applicability also for other versions of Unix
  - → Mixed environment: Using both for different sections
- Checklists for selection

Selecting OS: Considerations

- Selection should be seen from the business focus
  - → External requirements?
    - » Business partner systems, data conversion, legal framework, ...
  - → Does it enable the business functions? How good?
    - » **Required software available**, interface to other systems, ...
  - → What does it cost?
    - » Different kinds: Hardware, licenses, training, administration, ...
  - → How flexible is it?
    - » More computers/applications, new requirements, upgrades, compatibility with new hardware, ...
  - → What are the risks associated with it?
    - » Malware/security, uptime, support, vendors, misconfiguration, undesirable software (e.g. games), ...
  - → Employees views/experiences?
    - » Administrator knowledge, user's education, ...

Selecting OS: Windows Advantages

- Single vendor ensures "internal" compatibility
  - → MS software will work nicely with MS software
- Almost everyone knows how to work with it
  - → Many know how to administrate it to some degree
- Often comes preinstalled on clients and therefore "free"
  - → Guaranteed to work with this hardware
- Support and training easy to obtain
  - → Professional support may be expensive (MCP, MCSE, …)
- Easy to administer for the common use cases
  - → Graphical user interface, complex options hidden
- Everything out of a box
  - → Includes media player, firewall, anti-virus, …
    - → Quality?

Selecting OS: Windows Advantages

- Extremely wide software availability
  - → There is (almost) no software which does not run on windows
- Drivers for new hardware available fast and for everything
  - → Provided by manufacturer, i.e. best options, good quality
- (Clearly) defined licensing scheme
  - → Easy to calculate and predict (?); support for verification
- Vendor support and updates guaranteed for certain time
  - → Third-party support for a longer time available
- Wide variety of national versions (language, icons, ...)
  - → Interoperability between them guaranteed

Selecting OS: Linux Advantages

- Runs on older/slower hardware
  - → In general modest hardware requirements, except memory
- Driver support for older/"strange" hardware
  - → Support lasts longer before no longer updated
- Open for modifications/changes on every level
  - → If something doesn't work/is missing, it can be added
- Basic versions are completely free
  - → Low start-up costs
- Expert advice for free possible (!) on the web
  - → Even for esoteric topics, free support possible
- Interoperability of software generally better
  - → If it runs on Suses, it will probably run on Red Hat, BSD, Debian, *nix

Selecting OS: Linux Advantages

- Source code available for review/inspection/modification
  - → This is not as helpful as it seems… (see modifications above)
- Administration effort has slower increase
  - → High entry barrier, but then many things are similar
- Much software is also free
  - → Today often available for Windows too
- Less security risks
  - → Depends enormously on the application!
- OS itself is more stable (programs not necessarily!)
  - → If something happens to an app., the OS usually survives
- Different options: RedHat, Suse, BSD, Solaris, AIX, ...
  - → "Transfer" of programs rather/more easily

Selecting OS: Mixed environment Advantages

- Select the one best suited for each job
  - → Based on users/applications/hardware/task/...
  - → Some software runs only on specific OS or versions
- Improved overall security
  - → Attacks must work on several OS combined or sequentially
- Reuse/obtain old hardware/software/OS
  - → Significantly reduced costs possible!
- Investments in knowledge remains valid
  - → Both for users and administrators!
- "Never change a running system"
- Incremental change possible
- No lock-in to a single vendor/system
- Staff/users accustomed to change

Selecting OS: Mixed environment Problems

- More and more difficult administration
  - → Some things might have to be duplicated
  - → Something will always not work everywhere
- Requires more expertise in total
  - → Experts on several OS required
  - → Result often: Everything "working" but nothing "perfect"
- System interoperability partly difficult to obtain
- Overview on licenses/versions difficult
  - → Compliance can be a real problem….
- Detailed planning needed: Where can I install what?
- General commercial support difficult to obtain
  - → Individual support: "Problem is in the other system"
- Lower performance because of friction
  - → Conversions, standard protocols vs. private optimized ones, ...

