Windows 7

The "Product/Project Reflection" edition.

James Schaefer – DePaul Information Services – May 2012

What we'll cover today

50,000ft overview of the platform

Some of the more appealing "features" and apps

But… because this is a "Communications" function:

Project overview

Successes and failures

Q&A, Comments and Criticisms

Disclaimer

Consider the terms: I, me, we, they throughout this presentation.

Recall the concepts from "Good to Great". (And not just the ones that are easy to implement.)

The opinions here are strictly my own and do not necessarily represent those of Information Services, DePaul University, or Major League Baseball.

This presentation is being provided to you intentionally unrehearsed.

Meet Windows 7

64Bit OS (>4GB RAM possible).

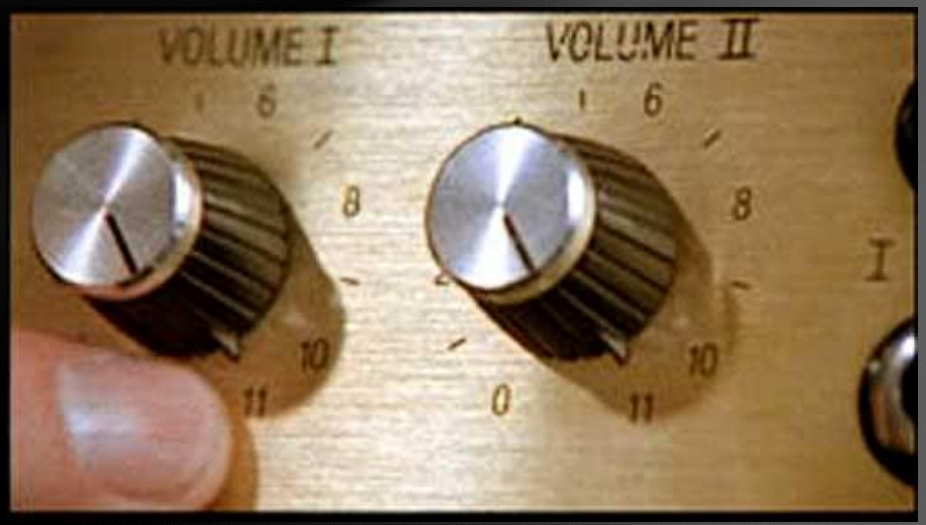
Released (RTM) July 2009.

Built on the Server 2008 kernel/platform.

Now handles up to 256 virtual cores.

Eliminates static-cling and ring-around-the-collar.

“What Vista was supposed to be”.



Under the hood…

Better user interface

"Aero" features, shortcuts, and ~~Sherlock~~ Search

Better hardware support

MS and third party partnerships, driver memory handling

Better power management

Sleep states, active patrolling, process coalescence

Tighter security

DNSSEC, better UAC integration, better firewall control, logging

Nifty features people might actually care about:

New Calculator

~~Widgets~~ Gadgets

The Snipping Tool

Sticky Notes (welcome back to Mac OS7)

Problem Steps Recorder

(All the crazy crap that comes with Office 2010)

Other fun stuff I like:

"Aero" Features

- "Areo Snap"
- "Aero Shake" (Shakey + Shakey)
- "Aero Flip" (Windows + Tab)
- Revolving (Aero theme) Backgrounds (chose your own adventure)

Improved Projector Connectivity (Windows + P)

Open a Command Prompt Here ("Shift + Folder")

POWERSHELL*** (Do a AMA on this one please!)

The Search Box

Task Scheduler

Customize the Notifications area (Taskbar->customize) and QuickLaunch arrangements

Yet more cool, hidden gems

Energy Analyzer

powercfg -energy -output report.html

"God" Mode

OpenSezMe.{ED7BA470-8E54-465E-825C-99712043E01C}

Multi-Threaded Robocopy (and other multi-threaded goodness)

(add the /mt switch at the end of your command line)

Resource Monitor (in the new Task Manager)

</salespitch>

The Windows 7 Project

Official kickoff on November 19, 2009

(Planned deployment of November 1, 2010)

"Man on the Moon Concept"

Deploy new workstations and provide a user driven "self service" migration system

Windows 7(64) with Office (64)

