

# Agile System Administration

# eXtreme System Administration

# Lean System Administration

# Lite System Administration

# Next Gen System Administration

*whatever.*

# Challenges for SysAdmins

# Challenges for SysAdmins

- Grid computing, cloud computing
- SOA (Service Oriented Architecture)
- Virtualisation



# Challenges for SysAdmins

- Grid computing, cloud computing
- SOA (Service Oriented Architecture)
- Virtualisation
- Vendor products
- Legacy systems
- Stack compatibility

# So how to improve?

# So how to improve?

- By way of comparison...

# The Leading Edge for Developers

- Kent Beck
- “Gang of Four”
- Robert C Martin
- Ivar Jacobson
- Martin Fowler
- Grady Booch
- ...

# The Leading Edge for Developers

- Object orientation
- Agile
- Refactoring
- Test-driven development
- Aspect-oriented development
- Use cases
- ...

# The Leading Edge for Developers

- eXtreme Programming (XP)
- Scrum
- DSDM
- Adaptive
- Crystal
- ...

# Do the Concepts Map?

# Do the Concepts Map?

- Test *everything*



# Do the Concepts Map?

- Test *everything*
- Don't ignore a broken test

# Do the Concepts Map?

- Test *everything*
- Don't ignore a broken test
- Working code over comprehensive documentation

# Do the Concepts Map?

- Test *everything*
- Don't ignore a broken test
- Working code over comprehensive documentation
- Pair programming

# Do the Concepts Map?

- Test *everything*
- Don't ignore a broken test
- Working code over comprehensive documentation
- Pair programming
- ...

# So what do SysAdmins have?

# So what do SysAdmins have?

- ITIL
- ASL (Application Services Library)
- COBIT, ISO9000, 6 Sigma, CMMI, ...

# So what do SysAdmins have?

- ITIL
- ASL (Application Services Library)
- COBIT, ISO9000, 6 Sigma, CMMI, ...
- Books?
- Technology-specific manuals
- Web, word of mouth, tradition

# The difference is the detail

- No equivalent to development best practices such as
  - xUnit
  - Object patterns (Gang of Four)
  - Code style (Size & scope of class/method etc)
  - Use of code version mgt repository



# So where's the leading edge?

# So where's the leading edge?

- Maybe development is easier to analyse
- ...or easier to create general rules
- ...or has more academic practitioners

# So where's the leading edge?

- Maybe development is easier to analyse
- ...or easier to create general rules
- ...or has more academic practitioners
- But is there nothing to learn?

- What are the Best Practices?
- What are the “smells”?
- What are the metrics?

# Some ideas for Best Practices

- Total Configuration Management
- Microcharging
- End-to-end Monitoring
- Automated App Deployment
- (Un?)Structured Documentation

# Total Configuration Management

- **everything** in the CMDB
  - ini files, xml configs, app startup env & params
  - yes, even hostnames, users...
  - track dependencies
- if it's not in the CMDB, it doesn't happen
- CMDB generates the envt
  - no more autodiscovery scanning
  - no more non-compliance reports

# Microcharging

- Charge costs back to the actual users
- You expect an itemised phone bill
- Automated measures of all costs
  - disk space, CPU, RAM
  - support calls (with caution)
  - license & s/w support
- True cost of legacy systems
- The customers can decide

# End-to-end Monitoring

- The runtime equivalent of TDD
- Monitor each “moving part” with realistic transactions
- Factor in dependency tree
- Ever more important with n-tier, SOA
- Monitoring processes etc is merely a (useful) supplement



# Automated App Deployment

- There's no place for a CD outside the test lab
- No wizards, prompts, or manual entry please!
- See under CMDB!

# Structured Documentation

- Ideally, every doc has its place
  - particularly business-as-usual procedures
  - show up the missing items
- Target docs to the reader's role
- Must be exceedingly easy to add a doc
- Rating system to reduce noise
  - entropy *\*will\** happen
- Excellent search is essential

# Some metrics?

- Uptime
- Tickets raised
- User satisfaction

# Some metrics?

- Uptime
- Tickets raised
- User satisfaction
  
- Better ideas?
- Or just not useful?

# How about some tools?

# Tools as they are

- Mostly proprietary
  - presumably sysadmins don't develop as much as developers
- Some open source (Nagios, RT)
- Why are so many, so bad?
  - poor tools, or poorly configured?

# Tools as they should be

- A CMDB that allows you to make safe changes in production
- A framework for regular (nightly/weekly) jobs
- Decent ticketing / work-request
- Decent documentation
- Version management for the dependency stack
- Resource tracking – charging, trend analysis, capacity planning

# Over to you

- Thoughts?
- A “call to arms”, or am I just tilting at windmills?
- More at [www.agilesysadmin.org](http://www.agilesysadmin.org)



# Agile Manifesto

- **Individuals and interactions**  
over processes and tools
- **Working code**  
over comprehensive documentation
- **Customer collaboration**  
over contract negotiation
- **Responding to change**  
over following a plan

# Compatibility stack – one product

- **Client:** OS + Browser
- **Mid-tier:** OS + Apache + Java + Servlet container + vendor code + custom code
- **Main server:** OS + vendor code (+ embedded Tomcat + Java) + customisations
- **DB:** OS + RDBMS
- Vendor plugins (may be on different OS)
- let's ignore network, load balancers, SAN/NAS...