Selecting OS: Checklist

- External requirements (specific software, partners)?
  - → Requirement for clients only or for servers/infrastructure too?
- Local expertise available/how much actually needed?
  - → Do-it-yourself? External support needed?
- Complexity of tasks?
  - → Training for non-Windows OS simple or difficult?
    - » Example: Only web-applications?
- Outsourcing of servers/security/... possible/desirable?
- Creating a new system/replacing old one/extending old one?
  - → Replacing old: Compatibility issues; incremental change
- Heterogenity and dynamics of change of requirements?
- What hard- & software is available?
- Customization needs ("standard" version sufficient)?

Virtualization

- Use one OS as the basis (host OS) and then run various others in virtual machines on it (guest OS)
  - → Duplication of administration
  - → Can bring cost reductions and flexibility!
    - » Load balancing through dynamic server migration
    - » Reduced server needs
    - » Improved system transfer and restore
    - » Run old systems on new hardware (drivers not needed!)
- Better solution: Virtualization directly on the hardware
  - → Still the same question: What guest OS to use!
- Good solution for testing:
  - → Run it on another machine
  - → Slower than when running natively

Selecting Software

- Similar to OS, software is also an important factor
- Some standard software every company will need
  - → Example: Web, E-Mail, Office
- For the actual business function (or supporting it), specific software will usually be required
  - → Production: Planning/Control system
  - → Services: Allocation of personnel/tasks
  - → Generally: Personnel, accounting
- Three general options available:
  - → (Real) Standard software: Buy and run
  - → (Customized) Standard software: Buy and customize
    - » Customization may range from small modifications to large changes, consultant needs, long introduction
  - → Custom software: (Let) Create for you specifically

Selecting Software: Basic strategies

- Use demo versions to assess suitability
  - → As far as appropriate (e.g. personnel management software might take long time to install and test)
  - → Virtualization can help here
- Assess lifetime of supplier
  - → Small/new companies might be cheaper, but continued support and development might be less ensured
- Assess size of supplier
  - → Large companies will be more willing to provide guarantees
- Upgrade possibilities
  - → Are there "larger" or "extended" versions available?
    - » Clear migration path for future extensions?
- Update cycles/portability
  - → How often can/must I update? Runs on other systems too?
- Past handling of bugs?

Please note: The price does **not** appear here!

Selecting Software: TCO

- Total Cost of Ownership (TCO): Complete costs/period
- Not only the cost of buying software is important, but also
  - → Installing and administration it, help desk
  - → Teaching users and administrators (explicit and implicitly)
  - → Updating and migrating from/to it (especially data conversion)
  - → Additional/newer hardware required
  - → Costs of downtime in case of problems
    - » Bugs, security problems, maintenance, ...
  - → Users "exploiting" the system
    - » Opportunities create needs; "Futz factor" (games etc.)
  - → Disaster prevention (redundancy, UPS, backups, ...)
  - → Productivity of users
  - → Energy, financing, consultants, disposal, air condition, ...
- These together are usually many times the license costs!

TCO: 6 main factors

- Purchase price
  - → Direct and indirect costs
    - » Hardware, OS, software (per seat); printers, servers; financing ...
    - » Recurring service gees, license renewal, …
- Training costs
  - → Formal and informal (=productivity loss) training
- Application costs
  - → Changes in existing systems for creating compatibility
  - → New systems required
- Maintenance and support costs
  - → Admin personnel, support, management of IT department
  - → Costs of problems (downtime, productivity, etc.)
- Environmental costs
  - → Network (cables + equipment), Internet connectivity,…
  - → Power, cooling etc.

Selecting Software: TCO

- TCO assessment is a complicated process
  - → Often with external consultants (obviously not for SMEs!)
  - → Can also be simplified; even then very helpful!
- Should contain all direct and indirect costs
  - → Indirect ones might be difficult: Use checklists if available
- Everything must be assessed in money
  - → Cooperation of IT and management required
  - → Might be difficult sometimes (e.g. cost of risk of downtime)
    - » What is the risk of downtime (once every year for 2 hours???)
- To be done after suitability and market research only!
  - → Market research: What offers are available; what do they exactly consist of; are some preliminarily removed?
  - → Suitability: Whether and how suitable is the solution?
    - » This must be compared with the resulting TCO at the end!

