Promises & Pitfalls of Agile Methods

14 Promises of Agile Methods

- Value – Delivers highest-priority customer capabilities, features, requirements, and needs.
- Risk – Reduces project scope, requirements, size, complexity, and risk.
- Discipline – Fast, flexible, and cost-effective, yet highly disciplined planning and delivery method.
- Efficient – Small strategy, portfolio, planning, process, work in process, batch, queue, and team size.
- Feedback – Uses planned and unplanned daily, bi-weekly, and release feedback cycles.
- WIP Constraints – Uses portfolio, capability, feature, user story, and iteration size constraints.
- Teamwork – Small, high-performing, fast, and cost-efficient cross-functional, multi-disciplinary teams.
- Requirements – Uses collaboration and rapid feedback to elicit hidden, inexpressible user needs.
- Architecture – Uses lean, just-enough, just-in-time, and high-performing architectures and designs.
- Design – High-performing, loosely-coupled functional slices validated and delivered one-at-a-time.
- Flexibility – Fast, inexpensive, and abstractive workflow, development, and delivery technologies.
- Quality – Automated verification, validation, configuration mgt., documentation, and deployment.
- Complete – Combines of state-of-the-art business, lean, and technical principles and practices.
- Improvement – Built-in daily, bi-weekly, and release process improvement cycles.

14 Pitfalls of Agile Methods

- Change – Use of top-down, big-bang organization change, adoption, and institutionalization.
- Culture – Agile concepts, practices, and terminology collide with well-entrenched traditional methods.
- Acquisition – Using traditional, fixed-price contracting for large agile delivery contracts and projects.
- Misuse – Scaling up to extremely complex large-scale projects instead of reducing scope and size.
- Organization – Unwillingness to integrate and dissolve testing/QA functional silos and departments.
- Training – Inadequate, insufficient, or non-existent agile training (and availability of agile coaches).
- Infrastructure – Inadequate management and development tools, technologies, and environment.
- Interfacing – Integration with portfolio, architecture, test, quality, security, and usability functions.
- Planning – Inconsistency, ambiguity, and non-standardization of release and iteration planning.
- Trust – Micromanagement, territorialism, and conflict between project managers and developers.
- Teamwork – Inadequate conflict management policies, guidelines, processes, and practices.
- Implementation – Inadequate testing to meet iteration time-box constraints vs. quality objectives.
- Quality - Inconsistent use of agile testing, usability, security, and other cost-effective quality practices.
- Experience - Inadequate skills and experience (or not using subject matter experts and coaches).

(***Note***. *Firms may prematurely "revert" to inexorably slower and more expensive <u>traditional methods</u> or "leap" onto <u>lean methods</u> that may not adequately address common pitfalls of adopting agile methods.*)