Managing Iterative Software Projects

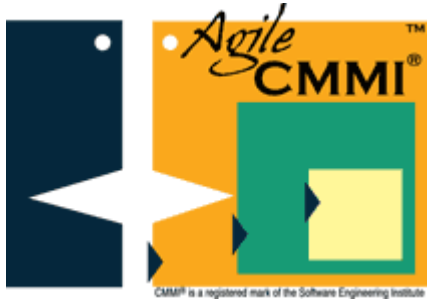
By

Tim Korson

tim@qualsys.org

Driving an Agile Peg in a CMMI Hole

Quasi-Agile Context

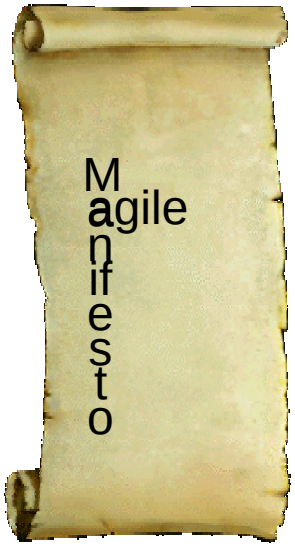


- Most corporations are still fairly traditionally structured
- Many software development teams are heading full steam into modern agile development techniques.
- This leaves management stuck coping with an organizational and technical paradigm shift that traditional project management practices are inadequate to handle.
- In the highly iterative, fast paced, environment characteristic of these agile development projects, traditional approaches to budgeting, testing, quality assurance, requirements gathering, scheduling and estimating, etc. break down.
- Managers trying to encourage best practices as recommended by CMMI and SPICE find themselves at odds with developers trying to adopt best practices as recommended by the agile manifesto.

Quasi Agile Context

- In the end no one wins.
- Because of the constraints of corporate policies and management edicts, developers can't fully adopt agile practices.
- Because the developers do adopt as much of the agile process as they can get away with, team leads find that traditional approaches to management don't work.
- Such projects must succeed in what I call a quasi agile development environment.
- In my experience these quasi agile development environments characterize a large percentage of today's significant software projects. Lack of explicit understanding of this reality, and failure to actively adapt to it, is causing significant problems in many software development organizations.

Agile Manifesto



We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

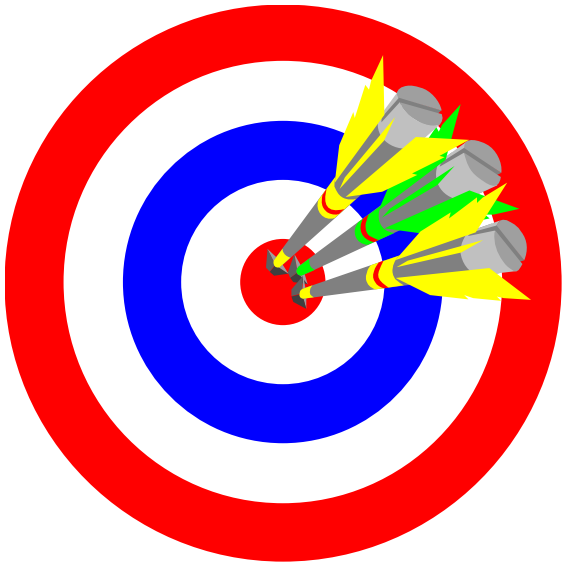
That is, while there is value in the items on the right, we value the items on the left more.

Quasi-Agile

- A quasi-agile environment is one where software developers are
  - implementing agile values and practices
  - within a traditionally structured organization that has policies and procedures derived from
    - waterfall concepts
    - typical CMMI or ISO philosophy

In Theory

- Both agile and plan driven approaches have the same goals
- The both seek to develop systems
  - Of high quality
  - As quickly as possible
  - With minimum cost
  - As predictably as possible



Better, Faster, Cheaper

In Practice

- Agile
  - Low overhead, minimal documentation
  - Aggressive iterative/incremental process
  - Creative optimizations flexibly tailored to the current environment and staff
- Plan Driven
  - Lots of process overhead and documentation
  - Waterfall like processes
  - Standardization across projects

Why

- Differences in philosophy about the nature of software development
  - Agile proponents believe that software development is very unlike manufacturing, building construction, or other activities that deal with physical construction materials.
  - CMMI has the concept of “institutionalization” at the heart of its philosophy
- Differences in philosophy about the goal of Software development
  - Meet requirements vs. solve business problems

