Beyond IPPD: Distributed collaboration in a Systems-of-Systems (SoS)-context

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

SuZ Garcia, Urs Andelfinger
SEPG Europe - 13 June 2008
Agenda

• Setting the stage
  • Governance-Demand Challenge and IPPD
  • Beyond CMMI V1.2 IPPD
  • Summary and conclusions
  • Backup
Questions from our abstract

• How do you resolve conflict among suppliers and customers when you (especially you the supplier!) do not have a direct line of authority?

• How do you navigate the different governance structures that lead to different processes being put into use that may not be compatible?

• How do you respond to the customer's "real" needs if you're a supplier?

• How do you know who to tell your real needs to if you're on the operations side?

We’ll answer these by looking at characteristics of complex systems of systems that drive issues in these areas, then look at possible solutions and the implications of those for IPPD content in CMMI v1.2
“Simple” definition of system of systems

A collaboration among autonomous systems (both technical and organizational) ...in relation to some use ...within a changing, unpredictable context
Achieve an effective emergency response: an intuitive example

A collaboration among autonomous systems (both technical and organizational) ...in relation to some use ...within a changing, unpredictable context
Some Problems Faced in Distributed Collaboration in a SoS context

- Unanticipated System Behaviors
- Supplier’s capabilities can’t keep up with user’s demands
- Unknown Participants
- Latent Incompatibilities
- "Common Terms" are understood differently
- No Universal View
- Supply and demand are no longer aligned
- Synchronization Challenges

An ‘IPPD’-attitude is required to successfully master the problems of distributed collaboration

... but IPPD ‘as is’ (within CMMI v1.2) does not completely address these problems
Agenda

- Setting the stage
- Governance-Demand challenge and IPPD
  - Beyond CMMI V1.2 IPPD
  - Summary and conclusions
  - Backup
Distributed Collaboration leads to an Agile World where the Customer Experience drives the solution

- Customers and users want specialized solutions in ever-shorter time frames, continuously adapted to their changing and evolving situations.
- Suppliers and systems have to become more agile to respond.

Product-Based

Users want products or services that can be provided in a way that is unaffected by how they are used

Solution-Based

Users want integrated solutions that are customized to their context, but in a way that can be specified beforehand

Customer Experience-Based

Users want integrated solutions that are customized in ways that change and evolve throughout the life of the mission that they support

‘Turbulence’ as per [Emery 1965]
Categories adapted from [Prahalad 2003]
Distributed Collaboration in a SoS context leads to concepts of Distributed Governance

- Number, type, and roles of participants are increasingly diverse, reflecting differing vested interests.
- Scarce resources and the need for concurrent uses make a single decision authority increasingly unlikely.

**Single Task “System”**

A single program directs composition
—little potential for conflict

**Single Enterprise**

A real or virtual entity directs how multiple entities collaborate to compose multiple programs
—resolves potential conflicts by imposing constraints

**Multiple Enterprises**

Multiple real or virtual directing entities making competing demands on SoS
—conflict resolution requires negotiating mutual constraints

Requires a fundamental shift
Combining Customer Experience-driven solutions and Distributed Governance leads to a Double challenge

2 - Collaborating effectively across enterprise boundaries

1 - Developing flexible responses to unanticipated situations

Nature of Response to Demand

Product based
Solution based
Customer experience based

Disruption due to addressing the multi-enterprise governance context
Disruption due to diverse demands from dynamic contexts-of-use

Comfort zone for traditional engineers, managers, and users

Single enterprise / Integrated teams
Multiple Enterprises

Single task system

[Boxer 2006-1]
IPPD currently addresses, at most, the product- and the solution-based responses to customer demand

IPPD (Integrated Product and Process Development) in CMMI V1.2:

- A *systematic approach* to product development that achieves a *timely collaboration of relevant stakeholders* throughout the product lifecycle to better satisfy *customer needs* (CMMI Glossary).

**Critical concepts contained in that definition:**

- *systematic approach*: guided by a priori defined principles, plans and patterns of action
- *timely collaboration*: collaboration relationships planned a priori (mainly synchronous)
- *relevant stakeholders*: stakeholders a priori identified, included in a plan and accordingly involved
- *customer needs*: assumes that the customer needs are systematically known and committed to a priori
IPPD currently builds on *integrated teams*

**Integrated team:**

- A group of people ... who are *committed to delivering specified work products in timely collaboration*. Integrated team members ... are *collectively responsible* for delivering the work products *as specified*. An integrated team should include empowered representatives ... that have a stake in the success of the work products. (CMMI Glossary)

**Critical concepts contained in that definition:**

- *Commited to delivering specified work products*: commited with respect to the a priori defined shared vision and product specifications
- *Timely collaboration*: means collaboration according to the plans
- *Collectively responsible ... as specified*: responsibility relates to the a priori defined specifications
IPPD currently assumes the existence of clear governance mechanisms and authority

Central Governance paradigm:
- IPPD assumes the existence of an organizational infrastructure with clear channels of responsibility and authority. (OPD, SG 2 & SP 2.1)
- This assumption is reinforced by the CMMI-definition of 'organization':
  - An administrative structure in which people collectively manage one or more projects as a whole, and whose projects share a senior manager and operate under the same policies. (CMMI Glossary)

Critical concepts contained in that definition:
- Clear channels of responsibility and authority: even if multiple enterprises are involved, a single source of ultimate authority is assumed
- Managing projects as a whole: assumes that the same norms are being used across a portfolio of projects
- Sharing a Senior Manager: assumes a single enterprise viewpoint of organization
### Critical Leverage Points for IPPD in a SoS-context

