

Distributed agile Development

Experiments at
Microsoft patterns & practices



Ade Miller
Senior Development Manager
Microsoft patterns & practices group

Who am I?

This talk is very much about our experiences at patterns & practices.

Who are patterns & practices?

- Deliver guidance to application developers
 - Mine expertise in the community
 - Guidance: books, code and frameworks
- Customer focused development
 - Deliver solutions driven by customer demand
 - Involve customers in projects
 - Make customers happier and more successful

Dev Center: <http://msdn.microsoft.com/practices>



We produce things like...

- Enterprise Library (and Unity)
- WPF and Silverlight Guidance (Prism)
- The Application Architecture Guide

CAVEAT...

patterns & practices

≠

Microsoft

Microsoft isn't one homogeneous entity.

My group is a very small part of it and doesn't represent what many other teams do.

Why Collocate?

- Communication
- Efficiency
- Risk

Increase team efficiency

Software development is about communication

Drive down risk (reduce failure rates by 20%)

--

About the picture: Hubble mission control

Why Distribute at All?

- Global Markets
- Global talent
- Reducing costs

57% of agile teams are distributed (VersionOne survey)

- Global Markets
- Global Talent
- Reduce Costs?

There's a tradeoff you get these things at the potential expense of performance and productivity

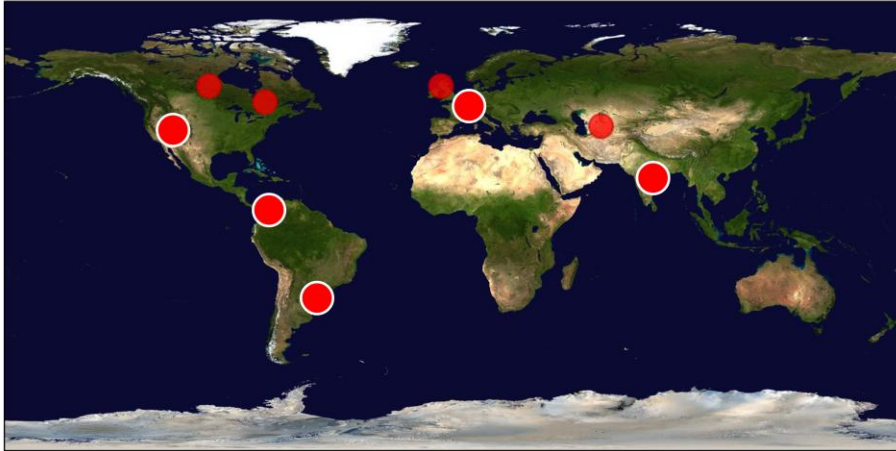
Distributed Teams at p&p

- 4-5 Small teams
 - Total of 25 employees and 30+ contractors
 - Teams often include subject matter experts
 - Strong agile culture: Scrum + XP
 - ½ located in a team room in Redmond
 - ½ distributed in 1-3 locations Worldwide
- Ship to customers every two weeks
- Projects last 3-8 months

p&p tends to distribute for the latter two reasons;

