



**Agile Project Management:
Key Differences with
Case Study Examples**

**Paul E. McMahon, Principal
PEM Systems**

Topics



- ◆ Similarities & differences of traditional & agile project management approaches
 - ✓ Help you decide if agile—or hybrid agile—may benefit you
 - ✓ Case studies included

***Reference: November, 2006 Crosstalk article
“Are Management Basics Affected When Using Agile Methods?”***

Project Management Basics

- ◆ Fundamental to Project Management
 - ✓ Planning
 - ✓ Monitoring
 - ✓ Controlling

- ◆ Monitoring & Controlling achieved by executing the plan & taking action

- ◆ Our focus is on planning
 - ✓ Five basics steps
 - What, Who, When, How & How Much

Step 1: “What” – Scope the Effort



◆ Traditional Approach

- ✓ Work Breakdown Structure (WBS)
- ✓ Manageable “chunks”

◆ Agile Approach

- ✓ Basic concept doesn't change

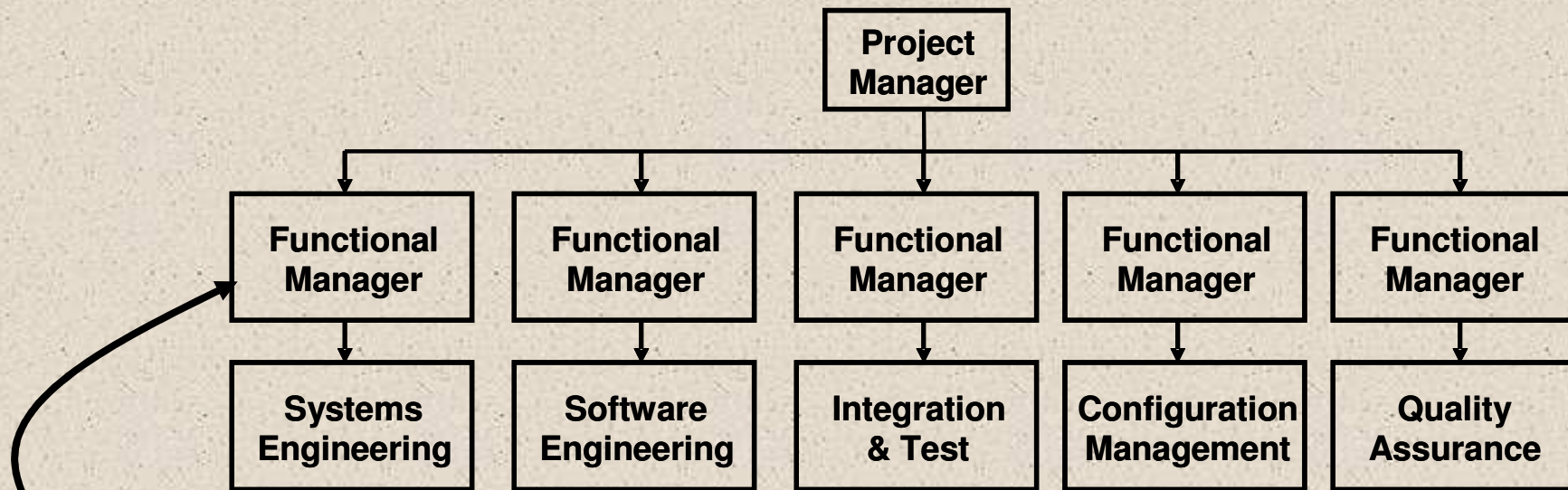
✓ May:

- Provide less detail early 
- Not fully scope all work up front 

Note: Addressed under Step 3: “When”

Caution: Sounds easy. But 2 critical areas that can lead to trouble if don't understand conditions where each appropriate