TCO: Time factor

- TCO changes over time
  - → Hardware is deprecated, but may have longer/shorter lifetime
  - → Training costs usually get lower over time
  - → Repair costs increase
  - → Productivity reduced compared to new products
- Minimum requirement: Calculating the TCO over the expected lifetime of the system!
  - → Better: Regularly reassess (e.g. yearly)
    - » This might be a reduced form, e.g. only noting changes

TCO: Problems

- TCO usually calculated per "client seat"
  - → Therefore changes in their number can significantly change the TCO if there are large fixed costs (e.g. admin training, custom software, fixed server size); also not linear
- Must be calculated independently for several solutions, not from one solution and then compared to the others
  - → See e.g. many commercial studies (Windows vs. Linux)
- TCO leaves out the actual gain (negative only!)
  - → What does this solution add to the business processes?
- Some small issues can result in huge impacts
  - → 95% availability vs. 99,9% availability, maximum wait time for support; usability/happiness of employees/customers

Cheapest is not always best!

Selecting Software: Custom

- "Purchase price" = cost of creation
  - → High: Custom software is created once only
- Training costs
  - → Low-Medium: Involvement in creation, specially suited for task
- Application costs
  - → Low: Modifications go in new and not old systems
- Maintenance and support costs
  - → High: Bug fixes more likely, no spreading of cost across several customers, most internal software is rather admin-heavy (often "hack"), quality might be lower
- Environmental costs
  - → Equal (more or less fixed costs for all types of solutions)

Selecting Software: Custom

- With respect to TCO, custom software is therefore always a bad decision, usually even extremely bad
- But it has huge advantages:
  - → Does exactly what is needed and probably absolutely right
  - → Can be extended to new requirements rather easily
  - → Tailored to all specifics of the company
- All this is not included in the TCO!
  - → TCO is only a part; must be weighted against the advantages
    - » TCO is perfect for weighing solutions with the exact same specifications (or at least requirements)
- As the main process and its IT support should "define" the company, there custom software is encountered often
  - → Alternative: "One-of-many Inc." using standard solutions for its specific area, which are already tailored to such businesses

Selecting Software: Standard

- "Purchase price"
  - → Low-Medium: Created once and sold often, but might require more hardware, software, etc. (as it must fit and support all!)
- Training costs
  - → Low: Often employees already know this software or training materials are readily available
- Application costs
  - → Low-High: Depending how well this solution fits
- Maintenance and support costs
  - → Low: Relatively bug-free, admin-interface usually well-defined, teaching readily available, "superfluous" features may come in handy (and for free!) later on
- Environmental costs
  - → Equal (more or less fixed costs for all types of solutions)

Selecting Software: Standard

- With respect to TCO, standard software is therefore always a good decision
- Important advantages:
  - → Quality assessment possible in advance
  - → Fixed costs (development/modifications may "expand")
  - → One responsible company only
- But it may have some disadvantages:
  - → Friction: Interfaces to other software
  - → Does what it is designed for, not necessarily what you need
  - → Adaptations may have to be bought instead of developed
- Standard software therefore fits those business processes which are non-differentiating (lower performance not that important), but which are similar for many companies
  - → Examples: Accounting, employee management

Selecting Software: Customized

- Purchase price
  - → Medium-High: Standard software + additional development
    - » Depends on the amount of customization needed
- Training costs
  - → Medium: Special training for modifications needed
    - » Rest similar to standard software
- Application costs
  - → Low-Medium: Modifications mostly go in here
- Maintenance and support costs
  - → Medium-High: Debugging difficult, administration two-fold, teaching material partly incorrect/incomplete
- Environmental costs
  - → Equal (more or less fixed costs for all types of solutions)

Selecting Software: Customized

- With respect to TCO, customized software is therefore a mixed bag; possible useful but not necessarily
- Important advantages:
  - → Combines costs of standard SW with flexibility of custom SW
  - → Costs can be assessed (fixed + variable component)
    - » Sometimes; Counterexample: SAP introduction
- But it may have some disadvantages:
  - → Modifications may conflict with updates and future versions (re-customizing necessary)
  - → Differences in UI/L&F/handling, quality, ...
  - → Friction losses: Patches might not always work or not do exactly what is required, ...
- Fits business support processes, which are similar for many companies, but where local differences exist