Talent and (perceived) costs

p&p Typical Team Distribution



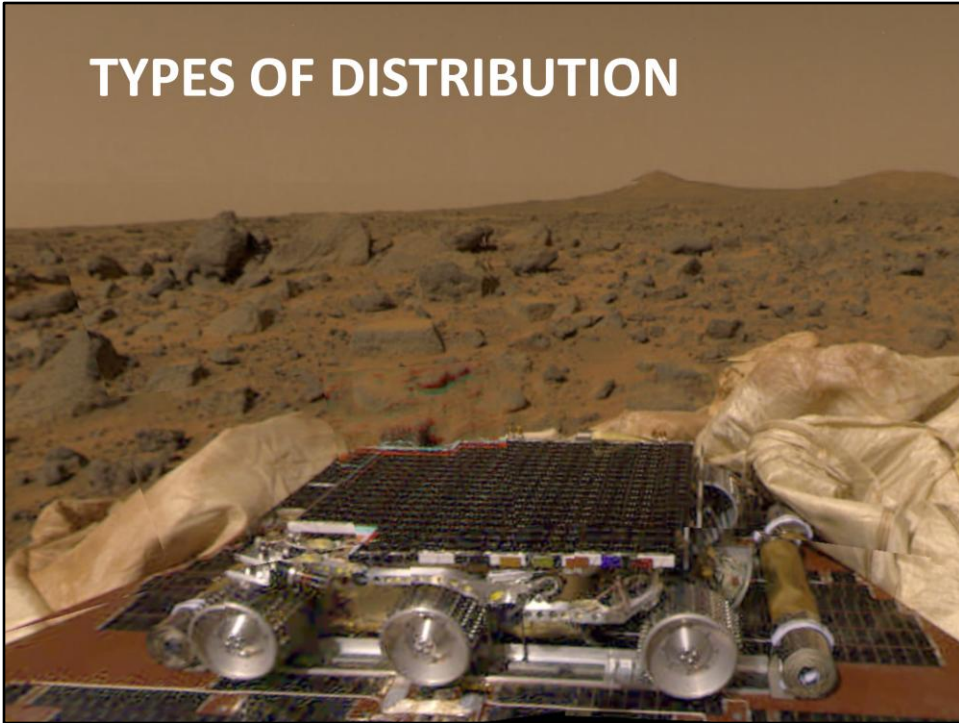
FTEs in Redmond and UK

Contractors in Argentina, Costa Rica, Holland, India and Ukraine...

And Montreal

SMEs from all over the place

TYPES OF DISTRIBUTION

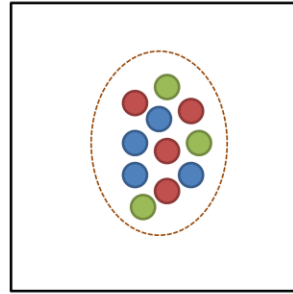


A mental model for thinking about distributed teams.
Roughly in order of difficulty...

Work in progress (new thinking not in the paper)

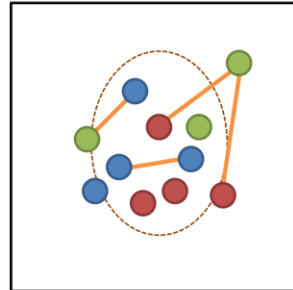
Not distributed

- Everyone in the same room
- High bandwidth
- Symmetric communication



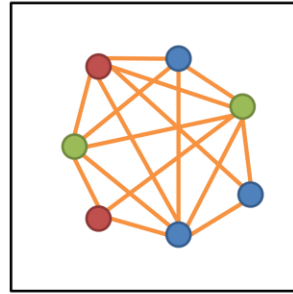
Distributed but in denial

- Team thinks it's together
- But acts distributed



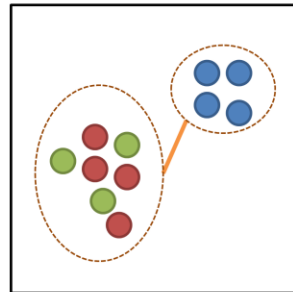
Completely distributed

- Symmetric communication barriers



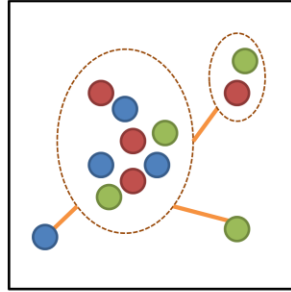
Distributed by function

- Asymmetric communication barriers
- Builds walls between disciplines



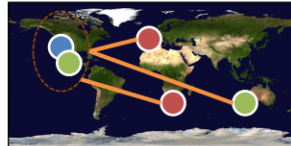
Ad-hoc distribution

- Asymmetric communication barriers



Distributed across time zones

- Adds temporal asymmetry
- Lowers bandwidth



CHALLENGES OF DISTRIBUTION

Vision...



