



Systems Engineering Complexity & Project Management

Bob Ferguson, PMP
NDIA: CMMI Technology Conference
November 2007



Outline

A conversation

Defining complexity and its effects on projects

Research into tools and methods



A Conversation

Manager: How big is this project?

Developer: I don't know. This looks really hard.

Manager: Well we need to know how big it is so we can estimate the work.

Developer: I'll have to figure out how hard it is so I can tell you how long it will take.

These two are talking about different things.

The developer believes that his estimate of size, will not recognize the uncertainty. He wants to know something about the complexity to adjust duration



Project Manager's Concern

The project manager is concerned with staffing and planning to meet the project's objectives.

The project manager may not understand what the engineer means by complexity.

- He may interpret the behavior as complaining.
- He may think “He always says that, but it doesn't help his estimate.”

The project manager does not know what questions to ask, nor has he thought sufficiently about engaging the SE in project planning.

How do we create a new “conversation”?



Analyzing the Questions

What do we mean by the word “complexity”?

What methods can help project managers resolve complexity?

What information can teams provide that shows the resolution of the complexity?

How should the project manager question the staff to identify the complexity?



Way forward

The project manager and engineer can deal with the complexity problem, provided that each understands and accepts the other's concern.

- The project manager asks the right kind of question.
- The project manager is amenable to creating a plan that will allow for resolution of the complexity by the engineering staff.
- The engineer understands how the project plan might help to mitigate the schedule and cost problems that result from complexity.
- The budget and schedule are not so tightly constrained that the project cannot be accomplished.

The remainder of this talk will describe some planning actions to help resolve select types of project complexity.



Outline

A conversation

Defining complexity and its effects on projects

Research into tools and methods



Existing Complexity Research

Research considering the complexity in product development projects

- Business Schools
 - Steven Eppinger and Nelson Repenning at MIT
 - Kim Clark at Harvard
- Engineering School papers
 - Ali A. Yassine at Univ. Illinois Urbana-Champaign

Research has considered task structure in response to complex problems in product development ..



Defining Complexity

Definitions seem to relate to the difficulty in learning a capability that a team or individual does not currently possess.

- “McCabe complexity” indicates difficulty in learning to maintain a set of code.
- “Technology introduction” entails learning a lot of different things: design, testing, technical communications, manufacturing, ...
- Invention is discovering (learning) a new design pattern

Resolving the complexity depends on some learning process –

- The organization must develop new capabilities.
- Some iteration or experiment is required for a satisfactory solution.
- The team must learn to work together.



Types of Product Development Complexity

Different learning requirements suggest an approach.

Big

The work has to be divided into teams or sub-projects in order to produce a result soon enough that it has value.

Deep

An unfamiliar design pattern is required. It may even require a new invention.

Conflicting Goals (Design Tradeoff)

Problem requires some form of experimentation, prototyping or other trade-off analysis. An optimal (but not perfect) solution is expected.



“Big”

Large projects require multiple work groups operating simultaneously and somewhat independently.

Potential Problems

- Synchronizing the work is very difficult. Teams must sometimes start work on incomplete information.
- Individuals who fail to fully participate in the work of integrated product teams (IPTs) place additional burdens on the other teams.

Things to be learned:

- Team boundaries (“We do this. You do that. Here’s how we decide.”)
- How to handle incomplete information
- How to declare completeness
- How to verify and validate each other’s work



Project Management Concerns for “Big”

Needs a picture of team and wbs structure
Relationships show learning

Structuring the teams

- Balance the workload to achieve desired schedule
- Teams have needed skills and resources

Product Concerns

- Sufficiently many integration points to demonstrate learning and product progress (depends on system architecture)

Required activities

- Learning to work together (say 8-24 hours face-to-face time)
- Specific understanding of interfaces and boundaries
- Describing exactly what is incomplete and how the act of completing may affect current results.



Dealing with Change on “Big” Projects

Consider that “Change Requests” are an out-of-cycle development request.

- i.e. some design work is already completed and now has to be re-done.

Considerations

- Affected work products
- Affected teams
- Coordination aspects
- Ripple effects



Measures for Big Projects

IPT

- Participation and battle rhythm
- Convergence on interfaces
- Issues and rework on interfaces
- Decision bottlenecks
- Design structure matrix to show distance between team members

Architecture

- Design structure matrix to show interdependency
- Structure for integration/verification



“Deep” problem

An aspect of the design is new to the development team.

Potential problems

- Capability to perform may be missing or have limited capacity for work.
- Productivity suffers and team generates a lot of rework.
- Lack of progress affects other teams and causes synchronization problems.

Things to be learned

- What technology works (algorithm, material, equipment, technique)?
- How and when does it work?
- How do we utilize it in the current product development project?
- Do we want to develop capability and capacity or buy it?