Step 2: “Who” – Organization & People

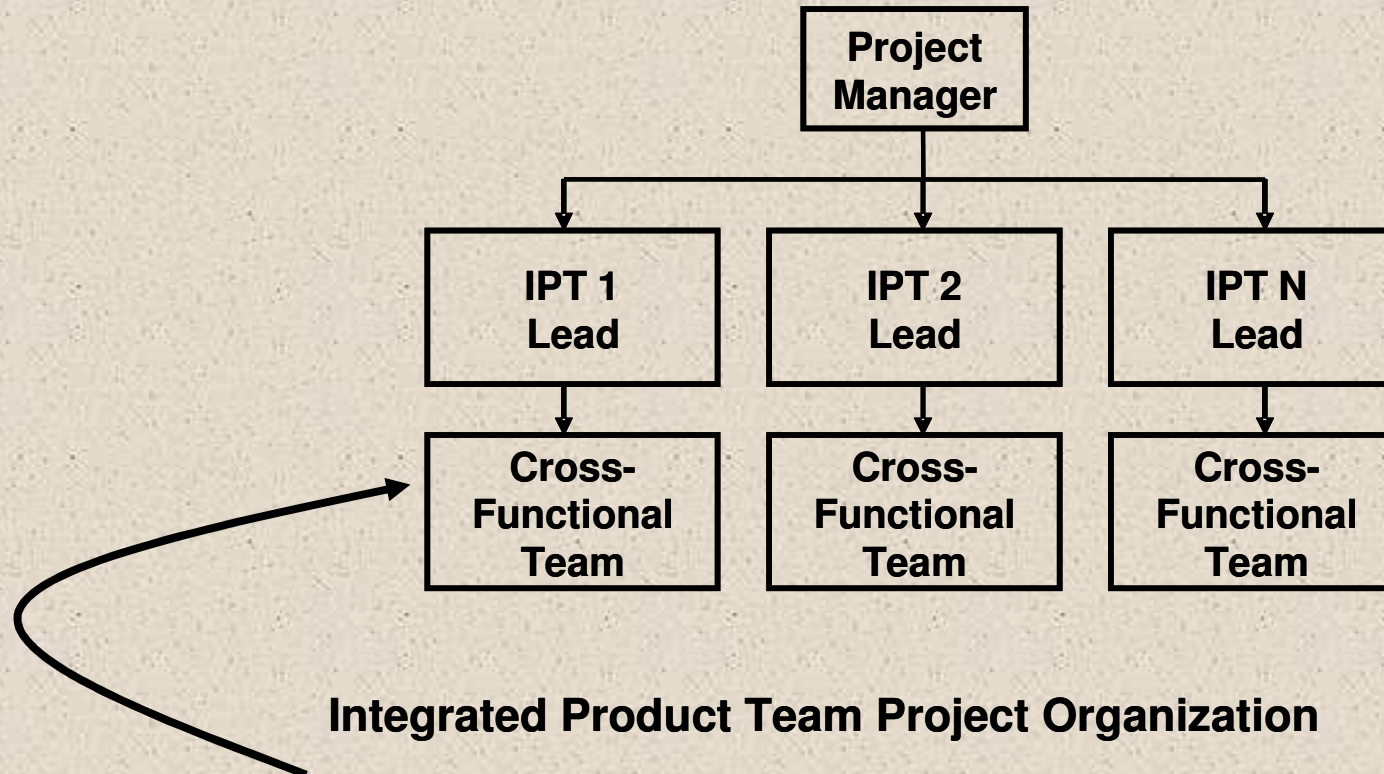


Traditional Project Organization

Functional Managers:

- *Allocate personnel resources*
- *Provide level of tasking*
- *Receive task status*

Step 2: “Who” – Organization & People (cont)



Traditional Integrated Product Teams (IPT): Cross-functional

- *Include all functions on each team for improved teamwork*
- *Tasking combination functional manager/IPT lead*
- *Historically large projects have “large” IPTs (e.g. 30-50)*
- *Sub-IPTs employed for specific tasking*

Step 2: “Who” – Organization & People (cont)