Reality...



What I want is on the left. What it feels like is on the right.

A story...

During the Agile 2008 distributed agile workshop I spent several hours listening to people talking about their distributed teams...

Distributed teams sound like dysfunctional teams!
Time zone differences make this worse!

Communication Challenges

- Lost meaning
- Lost trust
- Core hours and time zones

Teams rely on

- Person to person communication
- A set of mutually supporting practices

They use this to build trust!

Some Problematic Practices

- Story cards
- Pair programming
- Daily standups
- Design white boarding
- Coaching

Story cards

Pair programming

Everything is harder...

- Coaching

- Learning about the problem domain, architecture or design

- Story cards and pair programming are both examples... These need to be modified in the light of distribution

If you focus on the (stock) practices then this makes things worse!

Distribution Changes Your Product

“Organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations.”

- Melvin Conway

Distribution may start to effect the way your product looks!

- Conway's Law



Agile is a bunch of overlapping practices that rely on communication in one way or another.

Remove one and it weakens the other.

Example... Pair programming. Hard with two developers in different (non-overlapping) time zones.

Do pair programming when you can, supplement it with code reviews (1:1 & team).

Example... Story cards are really placeholders for conversations.

The goal... minimize the difference between people in the same room and people elsewhere.

Tools...

- Conference phones
- Web cams, if bandwidth allows
- Hands free headsets
- Projectors

--

About the picture: The VLA radio telescope



At pivotal points during the project...

- The beginning... requirements, release planning, design, metaphor
- Key points... releases, customer visits
- The end... retrospective, ship party

On regular rotations...

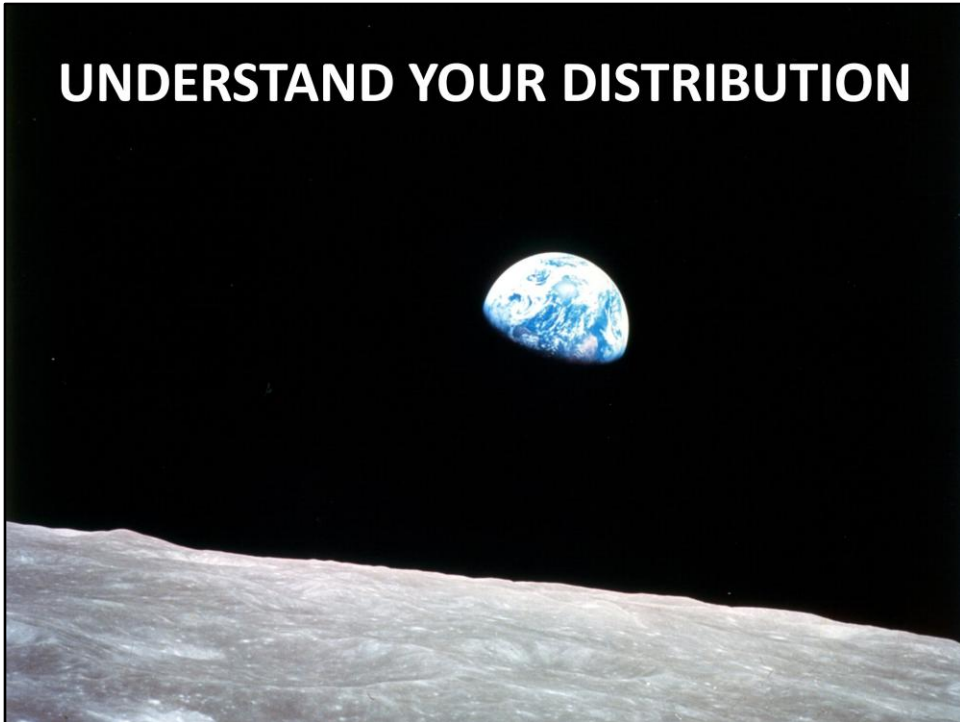
- Rotate team members throughout the project
- p&p does this 2 weeks out of 6

Really all about building *trust* and inter-personal relationships

So make sure it includes some social aspects!

