Agile Infrastructure & Operations
A bit of Context

• Not about Development infrastructure
• Server and Network Oriented projects
• Within Large Enterprise context
• IT people, Operations separated from Dev. by design
Who I am

- Patrick Debois
- Independent Consultant
- I mainly do Servers/Network/Security
- Guide development projects to operational status and beyond
- Currently developer ;-}
Any Developers here?
(coders, testers)
Any IT People?

(infrastructure, servers, network)
Any Operations?
(helpdesk, end-user support)
How Agile are your Developers?

- Waterfall
- In between
- Agile
How Agile is your IT department?

- Waterfall
- In between
- Agile
How Agile is your Operations team?

- Waterfall
- In between
- Agile
## How Agile is your enterprise?

<table>
<thead>
<tr>
<th>The Emperor</th>
<th>Darth Vader</th>
<th>Storm Trooper</th>
<th>Han Solo</th>
<th>Luke Skywalker</th>
<th>Yoda</th>
</tr>
</thead>
<tbody>
<tr>
<td>100% Waterfall</td>
<td>Talks Agile, walks waterfall</td>
<td>Doesn't care, just executes orders</td>
<td>Likes Agile, but doesn't practice it</td>
<td>Learning the Agile Powers</td>
<td>A true Agile Master</td>
</tr>
</tbody>
</table>

---

**Internet Survey**

60 People Participated

Posted on agile mailing-lists

By no means scientific!

---

Copyright: [http://www.lego.com](http://www.lego.com)
Survey Results

Three cases
Case 1: Infrastructure only, no development
Moving a data-center

- 50 applications for public use (government)
- no new development, maintenance only
- design was taking long time, no actual result
- “I don’t care if it is not finished, I need something now, you can improve later” (political deadline)
- Made us switch to Agile (Scrum)
Product Owner

- The people specifying the requirements where not there anymore
- Applications as a product owner (infrastructure requirements + SLA)
- Operational Team (monitoring, remote access, ...)
Product Backlog

- Ordering by value saved vs. added value
- Functional requirements of the application did not matter.
- Non functional requirements of the application = infrastructure functional requirements (security, performance, ...)

Agile2008 Conference
Sprint Backlog

- First sprint: prepare minimal working
- Second sprint: deploy first application
- Third sprint: mix of improvement + new application
User Stories

• As an administrator I want to connect to System X so that I can reboot the system

• As an application I need a database so I can store my data

• As a service manager I need a report of the CPU, Memory and Disk so I can report it on the weekly service meetings
Iterative/Refactoring

- host files -> DNS; server routing -> real Router; local disks -> SAN Storage; apache Proxy -> SSL accelerator; VLAN’s -> multiple physical network

- Doing the same story multiple times with improvements, at least we had something

- Was first seen as temp solutions as usual, but now there would be a followup.
Test Driven Infrastructure

- No OS or SAN unit tests exist
- Tests executed at the application level
- Implicit test of components
- Monitoring probes, Load testing as test scenario's
The Bad

Case 2: Infrastructure, Development and Operations
Disaster Recovery

- Infrastructure was failing
- Applications were crashing
- But they needed disaster recovery?
- Infra team put a lock on the door!
- Infra team did not care about the applications
Technical Debt

- No updates/patches because of unknown impact
- Machines maintenance expiring
- Restart scripts for fast fixing
- Migrations half finished
Group vs. Team

• Technical specialists (Desktop, Security, Network)
• Nobody did other tasks (not capable)
• Backlog was a list of TODO’s
• No application knowledge internal
• Injected middleware people to bridge
Multiple Product Owners

- What’s the ‘correct’ order?
- End-Users (no value)
- Project Managers (value added)
- Operational Manager (non functional)
- Got no ‘real’ decision ..
Daily Scrum

- Depending on the priorities people were interested or not (fatigue)
- Still it was a form of information radiation
- People would pair for tasks (spread knowledge)
Case 3: Agile Infrastructure and Development
Application Server Upgrade

- Developers need new application Server functionality
- They ‘check’ it (wizard style) -> It works
- But what about non functional
  - monitoring, redundancy, backup agent, JVM, OS libraries ...
Cross Functional Team

- Break the Agile Development by including infrastructure people in the team
- Infrastructural changes get radiated better
- Infra requirements sooner visible
- Problem with who owns the resource (project mgr, operations mgr?)
Deploy Often

- Nightly builds (config files)
- not only application
- also OS, Virtual machine, DB, AppServer
- Reconfiguration becomes reinstallation
- Patches tested every iteration
- Operations to use it after the project is finished= using unit tests to test OS patch
- simpler to setup; faster setup time; backup less
Conclusions?
Caveat!

Reality is complex, changing and is not always amenable to narrowly focused technical models.

Platonicity
Conclusions

- Three Levels need to be tackled
  - technical level (tools, skills, iterative working)
  - project (communication to teams, reach out to other enterprise)
  - operations (mix of non-planned things, operational mgr vs. project mgr priorities)
Agilista, This is your last chance. After this, there is no turning back. You take the blue pill - the story ends, you wake up in your bed and believe whatever you want to believe. You take the red pill - you stay in Wonderland and I show you how deep the rabbit-hole goes.
Questions?
Thank you!
Developers don’t care about the Server or the Network

No way

Maybe

Absolutely
Developers need more IT skills

No way

Maybe

Absolutely