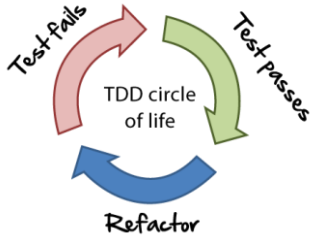
BITS Pilani
Pilani | Dubai | Goa | Hyderabad

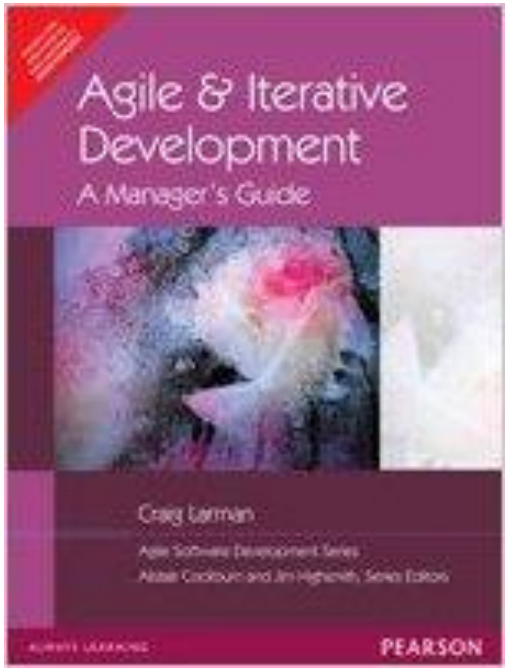


Quality Management in Agile

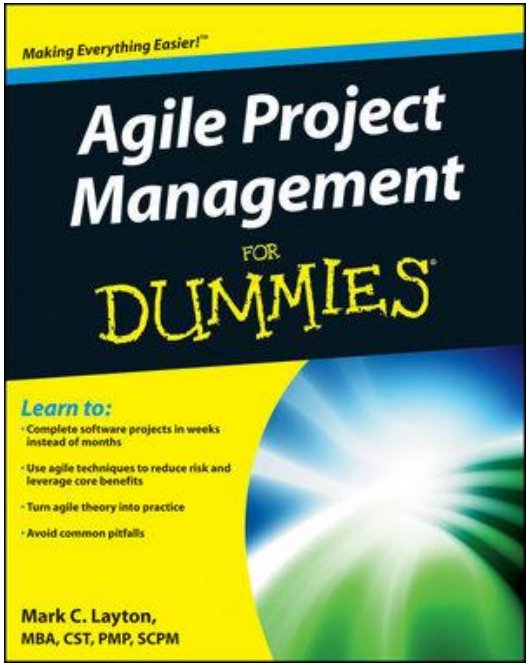
- Prof K G Krishna

Text/Reference Books

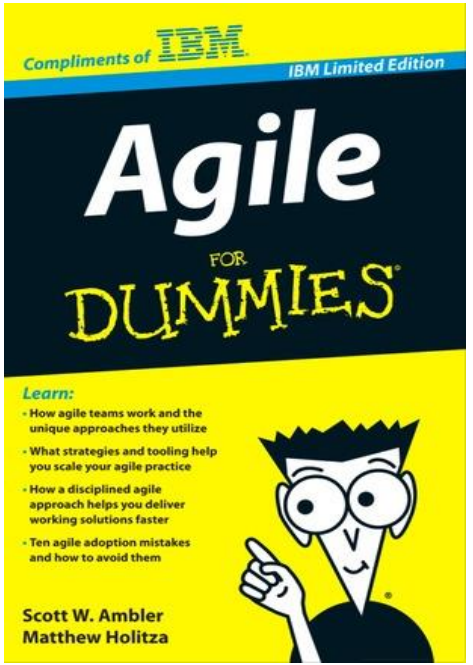
T1



T2



Compliments of IBM



➔ As this field is evolutionary, the student is advised to stay tuned to the current and emerging practices by referring to their own organization's documentation as well as Net sources

Topics

Quality Management in Agile

- Managing Quality in Agile
- Integrated Testing in Agile
- Managing Risks in Agile