**Agile:
All same functions required**

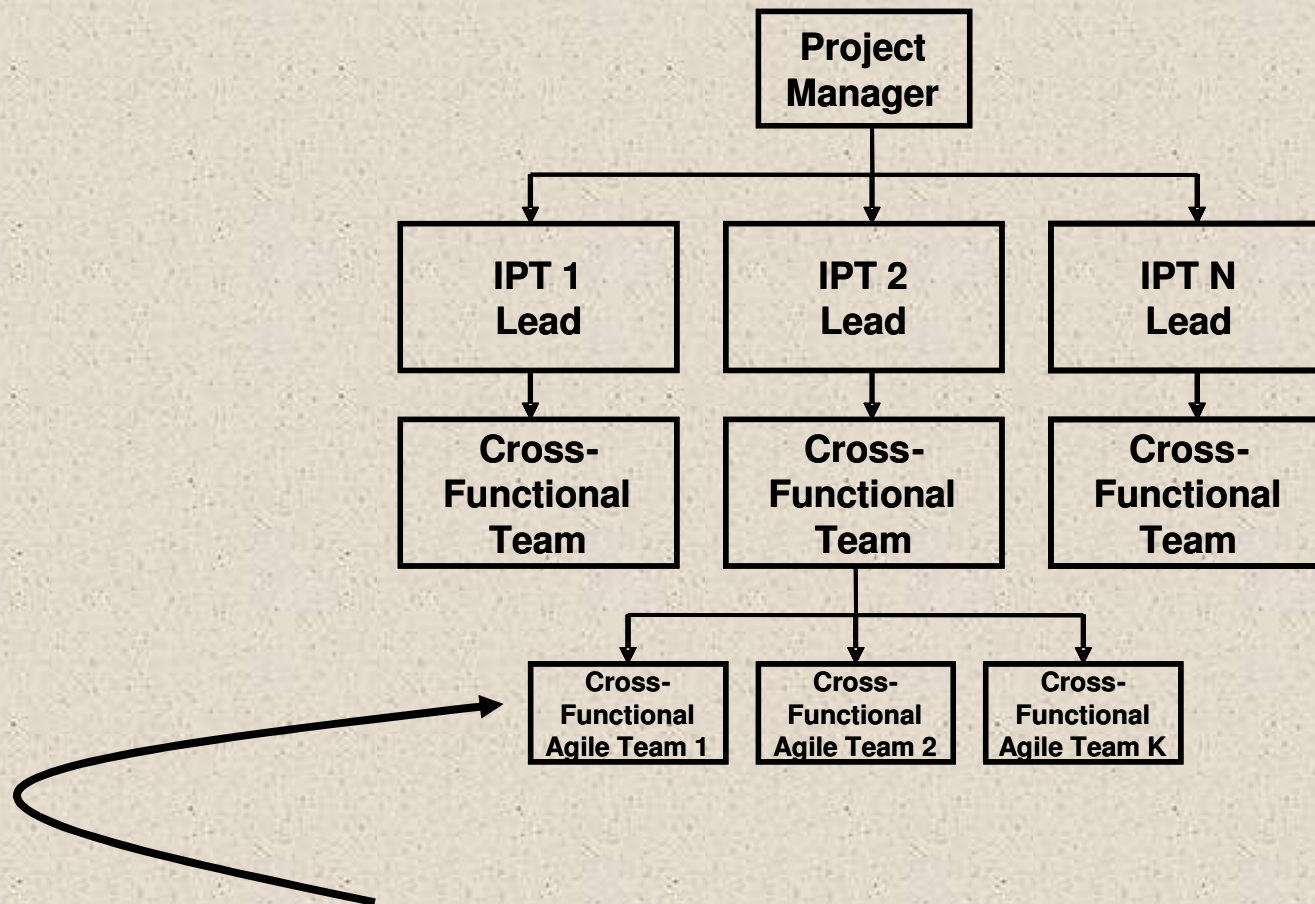
Potential Differences:

- 1. People interactions**
- 2. How receive tasking**
- 3. How communicate status**
- 4. Team size**

**Degree of Difference:
DEPENDS**

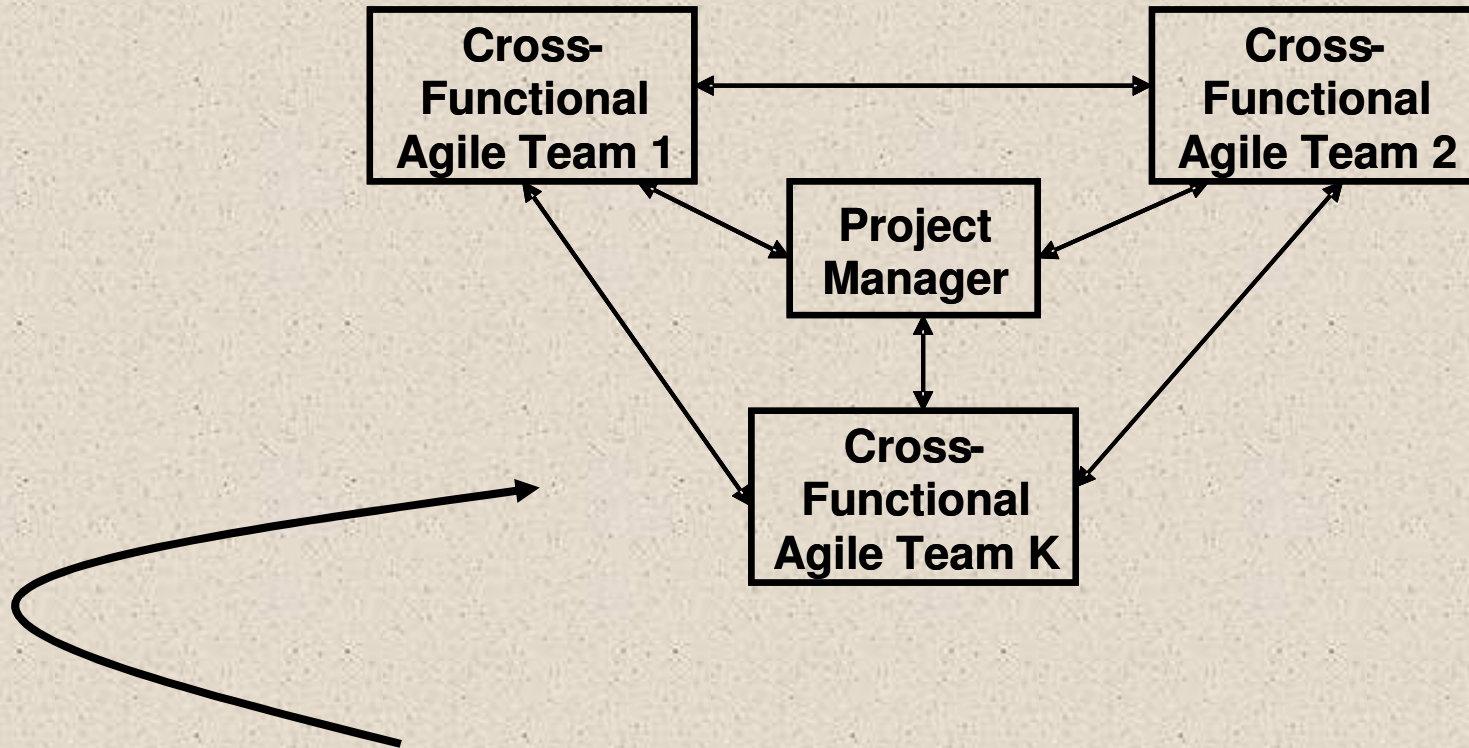
***Note:
2 Case Studies,
1st 2 views***

Agile Organizational View 1



- *Similar to IPT*
- *Agile team's operate as extension of IPT*

Agile Organizational View 2



- *Looks very different – “Hub” structure**
- *Drawn this way to stress*
 - *Interactions not – 1 way flow*
 - *Team involved in own tasking & estimation*