<table>
<thead>
<tr>
<th>Current CMMI 1.2 IPPD</th>
<th>IPPD for a SoS-context</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Governance &amp; Empowerment</strong></td>
<td><strong>Multi-party, decentralized governance</strong></td>
</tr>
<tr>
<td><em>Central Governance, planned a priori</em></td>
<td><em>Processes may be asynchronous; decoupling of development and integration tempos</em></td>
</tr>
<tr>
<td><em>Timely collaboration as planned a priori</em></td>
<td><em>Willingness and capability to integrate into dynamically evolving contexts</em></td>
</tr>
<tr>
<td><em>Shared vision and shared responsibility are key, defined a priori</em></td>
<td><em>Understanding the variety of demand contexts more important than trying to “pin down” a single set of requirements</em></td>
</tr>
<tr>
<td><em>Known and defined a priori</em></td>
<td><em>Interface conformance and Interoperability is more important</em></td>
</tr>
</tbody>
</table>

**Software Engineering Institute | Carnegie Mellon**
Agenda

- Setting the stage
- Governance-Demand challenge and IPPD

- Beyond CMMI V1.2 IPPD
- Summary and conclusions
- Backup
OEI SP 2.1: *Establish Leadership Mechanisms.*

This SP accounted for loose forms of governance like consultative and collaborative. Might be reformulated: *'Establish collaboration mechanisms'*

OEI SP 2.2: *Establish Incentives for Integration.*

- We have to compensate the loss of central governance by other appropriate means to motivate and promote collaborative behavior. Might be reformulated: *'Establish social rules / codes of conduct'* that every participant in the community commits to adhere to.
IPPD in a SoS – context
Implications for Governance & Empowerment

- We will lose the clear coordination and sanctioning mechanisms of a central governance authority
  - Instead, we will have to rely more on 'social' mechanisms to enforce effective collaboration
    - Trust will become more important
    - To foster trust, adopt mechanisms of the 'social software world' and of the 'open source world'
- Understand which processes are critical to interoperate among the involved enterprises
  - Need mechanisms for agreeing on process governance
- Empower the system component producers to take decisions appropriate to the situational needs – even if they were not planned a priori
  - Understanding the variety of ways that customers will use and engage with our products is key to going beyond pre-specified requirements
IPM SP 3.1: *Define Project’s Shared vision Context.*

- This SP explicitly acknowledges that we are a project with interfaces that does not operate in isolation. It also stresses the fact that we might not be responsible for all results of the overlying mission. Instead of trying to govern the whole mission we should restrict our efforts to clearly defined boundaries.

- If other constituents external to us make use of our products or services, we can abstract from it as long as we obey to the interface requirements.

- The growing importance of ‘developing against standards‘ instead of ‘developing against requirements‘ should be dealt with in a reformulated SP.
IPPD in a SoS – context
Implications for Collaboration & Integrated Teams

The concept of an integrated team will take on a different meaning:

- Loosely coupled teams will emerge with no clear lines of authority
- More negotiation skills than a-priori planned procedures and rules
- A differentiation of the roles will be likely:
  - System component producer
  - System integrator/synthesizer who integrates the system based on system components
    - This may very well be someone from within the traditional „customer“ environment
- Prerequisites on the side of the system component producer:
  - Willingness / readiness that his components will be integrated also in contexts that were never heard of or thought about before
  - Capability to deliver to the defined interface standards
  - Capability to characterize and understand the varieties of demand their systems will be subject to
Potential Change for CMMI (as opposed to enriching with CMMI V1.1)?

Generic Practices

- Add data management generic practice (similar to Registry Services in the SOA world)?
  - To highlight the need for being explicit about process data available & being shared
IPPD in a SoS context
Implications for Requirements & Operational Concepts

• Importance of interoperability of system components is growing, which includes growing importance of interface descriptions and conformance to standards; however, standards are not enough.

• Instead of 'development against requirements', 'development against a varied set of capabilities, tuned to expected situations', is becoming more important.

• Conformance tests to standards / standard test suites is becoming more important at system component level.

• Use of pilots and prototypes to cope with phenomena like emergent system behaviour can help test reasonableness of our suite of expected situations—not every system behaviour can be exactly planned for a priori nor tested exhaustively.

• Usage of industry-wide accepted reference models might become more important (healthcare industry is trying to go this direction).

• Need to look at multiple layers of interoperability.
We Need to Address Multiple Layers of Interoperability

• As we move up the stack
  • Standardization decreases in importance
  • Implementation becomes domain- and organization-specific
  • Implementation experience tends to decrease
  • Machine-to-machine interaction becomes less important, but the rest of the stack builds on it
Agenda

- Setting the stage
- Double challenge and IPPD
- Beyond CMMI V1.2 IPPD
  - Summary and conclusions
- Backup
Summary

- Whether you talk about complex systems, families of systems, systems of systems, etc., the characteristics that distinguish these systems and drive much of their complexity also drive a different character of solutions than we’ve focused on in the past:
  - More focus on layers of interoperability above behavioral semantics
  - Distributed governance, not just distributed teams, is needed
  - CMMI—DEVv1.2 is necessary, but not sufficient
    - Some additional Support comes from v1.1 content
    - Some issues are outside of scope of current CMMI instances
Next Steps

• TN in work: *Process Considerations in Systems of Systems*

• Processes, Tools, Techniques to support operating effectively in systems of systems environments: *Systems of Systems Navigator*
Related Materials

- Tutorial SEPG 2006: CMMI in a SoS Context
- Presentation SEPG EU 2007: SURVIVAL in a SoS World
- TN 2008: Process Considerations in Interoperable Acquisition

Please contact Suzanne Garcia if you have trouble finding any of these....
For Additional Information

http://www.sei.cmu.edu/isis

Suzanne Garcia  smg@sei.cmu.edu

Urs Andelfinger  ua@sei.cmu.edu
THANK YOU!!!