- Time for hanging out over coffee
- Team lunch
- Rock band party

UNDERSTAND YOUR DISTRIBUTION



Time zones just make things harder

- Make the best use of overlapping time
- Use a team room representative (more on that in a moment)

Asymmetric distributions are hard too

- Best avoided
- Team buddy
- Enforce symmetry

--

About the picture: This view of the rising Earth greeted the Apollo 8 astronauts as they came from behind the Moon.

Team Room Representative



It's often helpful to have someone in the central location represent those who are remote

At p&p for example:

Remote test team of 2-4 in India

1-2 testers in the room

Part of their role is to be the voice of the remote team



It's easy to abandon key practices if there's nobody there to keep you on the straight and narrow.

Coaches are *not* practice cops

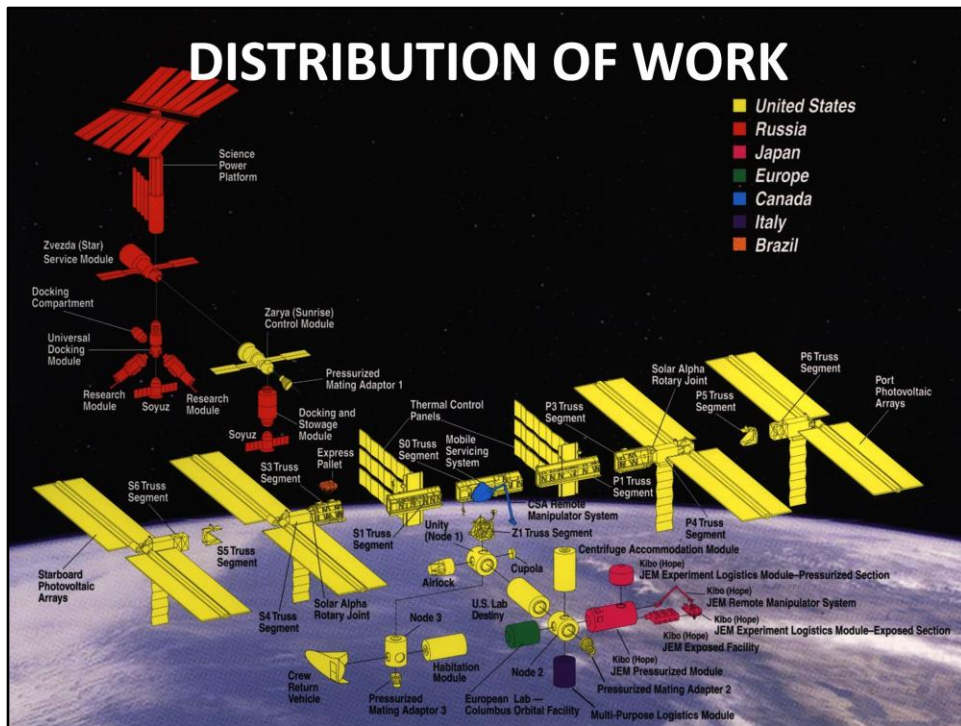
- Remind team of underlying principles
- Help them adapt and modify

Have someone who knows they're the coach!

Provide the team with training so everyone knows what to expect.

- @ p&p this is a wiki and books.

Hold retrospectives!



This is NOT how you should think about building a software product!