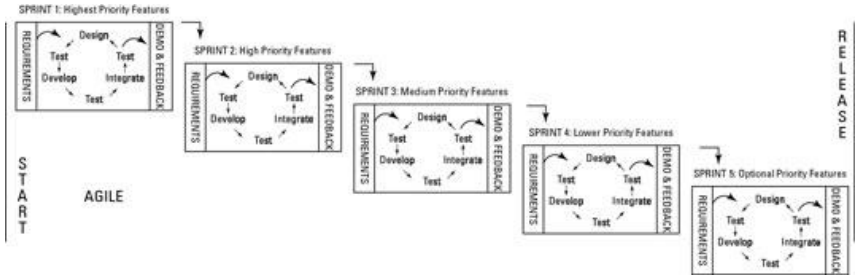
Ensuring Quality: Traditional vs. Agile

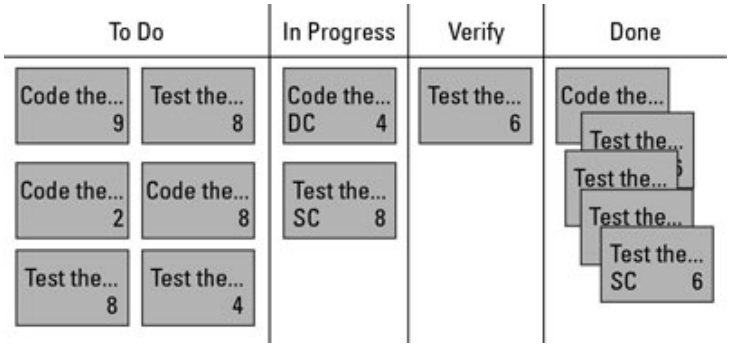
- Testing is the Last Phase of the Project vs. Daily Activity as part of Sprint
- Reliance of Manual Testing (unit) vs. Automated Test Tools (necessary)
- Reactive (fixing discovered bugs/issues) vs. Focus on Proactive practices (Pair Programming, Coding Standards, and Face-to-Face Communication)
- Risk is more when Problems surface at the end vs. Risk is avoided by addressing it in early Sprints
- Bugs are hard to discover at the end of Project vs. Finding and Fixing Bugs in small iterations of little code is easy
- Testing gets compromised as Project reaches deadlines vs. Testing is continuous activity and is integrated into daily development (Continuous Testing)

Ref: (T2-Chap 15)

Quality is Integrated into Agile Methods

1. **Test Continuously**: Testing begins in the first sprint, and the scrum team evaluates quality throughout each sprint, finding and fixing any problems immediately.
2. **Proactive Prevention**: Building Acceptance criteria into User Stories
3. **Inspecting Regularly and Adapting as Needed**: Product (during Sprint Review) and Process (during Sprint Retrospective) are reviewed and adapted as per the need
4. **Use of Automated Test-tools** help embed Testing in Daily activities and in every Sprint

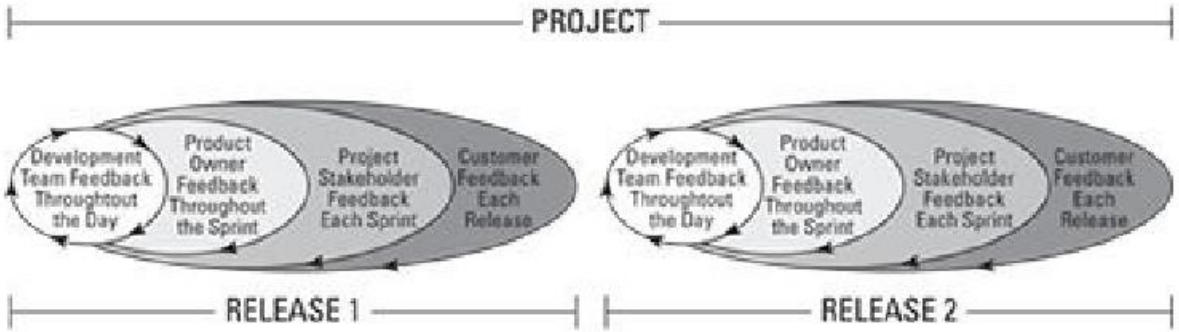




Ref: (T2-Chap 15)

Multiple *Feedback Loops* Feeding Quality

- Frequent (~Daily) Feedback from Development Team, Product Owner, Stakeholders and Customer (as part of Daily Scrum, Sprint Review, Release Planning)
- Involvement of Cross-functional Teams, Domain/Technical Experts
- Continuous Inspecting and Adapting ensures Quality
- Easier to Find Bugs in Short Code (of one Iteration) rather than Large Code (end of Project)



Ref: (T2-Chap 15)

