

PMI® AGILE CERTIFIED PRACTITIONER

EXCEL WITH EASE

THIRD EDITION

S. CHANDRAMOULI
SAIKAT DUTT



Pearson

FEATURES

- ▶ Covers all topics based on new PMI-ACP® syllabus
- ▶ 1000+ practical questions
- ▶ Four fully solved question papers following the PMI-ACP® exam pattern
- ▶ 400 key points that encapsulate important concepts
- ▶ Tips and tricks on “cracking” the PMI-ACP® certification exam

PMI AGILE CERTIFIED PRACTITIONER—EXCEL WITH EASE

THIRD EDITION

S. Chandramouli
PMP, PMI-ACP
Associate Director
Cognizant Technology Solutions

Saikat Dutt
PMP, PMI-ACP, CSM
Director
Cognizant Technology Solutions



My sincerest thanks and appreciation go to several people...

My wife Ramya

My son Shri Krishna

My daughter Shri Siva Ranjani

My parents Subramaniam and Lalitha

And my colleagues and friends

Chandramouli

*This book is dedicated to the people without whom the dream
couldn't*

*have come true—My parents Tarun Kumar Dutt and Srilekha
Dutt,*

my wife Adity and my sons Deepro and Devarko

Saikat Dutt

Contents

About the Authors

Foreword

New to the Third Edition

Preface

1 Introduction

Contents of the Exam

Tools and Techniques

Agile Knowledge and Skills

Chapter-wise Coverage of Exam Content

Eight Steps for PMI-ACP® Exam Preparation

Tips and Tricks for PMI-ACP® Exam

2 Introduction to Agile Principles and Mindset

Introduction

What is Agile?

Agile Manifesto Values and Principles

Agile Methodologies

Agile Principles and PMI-ACP Domains

Agile Frameworks and Terminology

Agile vs Plan-Driven Management

Agile Project Life Cycle

3 Agile Methodologies

Introduction

Variance in Agile Methods and Approaches

Applying New Agile Practices

Scrum

Extreme Programming (XP)

Test-Driven Development (TDD)

Feature-Driven Development (FDD)

Kanban Method

Agile Unified ...

About the Authors



S. Chandramouli, PMP, PMI-ACP, is an alumnus of the Indian Institute of Management, Kozhikode (IIM-K), and a prolific writer of business management articles dealing with delivery management, Program Management, competitiveness, IT, organizational culture and leadership. He has experience in Software sales management, client relationship management, competency management, people management, financial management and quality management. He has strong experience in Project/Program/Account delivery and escalation resolution- providing executive oversight and

monitoring the overall portfolio and program health.

He is certified “Green Belt” in six sigma methodology. He is also ITIL(F) Certified. He Conducts training programs on variety of subjects like Project Management, Program management, software engineering, quality system. He has been an invited speaker at various management conferences.

He is also the author of the book titled PMP Certification—Excel with Ease published by Pearson, which has been recognized as a reference book in various universities across the globe. Second edition of this book got released based on the latest PMBOK(R) 5th Edition.



Saikat Dutt, PMP, PMI-ACP, CSM, has 19 years of IT industry experience managing large-scale multi-location and mission-critical projects. He is a project management professional (PMP) and PMI Agile Certified Practitioner (PMI-

ACP), certified by the Project Management Institute (PMI) in the United States. He is also a certified Scrum master. He is involved in the coaching of large-scale Scrum implementations in a big MNC in the IT sector. His other two books—Software Engineering (ISBN: 9789332537293) and Software Project Management (ISBN: 9789332542143) are also best sellers and used as text books in reputed universities and business schools. He is also an active speaker on Agile best practices and project management methodologies in several forums.

Foreword

Agile Project Management is gaining increasing attention among executives, business leaders and IT management professionals. Thus, Agile Certification from PMI is becoming more and more important for the IT project manager to have in his toolkit. I look at this book in two ways. First, this is an excellent guide to put you on the solid path to becoming a PMI Agile Certified Practitioner (PMI-ACP). Second, even if you have mature Agile process capabilities in your organization, this is a great reference book to identify areas for improvement. This book is a serious compilation of Agile Project Management ideas, techniques, and disciplines and is written for professionals who have a passion and love for Agile Project Management.

I admire ...

New to the Third Edition

This third edition of the book, in keeping with the trend set by the earlier two editions, continues to focus on real-life situations and endeavours to hone the skills of those who aspire to crack the Project Management Institute (PMI) certification examination.

The first and second editions of the book were designed in accordance with the syllabus of the first version of the exam. With PMI rolling out its revised examination pattern in 2015 with changed syllabus and question structure, it necessitated a modification in the book's contents to suit the new curriculum and exam pattern. The current edition captures the defining features of the changed question pattern and maps the book's contents directly to the PMI ACP syllabus topics for the reader's benefit.