Examples

The first use of a genetic algorithm in the application.

- Who must understand the mathematics?
- How long does convergence take?
- How can we test the convergence and result?
- What do we need to document for maintenance?
- What unique bugs could occur in this type application?
- How will this technology affect manufacturing and setup?



Project Management Concerns for “Deep”

Deep problems take time, but not many people.

- Some very highly skilled individuals will not be available to the larger team for while the deep problem is addressed.
- If these people multi-task, the time required will be much longer.

Required activities

- A deep problem is not “solved” until the organization can utilize the technology to produce the final product.
- Technology transfer tools, events and mentoring

Costs and Risk Mitigation require investigating alternative solutions.

- Alternative implementations may be needed in the interim, but may not fully meet quality attribute objectives.
- Buy required technology and/or development capacity (risk transfer)



“Conflicting Goals” (CG)

Some stakeholder values are in apparent conflict.

- More power and less fuel consumption
- Faster performance and more security
- Flexibility to install devices and information assurance
- Faster product delivery and more robust design

Conflict may be between stakeholders increasing the difficulty

- Theory of Constraints work may help with conflict resolution



Conflicting Goals Problems and Learning

Potential problems

- Separate teams may attempt to achieve the goals independently. Each team then changes the resulting system behavior in some way opposite the other's goal.
- Slow decision process
- Usually requires multiple iterations for resolution.
- Conflict not exposed soon enough for appropriate resolution.

Things to be learned

- What are the important interactions? What values work?
- What are the sensitivity points and trade-offs inherent in our design (architecture)?
- How can we see that our required iterations are converging?



Project Management of Conflicting Goals

These problems always require some form of experimentation.

- Experiments include simulation, scenario analysis, trade studies and prototype products
- There is a cost to experimentation that can be hard to plan.

Required Activities

- Identify sensitivity points and trade-offs.
- Check modularity against team structure so that decision involves as few teams as possible.
- Plan some number of iterations before capability is required.
- Create extra integration points to show that complexity was actually resolved.
- Consider transforming problem into a “deep” problem. (Find a technological approach).



Outline

A conversation

Defining complexity and its effects on projects

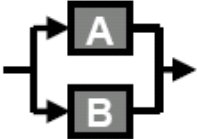
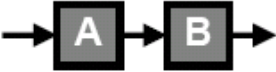
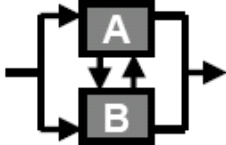
Research into tools and methods



Modularity Methods

Design Structure Matrix (DSM)

- DSM has proved to be a fairly successful approach to partitioning and analyzing very large systems. (picture)

Three Configurations that Characterize a System			
Relationship	Parallel	Sequential	Coupled
Graph Representation			

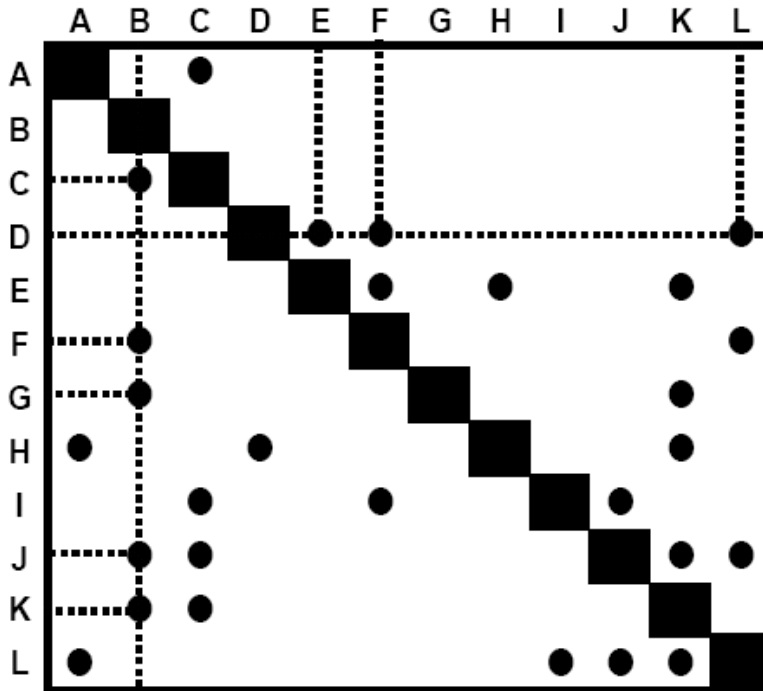
Three Configurations that Characterize a System																														
Relationship	Parallel	Sequential	Coupled																											
DSM Representation	<table border="1"> <tr><td></td><td>A</td><td>B</td></tr> <tr><td>A</td><td>■</td><td>■</td></tr> <tr><td>B</td><td>■</td><td>■</td></tr> </table>		A	B	A	■	■	B	■	■	<table border="1"> <tr><td></td><td>A</td><td>B</td></tr> <tr><td>A</td><td>■</td><td>■</td></tr> <tr><td>B</td><td>X</td><td>■</td></tr> </table>		A	B	A	■	■	B	X	■	<table border="1"> <tr><td></td><td>A</td><td>B</td></tr> <tr><td>A</td><td>■</td><td>X</td></tr> <tr><td>B</td><td>X</td><td>■</td></tr> </table>		A	B	A	■	X	B	X	■
	A	B																												
A	■	■																												
B	■	■																												
	A	B																												
A	■	■																												
B	X	■																												
	A	B																												
A	■	X																												
B	X	■																												