Users care about end to end stories not individual components

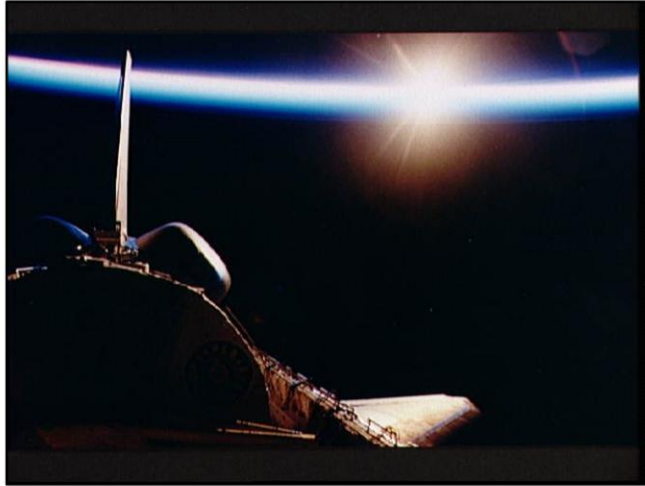
Conway's Law will tempt you to...

- Assign work by location
- Specialize sub-teams accordingly

Actually you should do the opposite

- Assign stories to the team
- Work to remove significant specialization

Follow the Sun



This isn't something p&p does but

About the picture: Earth limb at sunrise

BUILD THE TEAM OVER TIME



NASA didn't throw away the people and learning from the early space programs and start over with Apollo

- Teams take time to form
- Learn practices and working styles
- Learn the problem domain

Don't do this with your teams

- Build them round a core team
- Product additional artifacts (like a wiki) to ramp up new people

--

About the picture: This is Gemini 7 taken from Gemini 6. NASA didn't just get up one day and say "let's put a man on the moon"... That took a politician!

PROVIDE THE RIGHT TOOLS



The reason agile teams use things like sticky notes is because they're easy to adapt and don't constrain you.

When you provide tools that support distribution make sure they don't constrain you. Don't replace something simple with something overly complex!

Example... The WSSF:ME removed almost all the workflow from our VSTS template.

p&p uses...

SharePoint wikis

VSTS

Scrum for Team System by Conchango

LiveMeeting

SharedView

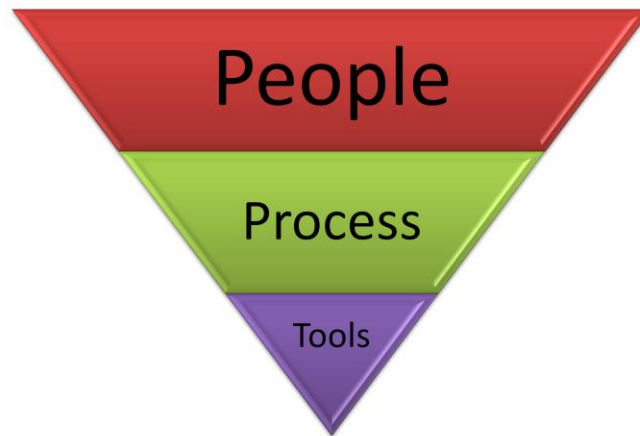
Conference phones

Instant Messenger

Hands free headsets

Projectors

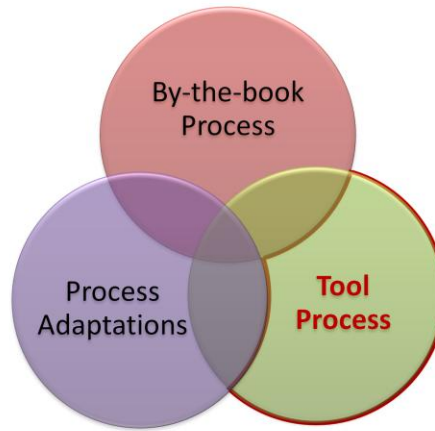
Tools... A Third Order Effect



The bad...

Tools are not the most important thing...

Tooling Driven Process



The bad...

