Disciplined Agility for the Development of Software-intensive Systems

Dr. Peter Hantos, I.S.T.J.
Consultant
Yet Another Manifesto

(1) I believe in Planning
(2) I believe in Architecture
(3) I believe in Documentation
(4) I believe in Process over People
(5) I believe that the analysis of myths and abuses, rather than relevant concepts led to the declaration of the Agile Manifesto
(6) I believe that the followers of disciplined methods must become more agile, but the methods leading to Disciplined Agility are not necessarily the ones publicized by Agile Methodologists.
Why do I believe these?

- Because I am an I.S.T.J.!

I.S.T.J. is my Myers-Briggs Personality Type Indicator:
- For energizing: Introversion instead of Extroversion
- For attending: Sensing instead of iNtuition
- For deciding: Thinking instead of Feeling
- For living: Judgment instead of Perception
I.S.T.J. People

- **Work Setting:**
  - Prefer focusing on facts, details, results
  - Want structure, order, and privacy for concentration without interruptions

- **Organizing Style:**
  - Usually have a great deal of factual information to deal with, and they take pains to properly label and file it. They also put emphasis on cross-referencing and easy retrievals

- **Leading:**
  - They respect traditional hierarchical approaches, and seek to reward those who get the job done by following the rules and standard operating procedures

- **Many Engineers and Engineering Managers are I.S.T.J.**
  - ... so their preferences matter!
Myth - The Big Bad Customer

- **Old Myth**: Our problem is that he can not make up his mind, hence requirements are constantly changing
  - **Reality**: The term “requirements” is not fully understood and it is misused. Most requirement changes are in fact specification changes, resulting from deeper understanding of the otherwise frozen requirements and the system’s architectural intricacies.

- **New Myth**: We should welcome if he changes his mind (Anyway, he *is* the Customer, and he is always right...)
  - **Reality**: The customer’s requirements must be stable enough to facilitate the development of a feasible and viable architecture. Assuming that the architecture of a serious system can be refactored on an ongoing basis is a suicidal management approach.

- **Corollary**: “Just because you are paranoid does not mean that they are not after you” - There ARE some Big Bad Customers...
Myth - It Is Strictly Our Issue

- Myth: Software managers are free to choose any agile methods and processes, since this is strictly a “private”, project matter
  - Reality: In the case of any serious product or system development numerous non-software organizations are involved and intricate interdependencies exist (See next slide)
  - Reality: Process owners/managers of activities on the critical path of a project should have even less freedom in choosing risky methods and processes
Example Parallel Processes

- Typical business processes of commercial, software-intensive product development:
  - Business Case Development and Financials
  - Integrated Program Planning and Management
  - Systems Engineering
  - Electro-Mechanical Development Process
  - Electronics Development Process
  - Software Development Process
  - Standards and Regulatory Compliance, Security
  - Globalization
  - Launch
  - Manufacturing and Logistics, Supply Chains
  - Third Party Arrangements, COTS, Tools
  - Marketing, Sales and Distribution
  - Customer Service and Support
  - Skills and Resource Planning
Example Parallel Processes - 2

Typical business processes of commercial, software-intensive product development:

- Business Case Development and Financials
- Integrated Program Planning and Management
- Software Development Process
- Standards and Regulatory Compliance, Security
- Globalization
- Launch
- Third Party Arrangements, COTS, Tools
- Marketing, Sales and Distribution
- Customer Service and Support
- Skills and Resource Planning
Organizational Dynamics

- **Myth**: It is assumed that the self-organizing and emergent characteristics of agile teams assure that the organization can address all needs for handling change
  - **Reality**: What Organization? The published agile methods do not comprehend the notion of “organization”*
  - **Reality**: Consequently agile methods do not address the issues of Organizational Dynamics

- **Myth**: It is assumed that - particularly on higher maturity levels - the CMMI Process Areas deal with the issues of Organizational Dynamics
  - **Reality**: CMMs don’t say anything about it either, in fact they have a fairly static view of how organizations are structured.

* **Organization - CMMI definition**: 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.
Organizational Dynamics - 2

- **Single Product Development**
  - Organizational structure usually reflects the product’s WBS

- **Multiple Product Development - Flat Organization**
  - Little cohesion between the different products

- **Multiple Product Development - Matrix Organization**
  - Program Managers draw on functionally structured teams, resulting in special resource allocation and timing challenges
    - People might work on more than one product at the same time
    - People are moving from one product to another

- **Platform-based Product Line Development**
  - There is a platform team and multiple product teams
    - Based on their domain knowledge, people might be moved between the platform and product teams as needed
    - Developers might be moved from one product team to another
Knowledge Management

- **Knowledge**
  - Dynamic ("Information in Action")
  - Absorbed by individuals via
    - Interpretation
    - Modification
    - Use

- **Expressions of knowledge**
  - **Explicit** and objective if formal expression is possible
  - **Tacit** and subjective if hard to formalize due to personal beliefs, experiences, values, viewpoints, etc.

- **Knowledge management**
  - Logical Step to develop strategies and competencies for the future
The Value of Knowledge

- **Agility**
  - Knowledge is a key aspect of the business’ ability to adapt to changing conditions

- **Innovation**
  - Tacit knowledge and interactions generate innovation
    - Innovation is essential to rejuvenation or adaptability

- **Transfer of knowledge**
  - It is a must to transfer tacit, individual knowledge to explicit, corporate knowledge
  - Selected organizational/corporate applications
    - Lessons Learned
    - Decisions Made
    - Actions Planned
    - Risks Mitigated
    - Results Correlated
Arrows represent the “Knowledge Links”, or the opportunities to utilize the information from one particular product’s Anchor Point Review at the planning of another product’s development phase and consequent AP reviews. (Blue represents indirect, Red represents direct AP relationships.)
Sweet Spots Everywhere ...

- **Myth**: If you like Pair Programming, then you should use it rigorously in every project
  - **Reality**: Programming Pairs are developing software faster and with higher quality than individuals.
  - **Reality**: With respect to productivity the trade-off is not trivial (The same software is not delivered in half time and with half defect density.)
    - It is the project manager’s responsibility to decide if the time-to-market pressure justifies the creation of pairs or not
    - Most likely the coding of those WBS elements that are on the critical path is justified in pairs, while the coding of other elements in pairs might not be the most economical solution.
A CMMI Myth...

- **Myth:** The concept of Defined Process prevents the organization from applying different processes to different situations, hence preventing the infusion of agile methods
  - **Reality:** The Organization’s* Process Asset Library contains a set of standard processes, to be used (tailored) by project teams to create their defined processes
  - **Reality:** The Organization’s set of standard processes may include multiple process architectures. See the relevant excerpts from the documentation of the Organizational Process Definition (OPD) Process Area of the CMMI.

* Please note the earlier definition of “Organization”
The Relevant OPD Details*

SP 1.2 Establish Life-Cycle Model Descriptions

- Subpractices
  1. Select life cycle models based on the needs of projects and the organization. For example, in the case of a development project, project life cycle models include the following: Waterfall, Spiral, Evolutionary, Incremental, Iterative...

...

SP 1.3 Establish Tailoring Criteria and Guidelines

- Subpractices
  1. Specify the selection criteria and procedures for tailoring

... Examples of tailoring actions include the following:
Modifying a life cycle model, Combining elements of different life cycle models, modifying/replacing/reordering process elements

* Source: Capability Maturity Model® Integration (CMMI®), Version 1.1, Staged Representation CMU/SEI-2002-TR-012
Any Questions?