DSM Types and Methods

DSM Data Types	Representation	Application	Analysis Method
Task-based	Task/Activity input/output relationships	Project scheduling, activity sequencing, cycle time reduction	Partitioning, Tearing, Banding, Simulation and Eigenvalue Analysis
Parameter-based	Parameter decision points and necessary precedents	Low level activity sequencing and process construction	Partitioning, Tearing, Banding, Simulation and Eigenvalue Analysis
Team-based	Multi-team interface characteristics	Organizational design, interface management, team integration	Clustering
Component-based	Multi-component relationships	System architecting, engineering and design	Clustering



DSM Example

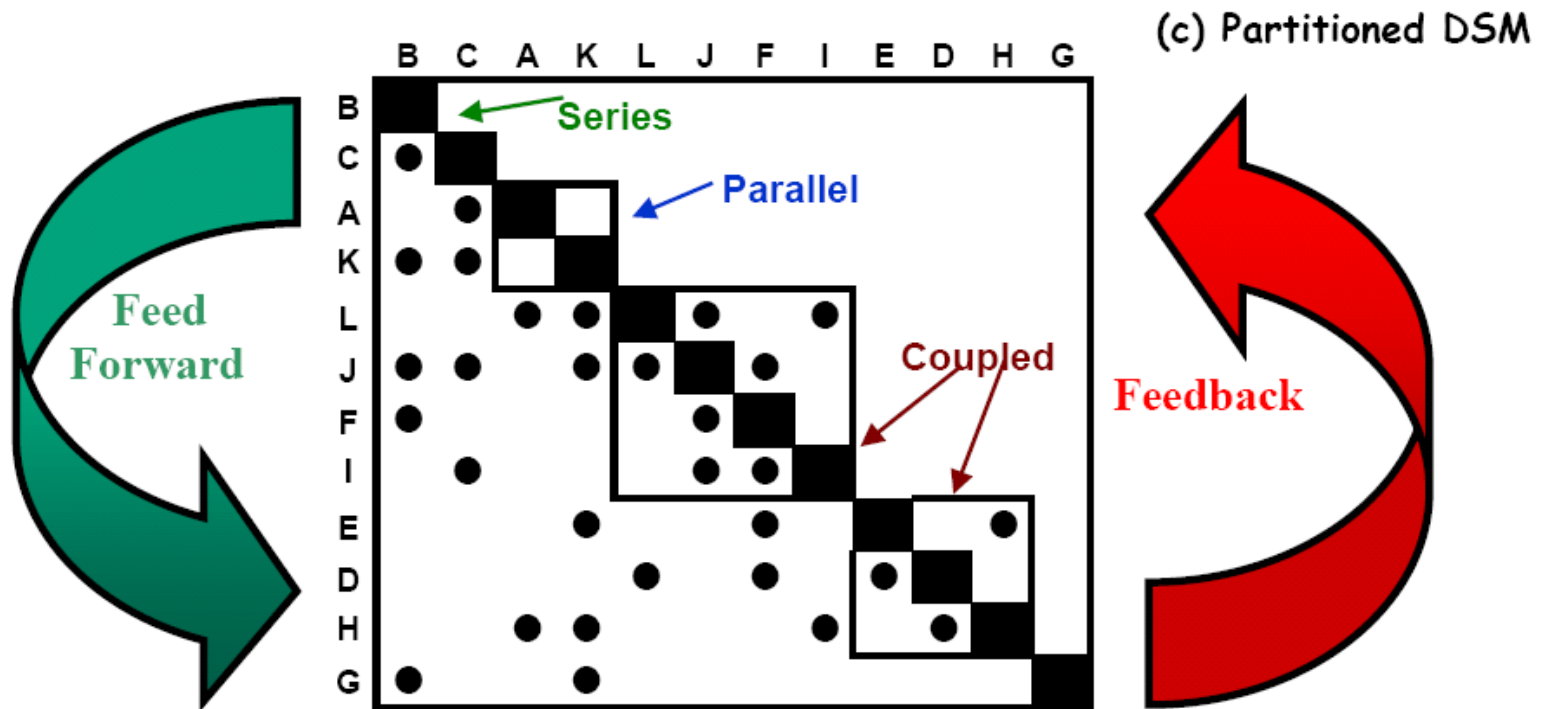


This matrix represents one with a lot of complexity.