Resulting Differences

Traditional	Agile
Plan entire project upfront	Scope upfront, details increment by increment
Lock in resources, details as early as possible	Lock in resources, details as late as reasonable
All requirements up front	Goals up front, details increment by increment
Resist change, control it, make it difficult	Facilitate change
Use data from past projects to plan current project	Use data from increment 1 to plan increment 2
Document anything that might possibly be useful	Only document things proven to be necessary
Put a process in place to prevent any potential error that could be prevented	Only put processes in place proven to be necessary
Rely on documents, processes, tools, plans	Rely on face to face collaboration

How do you drive an agile peg in a CMMI hole

1. Barely sufficient processes and documentation
2. Multi-cultural tolerance
   - Peaceful co-existence of processes derived from very different philosophies
3. Focus on the goal of traditional control structures
   - Show how the goal can be met in innovative agile ways

Barely sufficient processes and documentation

- You have to create a test plan, and the corporate template is 20 pages long
  - Use brief sentences, and lots of N/As
- Corporate processes require the use of a defect tracking tool to document defects and their resolution
  - Use face to face collaboration for the bulk of the tester/programmer interaction, but document selected results in the tool.

Multi-cultural tolerance

- Peaceful co-existence of processes derived from very different philosophies
- Recognition of the legitimacy of multiple interlocking processes
  - Management
  - Development
  - Testing
  - Deployment
  - Support

Multi-cultural tolerance Example

- The contract process might require a complete set of requirements up front
- The agile development process does not need or want all the details up front
- The quasi-agile solution requires peaceful coexistence of a traditional management process with an agile development process
  - Make sure the business objectives of the project are clear
  - Go ahead and let the manager get the requirements up front. Just make sure you don't believe them. Revisit them increment by increment and expect a lot of change.

Focus on the goal

- Show how the goal can be met in innovative agile ways
  - Documentation
    - Test cases
  - Process Improvement
    - Iteration retrospectives
  - Gated processes, “design complete”
    - Real goal is that the project manager has enough data to give more reliable estimates of cost, timeframe, and risk that at project inception.

Focus on the goal

- Show how the goal can be met in innovative agile ways
  - Inspections

Process Areas

- Agile methods do address most of the CMM process areas

Category	Process Area	XP	Scrum
Process Management	Organizational Process Definition	+	+
	Organizational Process Focus	–	–
	Organizational Training	+	+
	Organizational Process Performance	–	–
	Organizational Innovation and Deployment	–	–
Project Management	Project Planning	++	++
	Project Monitoring and Control	++	++
	Supplier Agreement Management	–	–
	Integrated Project Management	++	++
	Risk Management	–	+
	Integrated Training	++	++
	Quantitative Project Management	+	+
Engineering	Requirements Management	++	++
	Requirements Development	+	+
	Technical Solution	+	–
	Product Integration	++	–
	Verification	++	–
	Validation	++	+
Support	Configuration Management	++	–
	Process and Product Quality Management	+	–
	Measurement and Analysis	+	++
	Decision Analysis and Resolution	–	–
	Organizational Environment for Integration	++	++
	Causal Analysis and Resolution	–	–

CMMI - How versus What

- The CMM doesn't state how you have to do things, it only states what an assessor must look for
- CMMI has goals like “There's a process in place so everybody understands the requirements.” If you could show an assessor that instead of a requirements list, the user and the developer actually have a conversation, and you could easily get some assessors, not all, by any means, to say that's fine

Reality

- It is the cultures represented that are incompatible.
- Technically CMMI is compatible with Agile practices
- Integrating the two gives a Quasi-Agile Development Environment

Institutionalization

- When considering process documentation, the element that is missing from agile methodologies, which is crucial for the SW-CMM, is the concept of *institutionalization*, i.e., establishing the culture that "this is the way we do things around here."
- The key process areas in the CMM are structured by common features that deal with implementing and institutionalizing processes. The institutionalization practices for each key process area map to all the goals within the area, so a naive agile implementation that ignored these cultural issues would fail to satisfy any CMM key process area.