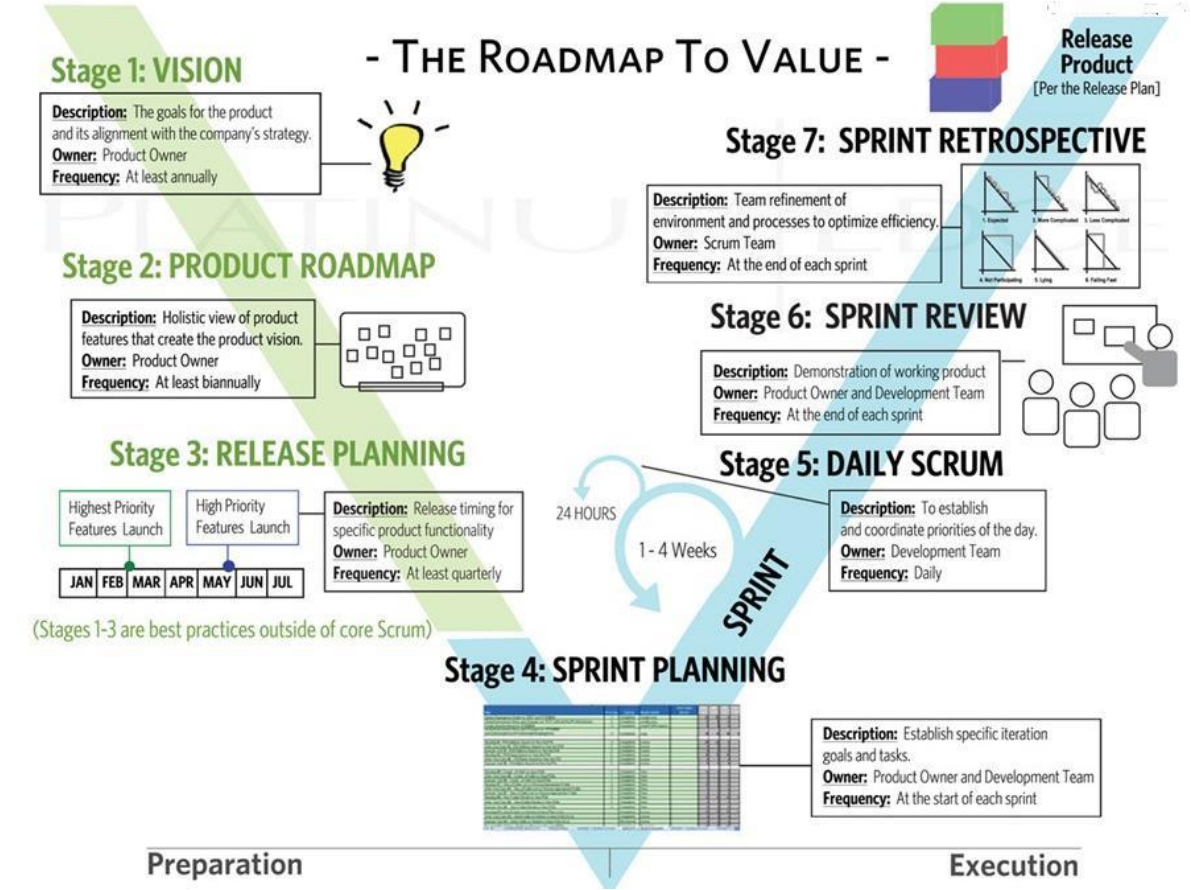
Agile Development Methods emphasize Quality

- **Test-Driven Development (TDD)**: Evolved from XP Practices, TDD focuses on creating Tests for the Requirement before coding, then ‘failing’ the Test, then ‘develops’ code to fulfil the test and then refactors the code by taking as much code as possible while the test passes
- **Pair-programming**: XP Practice in which Developers work in pairs—both developers work at the same desk and take turns of development and testing for the same Requirement; ensures quality by constant over-the-shoulder review by the peer by providing instant error checks-and-balances
- **Peer-reviews**: Reviewing each others’ code they are collaborative in nature by allowing peer-level experts in other groups to help ensure objective review
- **Collective Code Ownership**: Key feature of Agile where any Team member can create, fix or change any part of the code on the project; speeds up development, fosters innovation and help find bugs quickly
- **Continuous Integration**: Daily builds helps check how the current story built works with the rest of the product; allows resolve conflicts early on; daily code builds are necessary for running automated test-suites later in the night

Ref: (T2-Chap 15)

Managing Risks in Agile ➔➔

Every Meeting in the SCRUM Execution is a Risk Mitigation Meeting!



Source: (T2)

Managing Risk in Agile Projects

- Risk Management is integral to Agile – it doesn't have to involve formal Risk Documentation and Meetings. Risk management is embedded in these below Agile Principles (part of 12 Agile Principles Manifesto):
  - ❖ Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
  - ❖ Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
  - ❖ Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
  - ❖ Business people and developers must work together daily throughout the project.
  - ❖ Working software is the primary measure of progress.