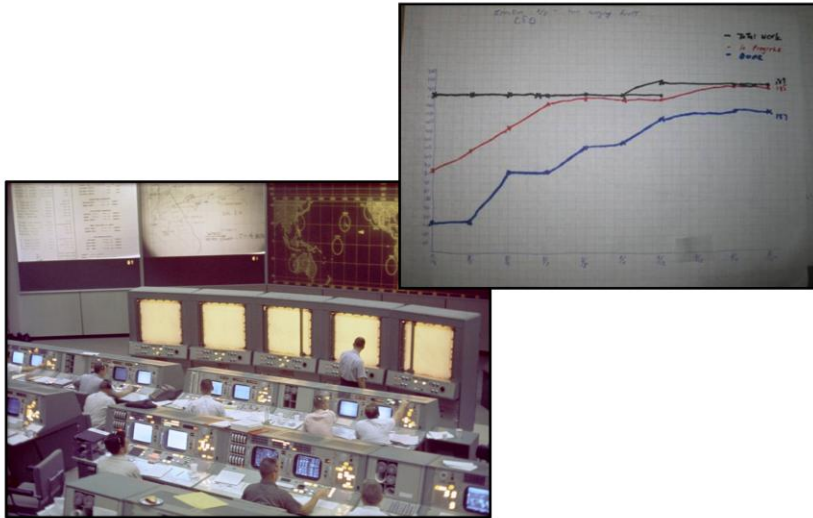
But... To be effective your tools must be aligned with your process.

Otherwise this will create an impedance mismatch – people will fight the tool.

Beware of tools which take over your process.

Especially a consideration for distributed teams because you can't skip the tools.

Big Visible Charts



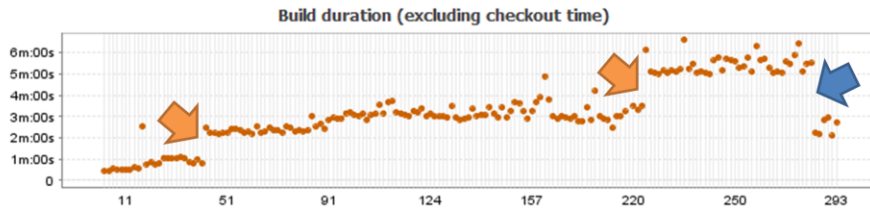
About the pictures: Gemini Mission control and an Enterprise Library 4.1 burn up chart.

Big Visible Charts

“A simple chart on the wall can bring important information to the attention of the team, the customer, and everyone else who passes through the area.”

- Ron Jeffries

Big Visible Charts and CI



- Charts are transient
 - Remove them when the problem is solved
 - Move on to the next problem area
- Track trends over time

What happened in this chart.

I added more and more tests (good)

The build took longer and longer (bad)

Eventually the chart embarrassed me so much I made some of my tests a lot faster.

Note: All my unit tests run in \ll 1s but I had a couple of acceptance tests that took a lot longer especially when running to gather code coverage data. This was the problem.

Do's and Don'ts

- Do** Maximize communication bandwidth
- Don't** Continually reorganize your teams
- Do** Plan to travel
- Don't** Distribute your work by components
- Do** Provide tools to augment existing practices
- Don't** Let remote team members get forgotten
- Do** Evolve the team's practices
- Don't** Forget retrospectives
- Do** Focus on coaching