Fundamental Flaw

- There's a fundamental assumption in the CMMI that processes can be repeatable, and that they are predictive processes, basically not empirical processes, says Michael Spayd, a principal with Cogility Consulting Solutions and a former CMM process assessor. That is the fundamental flaw in the CMMI, and that's actually why I don't personally believe in level 4 and 5. They are ridiculous and do not create value

Deep, Fundamental Divide

- Cannot leave two distinct cultures at war with each other
- Must forge a Local Manifesto based on shared goals, not practices
- You will not get agreement on best practices unless you first get an agreement on shared goals and corporate axioms
- The best practices will evolve over time

Cowboy Programmer?

- Despite the outward appearances, Agile development does not mean a return to the days of the cowboy programmer. These techniques are discipline-oriented just as the practices described in CMMs. They do, however, have a different philosophy and approach to achieving the goals of the CMMI and can be integrated without compromising the framework.

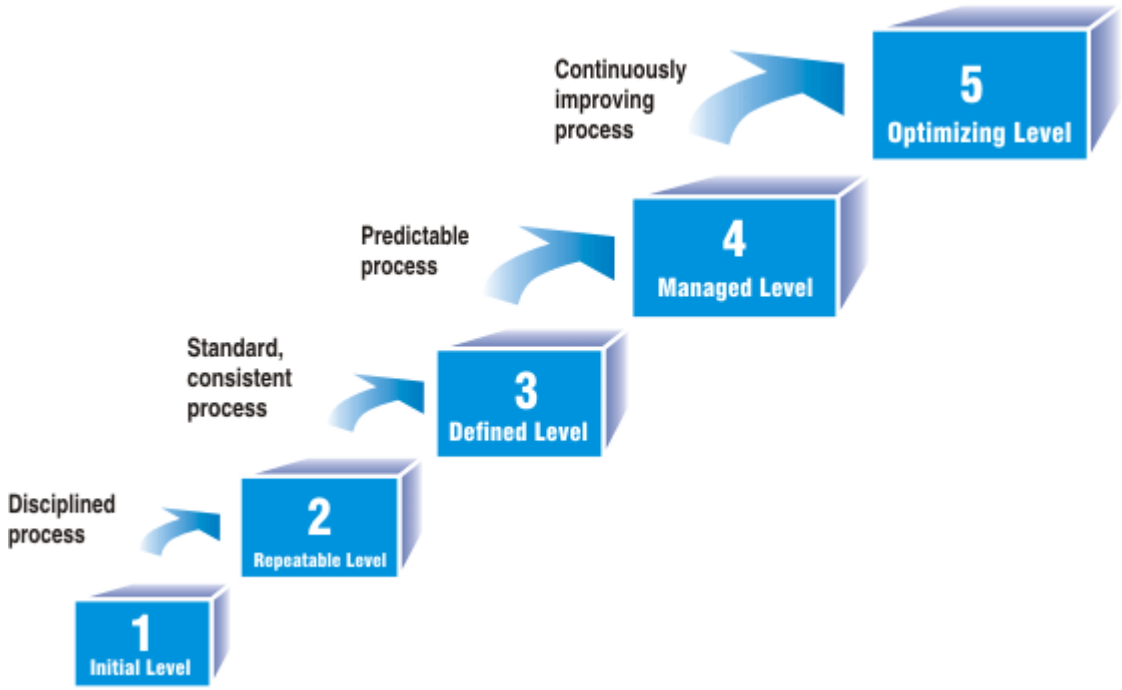
David Kane

Rigid, Bureaucratic vs. Immature, Undisciplined

- While an entrepreneurial Silicon Valley company has one culture, a space shuttle avionics team has another.
- Rather than recognizing the inherent differences between people, project teams, and organizations, we denigrate those who have different cultures by labeling them unprofessional, immature, or undisciplined. Or conversely, we label them bureaucratic, rigid, and closed-minded.

A mismatch of culture to methodology will hurt the project

My Observations



Real Process Improvement

- The cost of a process step must always be weighed against the cost of the implementation and the risk it mitigates

How do you drive an agile peg in a CMMI hole

1. Barely sufficient processes and documentation
2. Multi-cultural tolerance
   - Peaceful co-existence of processes derived from very different philosophies
3. Focus on the goal of traditional control structures
   - Show how the goal can be met in innovative ways

Thanks for coming

- On behalf of QualSys Solutions, thanks for attending this Webinar. We'd love to see you in future courses.
- Let us know about your agile experiences. We'd like to hear about your successes and your difficulties.
- My e-mail address is:

  tim@qualsys.org