➡ **But how Different: DEPENDS**

**Highsmith*

Two Organizational Case Studies



Client 1

“Unspoken agile subculture”



*Small teams form informally
“below radar”*

*“Software Engineer helps
with System Spec”*

*Functional Manager
tasking is high level*

“Integrated Tasking Model”

Client 2

*Systems Tasking strictly
through Systems
Engineering Manager*



*Systems Engineers who report only
to SE Manager write specs in
private office*

*“Software Engineer not allowed to
touch a System Spec”*

Changes take forever...

“Strict Hierarchical Tasking”

Very different...Both see value in agile...Seem surprising?

Different Reasons For & Challenges With Agile

“Unspoken agile subculture”

◆ Case 1

- ✓ Individuals willing to make decisions, but sometimes too quickly
 - Instability
 - Late night heroics

Looking for a little more thought before act

- ✓ Reason: Visibility of decisions
- ✓ Challenge: Self-Manage Work

Know when to “raise up”

“Strict Tasking..”

◆ Case 2

- ✓ Not responsive to customer needs
- ✓ Reason: Changes take too long
- ✓ Challenge: Move decision making down & increase collaboration

Note: Not Easy.. Means getting them out of office...

**Ask Yourself: “Where is My Organization?”
Which direction do you need to move?**

Step 3: “When” -- Life Cycle & Schedule

Note: The “What” was a static breakdown. “When” is time-phased

◆ Traditional Approach

- ✓ Partition work into requirements, design, implementation, test, detailed tasks, & schedule

◆ Agile Approach

- ✓ Functions same
 - When may be different
- ✓ Still need high level requirements allocation early

Issue: What to DEFER & how much to defer?

Case Study in Article:

• Driven by Customer Priorities

• Key: Keep it on the schedule

Common Mistake: Deferring Work But Taking Earned Value Now

What Are Valid Reasons To Defer Work?

Reasons To Defer Work

- 1. Customer Priorities**
- 2. Reduce Risk**

*Example: High risk early, but
“chunk-it-up” & “defer” pieces
Supports reduced risk of rework, by doing
“just-in-time” when information is best*

**Repeat Caution!
Potential Disadvantage to Deferring Work:
INACURRATE STATUS**

*Culture
dependent*

**Must have mature earned value system/culture
that doesn't assume a waterfall approach**

Scope



When Does It Make Sense to Not Fully Scope All The Work Up Front?

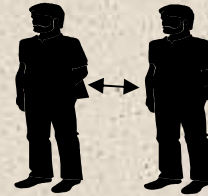
Scope Recommendation

Look at your situation



OK to not fully scope

Customer & Contractor working collaboratively for greatest value for available budget/schedule



Caution on not fully scoping

Historical adversarial customer-contractor relationship

- ◆ My recommendation is to always fully scope the work up front

- ✓ Provides a “reference point” for discussion
 - Helps with identification of “out of scope” work

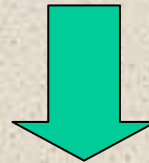
Can be done with varying levels of detail

Ultimately decision should depend on relationship with customer

Schedule

**How Can Agile Help Schedules
Become More Accurate?**

*Can help
selling
Agile to
Management*



Practical Guidance...

**If schedules accurately reflect the truth,
you may not need to change**

....but if they tend to often be out of date...