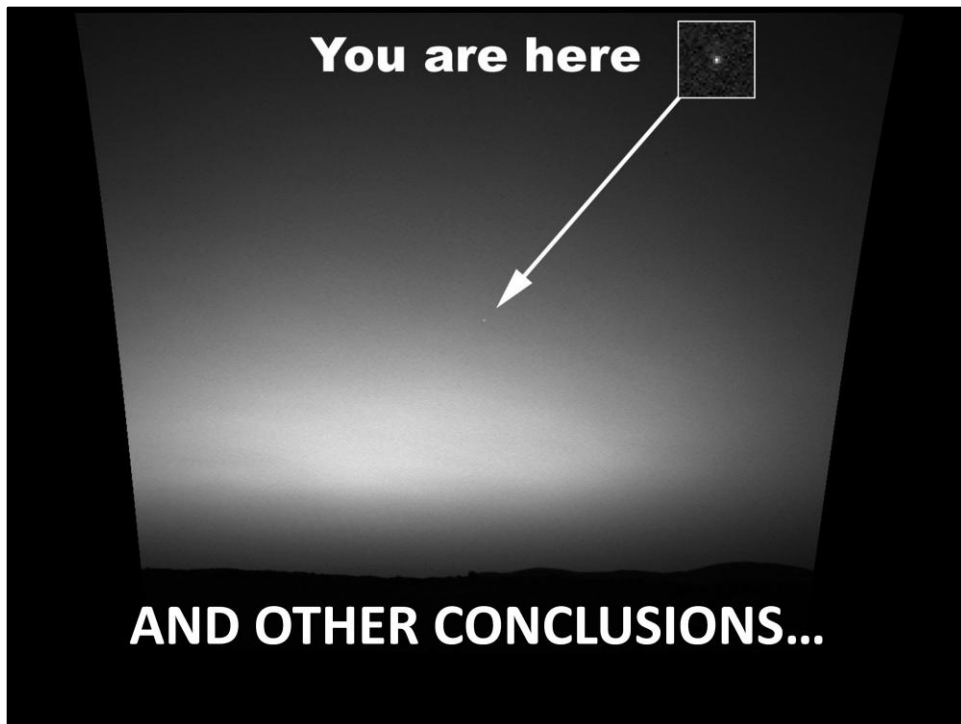
Here's a list of do's and don't from the paper. It's twenty pages so this is really just a flavor for some of what's inside.

Do

Adapt agile to your specific situation

Don't

Simply follow a stock set of practices



It's all about allowing the principles to guide your experiences rather than blindly following a set of practices that were designed for a team working in the same room

So what are the conclusions!

- You can be successful as we have been
- It requires (significant) effort
- Travelling
- Providing the right tools
- Coaching teams
- Thinking about what you're doing and why – principles not practices

--

About the picture: First picture of Earth taken from the surface of another planet, Mars.

More distribution not Less



This problem isn't going to go away so we had better start to figure out how to do it better!

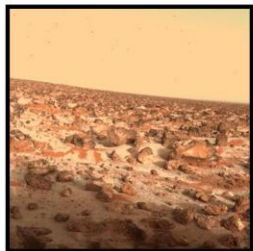
If you look at Microsoft over the past five or six years they've become much more distributed

There are other approaches like follow the sun which we've not experimented with.

Tools are important and new ones appear all the time.

--

About the picture: Humankind's best attempt at distribution. In 2005 Voyager 1 left the solar system after a 28 years of travel.



QUESTIONS?

I'm done!

I am *not able* to answer the following:

Is there life on Mars?

Is the Standard Model of particle physics correct?

How old is the Universe?

Come and talk to me, I'd love to hear about your experiences!

--

You can find out more about the questions, but not the answers, here:

<http://mars.jpl.nasa.gov/science/life/index.html>

<http://www.bnl.gov/bnlweb/pubaf/pr/2001/bnlpr020801.htm>

<http://starchild.gsfc.nasa.gov/docs/StarChild/questions/question28.html>



Resources

Full slide deck available here (soon):

<http://www.ademiller.com/tech/talks>

Download the white paper & video here:

<http://msdn.microsoft.com/practices>

Microsoft patterns & practices

<http://msdn.microsoft.com/practices>

Ade Miller's blog

<http://www.ademiller.com/tech>

Stock photos in this presentation

<http://www.nasaimages.org>

The slide deck for this presentation along with speaker notes can be downloaded
Ade blogs a lot about many of the things discussed here!