➔ "Failing-fast" early subsumes Risk in Agile Projects by prioritizing High-value and High-risk Requirements during early Sprints

Ref: (T2-Chap 15)

Managing Risk: Traditional vs. Agile

- Large Projects are challenged with Overruns and Scope-creep and nothing-to-show till the end vs. Catastrophic failures are eliminated with 'something-to-show' at every stage
- Conducting Testing (ST/UAT) at the end means finding serious problems pose grave risk to the project vs. Continuous testing which surfaces problems early and if required project can be abandoned early on without significant upfront investment
- Requirements change in later phases unacceptable vs. Requirements change can be welcome at any stage by dynamically adjusting priorities
- Inaccurate estimates of Cost/time made at the start when we know least information about the Project vs. No upfront estimates – only running estimates/restimates using Scrum Team's actual performance or Velocity
- Multiple Stakeholders having diverse view of the Product may end up arriving at conflicting Requirements vs. Single Product Owner responsible for end-to-end Product development from Visioning to Delivery
- Project Stakeholders may be unresponsive for issue resolution once the Project starts vs. Scrum Master with organizational clout dedicated to the Team to remove roadblocks
- Income generation/revenues from the Project only upon final delivery vs. Income generation may commences from the first release (or after first Sprint); a Self-funded project that generates additional revenue with every release has a good chance of continuing during a crisis

Ref: (T2-Chap 15)

.

Risk Management Artifacts/Meetings in Agile

- Product Vision
- Product Roadmap
- Product-backlog
- Release Planning
- Sprint Planning
- Sprint-backlog
- Daily Scrum
- Task-board
- Sprint Review
- Sprint Retrospective

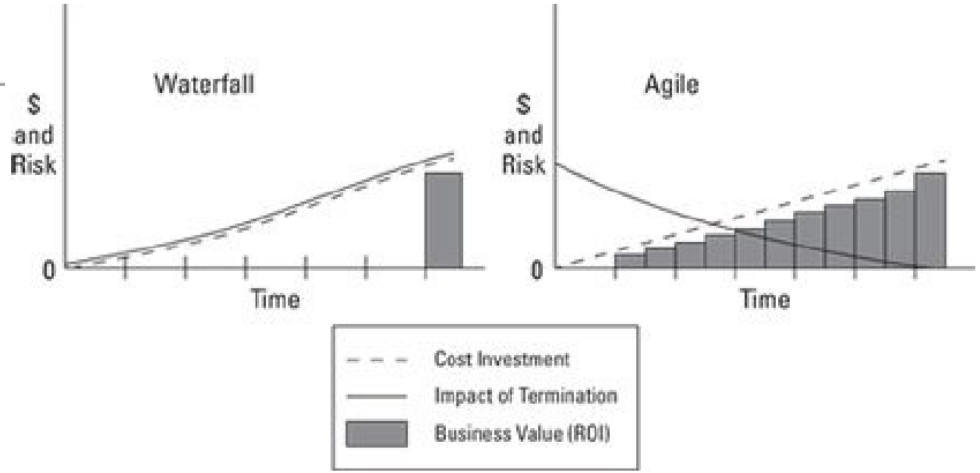
Ref: (T2-Chap 15)

Risk Mitigation is Inherent in Agile

- Short Development Cycles (Sprints) and Daily Scrum Meetings trap all potential risks early on in the Project and take decisions by adapting Product-backlog and Scrum Process if necessary
- The definition of "*Done*: Developed, Tested, Integrated and Documented" reduces risk factor by creating a Product that meetings this definition in *every* Sprint

Definition of Done

Sprint	Release	Risks Accepted
QA Environment Unit Tested Functional Tested Integration Tested User Acceptance Tested Regression Tested XDocs	Staging Environment Performance Tested Security Tested Enterprise System Integrated Focus Group Tested User Documentation Training Documentation	Load Testing



Ref: (T2-Chap 15)

Summary: Quality and Risk Management in Agile

- Quality and Risk Management are embedded in the Agile Process itself
- Testing is integrated into Development (‘*Done*’ definition includes all Testing activities for every intermediate working product of Scrum Team)
- Quality and Risks are reviewed and acted upon in every Meeting (Daily Scrum, Sprint Review and Sprint Retrospective)
- Product Owner and Scrum Master are empowered to deal with Risks of Quality or Schedule/Cost overruns as and when they surface
- Ensures Transparency and Open Discussion of issues of Quality/Risk through highly visible Task-boards and Face-to-face Communication
- Pair-programming ensures continuous Peer-review by working in pairs of Developer-Tester/Reviewer with one overlooking the other
- The inherent Structure of Scrum—Product Owner, Scrum Master and Collective Ownership of the Product along with involvement of Customer and Key Stakeholders—safeguards against many of the impediments to Quality or Risks in the Project

Thank You