Selecting Software: Conclusions

- Core business processes
  - → Look for custom software
    - » This is what defines the company and makes it better than competitors, therefore must be supported extremely well
  - → Alternatively look for "Branchenspezifische Software"
- Business support processes
  - → Customized software (patches for individual peculiarities)
    - » Not important enough for custom software, but standard software cannot fulfil the requirements
  - → When you want to use standard software, but absolutely require some modifications
  - → Or when custom software is desirable, but not affordable
- Administrative/other processes
  - → Use standard software
    - » Much cheaper and slightly lower performance less important
  - → Small modifications as necessary

Selecting providers

- Especially for small companies not everything can be provided for in-house
  - → Example: Webserver (needs better Internet connection, complex configuration to be secure, DMZ, ...)
- Not all possibilities will be discussed here, only:
  - → Internet connection (ISP)
  - → Web hosting & E-Mail
  - → Domain names
- Excluded are:
  - → Managed security (firewall, IDS, patches, ...)
  - → ASP (Application Service Providers)
  - → Administration & Help desk
  - → E-Commerce systems & payment gateways
  - → ...

Selecting providers: Outsourcing

- Outsourcing: Moving previously internal processes/services (non-core!) to specialized external service providers
  - → Not always directly applicable to small companies, but the principle remains valid!
- Principle: Do only what is specific to your company, i.e. the main business process and externalize support processes
- One essential requirement for successful outsourcing:
  - → SLA (=Service Level Agreement)
    - » As this is a process tightly interwoven with the company, any problems with it will have large repercussions
    - » An exact definition what must be provided when in which quality (and what happens if not) is therefore a "must"
  - → This is important also for other providers, although for SMEs this is reduced somewhat (not as strict)
    - » Guarantees usually cost money…

Selecting providers: SLA

- Contains specifications on:
  - → Service definition: What is to be provided?
  - → Performance tracking: How will the performance be measured?
    - » This is extremely important: Ping vs. applications response!
  - → Problem management: What happens in case of problems?
    - » Important: E.g. obligations to help in moving to another provider!
  - → Compensation: How the service fee is calculated
    - » Per seat, per minute, combinations, ...; exceeding limits → ?, ...
  - → Customer duties: Separation of responsibility
  - → Warranties and remedies: Liability for damages
  - → Security: What security and safety measures are required?
    - » Partitioning to other customers, general measures, ...
  - → Legal compliance, IPR, privacy
  - → Termination: When and how to end the service
    - » And what happens afterwards (e.g. content & usage data)

Selecting providers: SLA

- Problems of SLAs:
  - → SLAs are only sensible, when the provider can actually influence the service level
    - » This will be difficult if there are many providers with interdependent services!
      - – One provider for everything
  - → The exact level might be difficult to find
    - » E.g. a small reduction in allowed outage time might have large increases in costs
    - » Do you actually know what you really need (not just want!)?
  - → Evidence problems: "Something does not work"
    - » But who created the problem might be difficult to identify/prove
  - → Often very complicated, long and legal
    - » Makes it very hard to enforce or even determine whether a breach occurred
- SMEs: Use short, precise and easily measured SLAs

Selecting providers: ISP

- ISP here means "Internet connectivity" provider
- Selecting the ISP is crucial for all E-Business providers
  - → You must be reachable at any time
  - → E-Mails must be accepted every time
  - → Your webpages must be accessible even under heavy load
    - » See next part if not hosting them yourself!
  - → Servers and special services (e.g. VPN) must be possible
    - » Even better: Available through the provider
- Important here is the current situation (what services are provided in-house) as well as plans for future expansions
  - → Changing an ISP might not be that easy as it looks!
    - » E.g. dedicated lines, termination equipment, outage time, static IP addresses, custom routes, reconfiguring equipment, etc.

Selecting providers: ISP Checklist

- What services are needed?
  - → Pure connectivity or other services as well (DNS, hosting, VPN, content creation, mail server, virus scanning, spam prevention, intrusion detection, mobile access, ...)
- What bandwidth is required now and perhaps later?
  - → Is there an upgrade path available?
  - → Dialup or permanent connection?
- What about servers?
  - → (Dis)Allowed, static IP addresses, ...
- SLA?
  - → Guarantees for bandwidth (up to where), connectivity, ...
- Bandwidth? (A)Symmetric? Shared? Traffic?
  - → Including: How is the ISP connected to the Internet?
    - » Bottleneck, speed, redundancy?
- What happens in case of problems (DDoS)?

