How Evolution of Project Management Has Redefined Team Roles

June 5, 2015
Al Cline, PMP, PMI-ACP

from “The Agile Project Manager in the Real World”,
A. Cline, ASME Press, 2015

About the Presenter

• President of Carolla Development, Consulting Agile coach, PM and international trainer
• Agile practitioner since 1999; Concurrent Engineering (Kaizan) before that
• Lecturer at OSU for 9 years; Director of Nationwide’s Requirements COE for 2 years
• APM for $3.1M Air Force agile early-adopter project to fit agile to traditional environment
• Can be reached at acline@carolla.com, or www.carolla.com

© Alan Cline, 2015. All Rights Reserved
Evolution of Project Management

1. Ancient Project Management
2. Formal Development of Project Management
3. Project Management for the 21st century
4. Evolution of Agile Development
5. Software Roles in the New World
6. Comparing Practices: Agile vs. Traditional
7. Comparing Results: Agile vs. Traditional

1. Ancient Project Management

- Projects created for centuries
- Success was personality-dependent
- No body of knowledge (science) until 97 A.D.
  (Roman Fortinus, aqueduct city administrator)
- Projects used craftman model (master-apprentice relationship)
Ancient Project Management
Persian vs. Phoenician Canal Digging

2. Formal Development of PM
   - Origins -

1960’s:
   • DoD for managing risk and cost control
   • SEI as watchdog
   • PMI for professional certification
   • Origin of PM-1 (predictive life cycle)
Formal Development of PM
- Software Development as a Manufacturing Metaphor -

1970-80’s:
- Concurrent Engineering (Kaizen)
- Assembly line process steps
  + Very short feedback cycles
  - People are not repeatable workstations
  - Software is unique, not duplicate product units

Formal Development of PM
- Moving Toward Software Engineering -

1990’s:
- More quantitatively controlled science of development
- System view approach (input/process/output)
- System view feedback is non-linear (unpredictable)
  - 84% fell short in budget, schedule, performance, or customer satisfaction (16% full project failure)
3. Project Management for the 21st Century
   - Evolutionary Levels -

PM-1 – predictive life cycle, includes waterfall
PM-2 evolved from PM-1 by explicit inclusion of:
   • Social factors (stakeholder expectation management)
   • Psychological factors (team dynamics)
   • Anthropological factors (organizational culture)
   • Integration of these (political forces)

PM-3 and PM-4 are future evolutionary levels

Project Management for the 21st Century
   - Structural Regions -

Influencing Regions defined for PM-1 and PM-2:
   • Region 1: Technical
     o Project team dynamics
     o Self-empowered, self-organizing teams
     o Incremental & iterative development (agile)
   • Region 2: Organizational
     o Cultural & infrastructure influences
     o Project portfolio management
   • Region 3: Institutional (policy generation)
Project Management for the 21st Century
- Modern PM Structure -

© Alan Cline, 2015. All Rights Reserved

Project Management for the 21st Century
- Cost of Change Problem-

© Alan Cline, 2015. All Rights Reserved
4. Evolution of Agile Development
- Key Approaches -

Extreme Programming (XP; Beck)
- Attempt to flatten the cost-of-change curve
- Succeeded for small teams
- Size of project trumps methodology
- Results: Higher quality and more adaptive with customers

Scrum (Empirical Process; Schwaber & Sutherland)
- Shorten release cycles
- Predictive process doesn’t work
- Changes are hard to incorporate
- Quality deteriorates with time pressures
- Death marches hurt morale

Acid test: If you are not delivering software every 30 days, you are not doing agile!
Manifesto for Agile Software Development

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.

Evolution of Agile Development

- **All Approaches** -
  - Must complies with Agile Manifesto, but tailored
  - Incremental or iterative development or both
  - Historical data (e.g. velocity) drives predictable planning
  - Requirements analysis, design, coding, testing within each 2-4 week iteration
  - Easy and quick changes; informal change mgmt
  - Refactoring: continual technical improvement
  - Automated testing
  - Self-empowered, self-organized teams
  - Production-quality software (partial product) at end of each iteration
  - User demo for stakeholders & customer feedback every 1-3 iterations
  - **Software ready for release within 30 days.**
Applying Agile Iterations
- Progressive Elaboration for Traditional Project -

Applying Agile Iterations
- Iterative Development for XP Project -

Extreme Programming (XP) Approach with Repeated Iterations

© Alan Cline, 2015. All Rights Reserved
Agile is a pre-configured subset of PM-2.

The agile operating mode is very similar to the previously described “evolutionary acquisition model” of DoD. Ultimately, it is also an application of the cyclic evolution process “variation-selection-keeping.” Therefore, the agile project management corresponds to the principles of PM-2, a precise cooperation of World 1 and World 2. But the phenomenon of the evolutionary overlapping of traditional methods... emerges at this point, because the evolutionary elements are realized unconsciously.”