How Agile Can Help Schedules

Scenario 1

Project Lead:

“ We should have less detail in our schedule because we are agile ”



Senior Manager:

*“No!
I am afraid we will lose control!”*



Often leads to stalemate

Scenario 2

Help small teams self-manage work through task lists, team meetings



More accurate reporting up-the-chain leads to more accurate schedules



Management starts asking for less detail at the top



Over time schedules become more “agile friendly”

Step 4 – “How” – Tools & Motivating Teamwork



◆ Traditional Approach

- ✓ Plan up front
 - Reviews
 - Methodology
 - Tools

◆ Agile Approach

- ✓ All still must be planned, but focus shifts
 - Personal interactions
 - Real project status



**Two Case Studies
related to tools**

Tool Case Studies



Case 1

“Client that uses Scrum”



Reporting improved status

Task lists = “sticky notes” on Conference Room walls



“Commercial Tool”



Degraded Status



Tool not easy to use

Note:

Common agile practice – sign up for tasks to help commitment

Case 2

Client doesn’t refer to self as agile



“Commercial Task Tool”



Many don’t like Tool



Not “self-directed” but achieve intent



Helps distributed team members

**Where is your organization?
Which direction do you need to move?**

Motivating Teamwork



Self-Directed Teams:

- People sign up for tasks



“Why should I do my job, & yours too?”



Rating components:

- Customer
- Team
- Company perspective

- ◆ Jeff Sutherland
 - ✓ Weighted average individual performance review process



Can be done with agile or traditional approaches

Step 5: “How Much” – Cost & Metrics

- ◆ Don't yet know fully how cost affected
 - ✓ Mistake to think costs less because less engineering
 - ✓ Partition work to do it at best time
 - ✓ Should be less cost due to less rework



“By going agile it will cost less because we have less to do!”

Can be done with “hybrid-agile” & traditional approaches as well

Agile & Burn Down Charts

- ◆ Some organizations – traditional & agile-- use burn down charts to show schedule
 - ✓ Strict agile means burn down charts owned by team, not functional manager
 - ✓ Another side
 - Outside view can help
 - Team member experience

Ask:

• *Who owns burn down charts?*

• *Team's view?*

• *Separate Mgr view?*

Not necessarily bad depends on culture & project conditions



**Next Set of Questions To Ask:
Hitting Schedules? Satisfied Customers?**

Which direction do you need to move?

Agile Tailoring of Burn Down Charts for U.S. Government Projects



- ◆ Strict agile means only report progress when code tested & works
- ◆ Recommend tailoring to include ALL Work
 - ✓ Documentation
 - ✓ Reviews...

When burn down chart hits zero..

*I want it to mean we are really **DONE!***



Agile & Lines of Code Metric

- ◆ Agile doesn't use lines of code as a measure of productivity
- ◆ Nor do many traditional organizations
 - ✓ Problems well documented
 - ✓ But there is another side

Tracking lines of code changed from one build to next can be useful trend indicator



• *Build stability*

• *How close to really done*

No difference on agile or traditional project

Conclusion

Five Basic Steps of Planning Remain, But With Agile Methods They May Be Carried Out With Some Key Differences

- ✓ Opportunity for more accurate project status through self managed teams

May be least understood, recall earned value cautions

- ✓ Opportunity for more rapid change processing

Keep in mind as move decision making down affects skill needs (e.g. collaboration)

- ✓ Case studies indicate hybrid agile-traditional approaches often appropriate

How different agile project management is depends on your organization

Many Degrees of Agile Anticipate Many Decisions along the Way

Contact Information & References

◆ Contact Information:

- ✓ Email: pemcmahon@acm.org;
- ✓ Web: www.pemsystems.com; Ph: 607-798-7740

◆ References:

- ✓ McMahon, Paul, Are Management Basics Affected When Using Agile Methods?, Crosstalk, Nov, 2006
- ✓ Highsmith, Jim, Agile Project Management, Addison-Wesley, 2004, pp 239-240
- ✓ McMahon, Paul, Integrating Systems & Software Engineering: What Can Large Organizations Learn From Small Startups?, Crosstalk, Oct, 2002
- ✓ Cockburn, Alistair, Agile Software Development, Addison-Wesley, 2002, 2006, 1st & 2nd Editions
- ✓ Schwaber, Agile Project Management, 2003

Acronyms

- ◆ WBS = Work Breakdown Structure
- ◆ IPT = Integrated Product Team