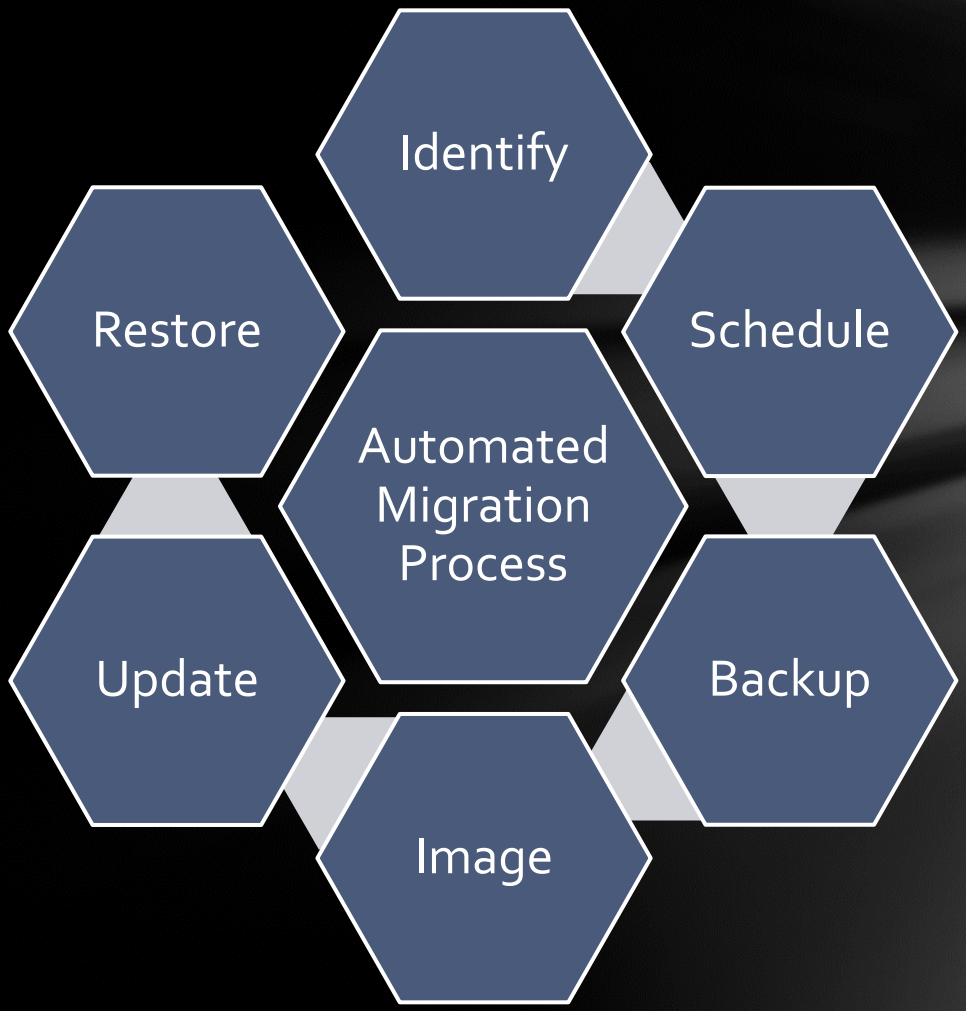
Clean up some of the "sins of the past"

Two project managers, all three divisions of IS involved

Desktop deployment to set the stage for future technologies

Directory, Exchange 2010, DirectAccess, VM Computing, R. profiles

Awesome "SmartArt Graphic":



Project Challenges

Right out of the gate:

No hardware inventory of our “fleet”.

No solid intel on what applications will/will not work (or even what we use).

We have never been in the business of backing up user data.

We had only performed a “come and get it” upgrade once before (Office 07)

What we figured out over time:

Our software deployment system was not functioning to our expectations.

We had so many variants deployed into the current environment, we didn’t know what we didn’t know.

Lots of people have great ideas (so long as they don’t have to implement/support them)

What happened:

We built an image, got it working, and got it deployed, but…

We ended up changing the Office version at the last minute.

We ended up throwing together a new scheduling system "on the fly".

We ended up splitting the technical team where one person built a completely new deployment system separate from the rest of the group.

We ended up splitting the group into a "image" team, a "deployment system" team, a "training" team, a "communications" team, and a "project management" team. (Note the lack of a "Windows 7 Project Team").

What else?

We started holding "non-productive" weekly status meetings while email updates could/should have been arranged instead.

We tried to steer the project by committee (this rarely works).

We allowed "the group" to weigh in on decisions that should have never been decided by that group.

Sounds like a train wreck… did ANYTHING work out right?

Project Successes

Roll out application and process was an overall success and very unique.

Gave early adopters one option and hold-outs another*.

We were able to provide a stable, solid OS to our users.

We found a “work around” to most of our major issues.

Avoided the "Rolling Thunder" upgrade experience for users.

We now have a platform to collect better intel for the next rollout.

The University is in a better position due to this project.

Since January, we have deployed about 1200 new systems, migrated 810 systems and with that, backed up over 12.6 TB of User data.

(So far as I know, we have only lost two workstations)

Project Failures

<u>Overt Failures:</u>

We didn't meet our deadline of November 2010 (January 2011).

Failed to successfully leverage our existing software deployment platform for the project.

Failed accurately estimate the amount of work that may be required to complete the project. (other things suffered)

Lesser Noticed Project Failures

Not so overt failures: (my top six anyway)

I failed to understand how different people approach tasks, and did not adjust for those differences (did not consider the personalities involved).

I did not allow the project managers to do their job.

I did not push back on arbitrary deadlines and last minute add-ons (arrogance versus reality).

I failed to take direct control of specific project assets when I saw interference interdicting from outside forces.

I did not do everything I could to maintain the "MOTM" concept and project vision. (Failure of Leadership)

We ended up with a deployment system that works, but it doesn't work as well as it could. (And support is still an issue)

So what can we do next time?

Pay closer attention to what we're delivering.

(what do people want?/know our customer)

Spend more time and effort in R&D and prototyping before project planning.

Get buy-in at 'the top' but elsewhere as well.

Adjust project schedules (deadlines) as requirements change.

Create a better chain of communications between and among project members and teams.

Use technology where you can, and avoid it where it isn’t appropriate.

Implement the concept of the “change freeze” and mean it.

Personal Lessons Learned

It's never as easy, it always takes longer, and it always costs more.

Measure twice, cut once. (Office version, software capabilities)

Nothing is as valuable as hard data. (An ounce of intel…)

<u>Additionally:</u>

If you're supposed to be playing air traffic control, don't try to fly the plane.

If you want to/offer to help, really help (don't just do).

"Trust, but verify".

If you're going to assume (which you shouldn't), assume the worst.

Decision points need defaults.

"Leadership is about people."

Questions, Comments, Concerns, or Clarifications?

Cool. Thank you!