Modularity and team arrangement is not clear.

By re-ordering the matrix we can achieve a better team structure and better modularity of both task and design.

DSM Example



Project Planning Strategies

Scheduling

- DSM provides useful information about
 - Team interdependencies – requires exchange of incomplete knowledge and active participation
 - Component interaction – Requires documentation and tests
 - Iteration – requires planned extra steps

Team Learning

- Joint scenario work
- Simulations of work flow
- Joint inspections
- Facilitators for planning



Known Methods for Deep and CG Problems

Best known method **TRIZ** (treez)

- Addresses both “Deep” and “Conflicting Goals”
- Consists of ~50 strategies for innovation and problem solving.
- Applies mostly to hardware engineering
 - Such as physical separation of function
 - Time-dependent separation of function

QFD relates design goals to design with cost elements and exposes conflicting goals

QAW, ATAM expose many conflicting goals problems

Design Structure Matrix

- Has potential for mathematical approaches such as “work-eigenvector” and simulation of task structure to represent high complexity elements.



Project Planning for Deep

Dedicated, highly skilled resources

Knowledge transfer, process implementation

- The new technology has to be adapted to the rest of the product development team. It may require additional resources.

Validation of utility of results (testing, learning, etc.)

- New capability will include design patterns, test patterns, documentation skills, customer support skills, etc.

Highly skilled resources are not always good at technology transfer. Senior engineering management, developers and testers all need to learn something from a deep problem. Some participation in progress reviews and experiments needs the support of these other people.



Risk Reduction Methods for Deep Problems

Alternative method

- Parallel teams attempt different solutions
- Purchase products or the development capacity from outside

Experiments

- Trade studies, prototypes, simulation

Project management consideration

- Resolution of deep problems has to start as early as possible or the schedule will grow while capability and capacity problems are resolved.
- All methods associated with deep problems have the possibility of taking a very long time to resolve.
- It is essential to have a reasonable method at the time of integration even if the solution is not optimal.



Interactions of Complexity Type

Partitioning of “Big” can aggravate “Conflicting-Goals.”

- Separation of concerns approach may allow engineers to view their responsibility for <quality-attribute-A> as independent from <quality-attribute-B> resulting in a sub-optimal design.

Sometimes work on “Deep” problems results in “Big” or “Conflicting-Goals” problems.

- As when the primary solution to the Deep problem is to partition it into several other problems.

Some “Conflicting-Goals” problems can be addressed algorithmically resulting in a “Deep” problem.



CMMI Relationship

IPPD goals address the Big problem and Conflicting Goals problem

- IPT structure is key
- Must monitor IPT learning and non-learning (issues, etc.)
- IPT must discuss content as well as schedule if members are to learn.
- Integrated Product concept has to be at the forefront of the project manager's attention as the primary near-term goal for each IPT.

Technical Solution in the CMMI

- CMMI does not satisfactorily address Deep problems.
- We must include specific efforts to develop the competencies and capabilities of staff and process to introduce a technical innovation.
- Even choosing an outside supplier for the solution requires development of new internal capabilities.



Staged Representation (ACQ in blue)

Maturity Level	Process Areas
Optimizing	Causal Analysis and Resolution Organizational Innovation and Deployment
Quantitatively Managed	Quantitative Project Management Organizational Process Performance
Defined	Organizational Process Focus Organizational Process Definition Organizational Training Integrated Project Management Risk Management Acquisition Technical Management Acquisition Verification Acquisition Validation Decision Analysis and Resolution
Managed	Acquisition Requirements Development Agreement Management Project Planning Project Monitoring and Control Requirements Management Configuration Management Process and Product Quality Assurance Measurement and Analysis Solicitation and Supplier Agreement Development



Summary

We can teach project managers and systems engineers (architects) to talk with each other about complex problems.

This talk described complexity as 3 different type problems.

- Big, Deep, Conflicting Goals

Addressing each type of complexity calls for different project management strategies.

Each strategy must address the technical problem, product integration, learning events and the project social network.

- We need to identify ways to monitor that the development team is actually learning as a means of checking progress.



Questions?

Bob Ferguson, PMP

rwf@sei.cmu.edu