(Saynisch, p15; emphasis added)
Agile in the Industry

- PMI has created PMI-ACP for agile-certified PMs
- Multiple Agile certifications now available
- Software Engineering Institute (SEI) incorporates agile practices into their CMMi Level 2 and Level 3 maturity framework
- IIBA incorporates some agile practices into CBAP (BA professional certification), and even extends slightly

5. Software Roles in the New World
- What the Agile books say -

- Scrum: Product Owner, Scrum Master, Team Member
- XP: Customer, Developer, Coach/Tracker
- Traditional PM’s and BA’s are not mentioned
- Testing is done within the iteration, but testing role is not mentioned, or...
- “Developers” do everything
Software Roles in the New World
- Region 2 -

- Technical team not yet available
- Sponsor: Establish project, charter, funding
- PM: Traditional, establish team, manage stakeholder expectations, monitor project progress
- BA: Traditional, business workflow analysis, prioritize feature catalog with customers; support PM with stakeholder expectations

Software Roles in the New World
- Iteration 0 -

- PM: Business & technical team kickoffs; help establish infrastructure
- BA: Refine product backlog w/customers for large-grain requirements (features to epics, themes, or use cases)
- Technical team: Define product architecture
- All: Build Release Plan with estimated-scope iterations
Software Roles in the New World
- Region 1: All iterations -

- **PM:** Move to APM role (coach), stakeholder reporting, servant leader, facilitates self-empowered team
- **BA:** SME for requirements, help UX design, help team build iteration backlog (epics, themes, or use cases to user stories); liaison for change mgmt
- **Team:** Builds and commits to iteration backlog, responsible for producing working software at end of iteration, user demo
- **Developer:** Designs, codes, unit tests, repairs defects; calculates change impact
- **Tester:** Builds integration tests concurrently with coding; facilitates defect and changes

6. Comparing Practices: Agile vs. Traditional

**Traditional**
- *Big up-front requirements and design.* Voluminous specs and design before construction. Detailed WBS for entire (multi-year?) project
- *Overly-precise and under-accurate specifications.* Change is inevitable, so high-effort plan obsolete very soon; quality drops as mtc is high and TLDR
- *Cost of change exponentially high.* Boehm’s curve; sometimes mgmt forced compliance due to cost; common budget and schedule overruns
- *Lack of sufficient customer interaction.* Reqmts “tossed over the wall”, product tossed back; requesters could be gone; installed without checking customer needs
- *Technology diversification and redundancy.* BU workarounds; private programmers; lost economies of scale and competence

© Alan Cline, 2015. All Rights Reserved
6. Comparing Practices: Agile vs. Traditional

Agile

- **Big up-front requirements and design.** Adaptive planning: the more imminent the implementation, the more detailed the estimates
- **Overly-precise and under-accurate specifications.** Reqmts only as detailed and timely as needed. Mona Lisa example
- **Cost of change exponentially high.** Small partial product for smaller changes, better control
- **Lack of sufficient customer interaction.** Frequent interaction, partial product periodically reviewed, change is minimized, needs better met
- **Technology diversification and redundancy.** Agile teams are small teams, can be in BU or IT, proficiency and productivity rises

7. Comparing Results: Agile vs. Traditional

- Agile projects almost always complete on budget and on time, and most have zero defects when released (Standish, 2013; Forrester, 2013).
- [The studies] cited an average of 29% improvement in cost, 71% improvement in schedule, and 122% improvement in productivity performance. Quality improvement averaged 75% and customer satisfaction improvement averaged 70%. Over 29 of these studies had the data necessary to estimate the average return on investment of 2633%. (Rico, Sayani, and Sone, p96)
- Gartner predicts that in the next couple of years “agile development” methods will be utilized in 80% of all software development projects. (Agile Transformations, p43)
Comparing Results: Agile vs. Traditional
- PM Network magazine -

- Since 1994, [Standish Group] for 10,000 projects around the world, showed that only 37% of projects succeeded, i.e., came in on time and budget (32% in 2008, 28% in 2004). However, “the 2011 results represent the highest success rate in the history of [those reports]”. (PM Network magazine, Gale, 2011, p10-11)

- Reasons:
  - Partly to economic market recovery
  - Partly to the new way that organizations are approaching project management (installing PMOs)
  - Integrating fast-reacting, adaptive project management into their organizations, and targeting projects to fit agile practices

Last Thoughts on Agile

- Has great ideas to improve software development
- Moves PM to a higher level of evolution (PM-2)
- Lots of myth (no schedules, budgets, planning)
- Lots of dogma (engenders “religious wars”)
- Lots of hype (irrelevant practices)

See Bertrand Meyer’s

“Agile! The Good, the Hype, and the Ugly”
Last Thoughts on Planning

In 480 B.C., King Xerxes and his advisor General Artabanus had a parting of the ways, differing on many points of view.

General Artabanus: *the best man, in my belief, is he who lays his plans warily, with an eye for every disaster which might occur, and then, when the time comes, act boldly.*

King Xerxes: *Certainty, surety, is beyond human grasp. But however that may be, the usual thing is that profit comes to those who are willing to act, not to the overcautious and hesitant.*

What is your view of planning?

© Alan Cline, 2015. All Rights Reserved

Bibliography


© Alan Cline, 2015. All Rights Reserved
Bibliography
- continued -


Bibliography
- continued -


Bibliography
- continued -


© Alan Cline, 2015. All Rights Reserved
QUESTIONS?
acline@carolla.com