Selecting providers: ISP Checklist

- Customer support/hotline?
  - → When, how often, charges, response time, ...
- Any additional requirements?
  - → Existing telephone line, ISDN, VoIP, restricted service area, interface hardware/protocol ...
- Security services available/imposed?
  - → Closed ports, mail filtered; optional or mandatory?
- Reputation of ISP?
  - → Being on AOL might be a disadvantage ...
- Contractual restrictions?
  - → Termination times, payment methods, ...
- What equipment is provided?
  - → Where is the exact boundary and which interface is there?
- Pricing?

Selecting providers: Hosting

- Hosting for a business website is more complicated than with private/personal webpages
  - → Server reliability directly translates to money
  - → Security much more important (availability, defacing, ...)
  - → Payment gateways are too expensive/complicated for small companies; support needed
- Compared to an ISP, hosting is much more complicated
  - » Unless you have static webpages only...
  - → DNS is tied in tightly (nameservers and targets)
  - → Different server environment needed
    - » Database, programming language, server extensions
  - → Secure server (cryptography; e.g. SSL) requires hardware support or powerful servers
  - → Much more differentiation of service possible and available!

Selecting providers: Hosting Checklist

- How reliable is it?
  - → Average uptime (independent verification possible, backup systems, Internet connectivity, redundancy, ...
- Which security measures are in place?
  - → Physical security, fire suppression, network security, IDS, partitioning, firewall, applying patches, ...?
- Management of the server arranged how?
  - → Shared (=virtual server), Co-located (=own hardware in ISP data center, using UPS, Internet, etc.), unmanaged dedicated hosting (=server is leased; similar to collocated), managed dedicated hosting (=outsourcing; only content provided)
  - → Customer access when/how; content updates, ...
- Server characteristics?
  - → Amount of space/traffic, server extensions, database, software installation possible (free/from list), SSL, ...

Selecting providers: Hosting Checklist

- Usage statistics available?
  - → Details, analysis (e.g. geographical), periods, raw logs, ...
- Any free add-ons?
  - → E-Mail, FTP, automatic monitoring, content verifications, ...
- Surrounding issues?
  - → Porn, spam, game servers around (blacklisting!)?
- How good is customer support?
  - → Response time, technician/"sales", sysadmin available, ...
- Can specialties be provided?
  - → Multicasts, streaming video, webradio, ...
- What kind of monitoring is done?
  - → Technician on site 24/7, remote, per server/general, ...
- Pricing?

Selecting providers: DNS

- Previously domain names were available only by a single provider each; still the case for most CC domain names
  - → Especially ".com" is available by many providers
  - → Similarly, many providers "sublicense" names through acquiring "bulk registrations" by the (monopoly) registrar
- Separating DNS from hosting?
  - → Makes administration more difficult, but allows changing either of them much easier!
- Especially for SMEs this is a relatively easy decision: Go for the cheapest possible one!
  - → But take care of the nameserver: Provided by DNS provider or by your web hoster (or yourself)?

Selecting providers: DNS Checklist

- What about the nameservers?
  - → Provided (cheap registration often excludes them!), how many, quality, ...
- Who is named in the person records (owner, admin-c, ...)?
  - → Provider, hoster, you, ...
- How can changes be made?
  - → Additional subdomains, mail server entries, SPF/Sender-ID, domain transfer, domain name blocking, ...
- How are time-issues handeled?
  - → Automatic renewal, notification mail, etc.
- Any free add-ons available?
  - → DNS forwarding, E-Mails, name protection/monitoring, search engine registration, ...

Selecting providers: DNS Checklist

- What technical infrastructure is available?
  - → Own/foreign DNS servers, redundancy, Internet connection, ...
- WhoIs service content?
  - → What will be disclosed, where stored, availability, privacy protection schemes, ...
- What customer support?
  - → Reachability, topics, ...
- What about domain name disputes?
  - → UDRP, custom arbitration, enforcing court orders, ...
- Price and payment?
  - → Methods of payment, currency, location of provider, ...