Slides and Other Cool Stuff

<http://ademiller.com/tech>

- This and other decks
- Experience reports

<http://microsoft.com/agile>

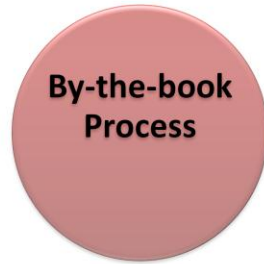
- White papers
- Videos of the p&p team rooms

ade.miller@microsoft.com

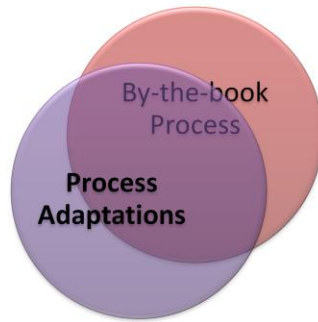


APPENDIX

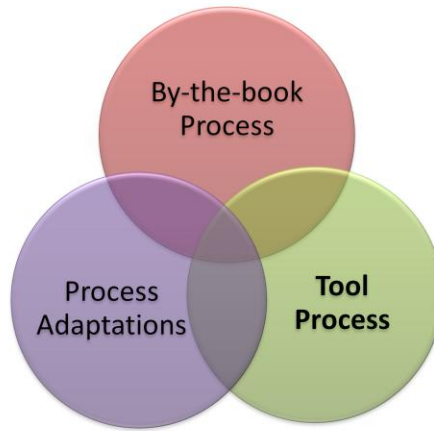
Process by the Book



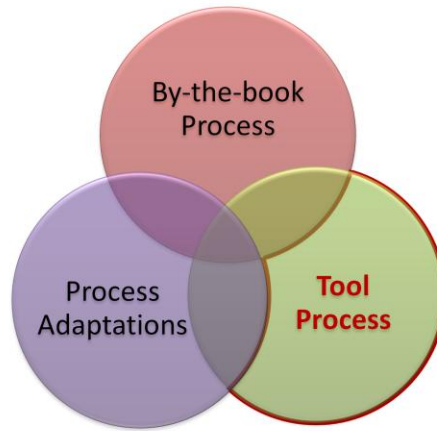
Process Adaptation



Process and Tooling



Tooling Driven Process



Microsoft Redmond, Washington
October 12-16, 2009

patterns & practices Summit

sponsored by Microsoft
www.pnpsummit.com

The International patterns & practices Summits showcase the official Microsoft "patterns & practices" for developers, designers, and solutions architects who need to learn how to integrate architectural design, patterns and procedures with the technology provided by Microsoft's .NET platform.

The patterns & practices Summits packs into five full days unique presentations offered by many of the industry's best speakers. An evening reception sponsored by Infragistics will give attendees an opportunity to meet with the patterns & practices team—a second evening features a reception with Microsoft Research. Attendees will come away with a strong foundation in the architectural principles underlying Microsoft's .NET technology, which will prepare them to construct the next generation of enterprise-scale applications.



Keynote presentations by:



What Previous Attendees Have Said...

- "Overall, conference was **great**, gave us good roadmap to follow."
- "Thank you for all the **great ideas**. I am looking forward to going home and applying them."
- "Many of the things I learned here will revolutionize our practices, processes, designs, and patterns. **THANKS!**"
- "Speakers were **great**"
- "**Excellent conference**, just first day alone was worth the money. Thank you."
- "This was an **excellent value** and I would recommend it to **everyone!!**"
- "All speakers were **very knowledgeable** and they addressed many issues that I face every day."
- "This was an excellent conference for me. I'm taking away some **useful tools**."

More information:

<http://tinyurl.com/pnpsummit>

- **Facebook:** Stay tuned to updates on p&p summit by becoming a fan of p&p summit in Facebook. Click [here](#) to become a fan.
- **Digg It:** Show your support, by digging the p&p summit announcement [here](#).
- **Twitter:** In your tweets about p&p summit, please add the tag "#pnpsummit".