In addition, the contents of the book's second edition have been enhanced on the following

fronts:

- Chapter coverage: Chapter 1—*Introduction* gives a direct mapping of all Knowledge and Skills and Tools and Techniques mentioned in the PMI ACP course content. The details of PMI ACP exam process starting from preparation to filling up the form and guidance for requirements of making payment are provided in this chapter.
- Added topics: All topics newly added to the syllabus are added in relevant chapters. The Q&A section at the end of each chapter is augmented with new questions on these topics. In-depth discussions have been included on topics like XP, Lean, VSM, user stories, relative sizing, requirement management and estimation topics to match with the depth of knowledge now expected in the exam.
- More diagrams: A picture speaks a thousand words. To make complex concepts clearer, more diagrams are added in the second chapter. These diagrams will help the reader relate to the actual scenarios while going through the concepts.
- There are four full-length question papers at the end of this book which would come in handy at the time of preparing for the exam. All question papers have been modified to match the actual question patterns and topic coverage to ensure that readers are not caught on the wrong foot when taking the real exams.

It is our continuous endeavour to keep ourselves abreast of the changes happening in the field of Agile. This is a very dynamic area, with new concepts evolving every day. We hope that this third edition is helpful to all our readers.

S. Chandramouli and Saikat Dutt

Preface

Why Take the PMI-ACP Exam?

The PMI-Agile Certified Practitioner (PMI-ACP) exam focuses more on real-life situations and hence when you are preparing for the exam by practising real-time scenario, the confidence level automatically increases—there is no magic in it. Certified people are paid at least 10% more than non-certified people. I have come across many situations where professionals are getting higher salaries just after acquiring PMI certification. I wish the readers of this book to get the benefits of PMI-ACP certification.

About PMI-Agile Certification

The PMP® and PMI-ACP® certifications are the project management profession's most recognized credential from PMI. At present,

PMI's project management certifications are well-accepted ...

1

Introduction

KNOWLEDGE AND SKILLS LEVEL

PMI ACP Exam Content

- Tools and Techniques
- Knowledge and Skills
- Domains and Tasks

Chapter-Wise Coverage of Exam Content

The PMI Agile Certified Practitioner Exam is based on the Agile topics that are essential for the Agile professionals to master. For the purpose of the exam and also to grasp the

concepts of the Agile framework, it is necessary to know the exam structure and contents very well. In this chapter, we will discuss the exam structure, content, and how this book has covered all the topics suggested by PMI.

CONTENTS OF THE EXAM

PMI Agile Certified Practitioner Exam is based on the Agile topics which are essential for the Agile professionals to master. for the purpose of the exam and also to grasp the concepts of the Agile framework it is necessary to know the exam structure and contents very well. In this chapter, we will discuss the Exam structure, content and how this book has covered all the topics suggested by PMI.

The web site of project management institute, www.pmi.org, contains the ‘PMI-Agile Certified Practitioner (PMI-ACP) Examination Content Outline’ which describes the topics covered in the exam.

The exam has **120 multiple-choice** questions. Only **100 questions** will be evaluated for scoring

purpose. The remaining **20 are pre-test** questions that will be randomly distributed in the question paper.

The exam gives equal importance to the Agile tools and techniques and the Agile knowledge and skills.

The exam content is distributed across seven domains. The tasks performed by an Agile practitioner are grouped under these domains. For the exam, it is very important to understand the domains, sub-domains and the tasks within each of them. The table below gives the indicative percentage of questions from each of these domains.

Domains and Tasks

<i>Domain</i>	<i>Percentage of Items on</i>
----------------------	--------------------------------------

Domain I. Agile Principles and Mindset	16 %
Domain II. Value-Driven Delivery	20 %
Domain III. Stakeholder Engagement	17 %
Domain IV. Team Performance	16 %
Domain V. Adaptive Planning	12 %
Domain VI. Problem Detection and Resolution	10 %
Domain VII. Continuous Improvement (Product, Process, and People)	9%

Domain I Agile Principles and Mindset

T a s k 1	Advocate for agile principles by modeling those principles and discussing agile values in order to develop a shared mindset across the team as well as between the customer and the team.
T a s k 2	Help ensure that everyone has a common understanding of the values and principles of agile and a common knowledge around the agile practices and terminology being used in order to work effectively.
T a s k 3	Support change at the system or organization level by educating the organization and influencing processes, behaviors, and people in order to make the organization more effective and efficient.
T a s k 4	Practice visualization by maintaining highly visible information radiators showing real progress and real team performance in order to enhance transparency and trust.
T a s k 5	Contribute to a safe and trustful team environment by allowing everyone to experiment and make mistakes so that each can learn and continuously improve the way he or she works.
T a s k 6	Enhance creativity by experimenting with new techniques and process ideas in order to discover more efficient and effective ways of working.

T a s k 7	Encourage team members to share knowledge by collaborating and working together in order to lower risks around knowledge silos and reduce bottlenecks.
T a s k 8	Encourage emergent leadership within the team by establishing a safe and respectful environment in which new approaches can be tried in order to make improvements and foster self-organization and empowerment.
T a s k 9	Practice servant leadership by supporting and encouraging others in their endeavors so that they can perform at their highest level and continue to improve.

Domain II Value-Driven Delivery

Define Positive Value

T a s k 1	Define deliverables by identifying units that can be produced incrementally in order to maximize their value to stakeholders while minimizing non-value-added work.
T a s k 2	Refine requirements by gaining consensus on the acceptance criteria for features on a just-in-time basis in order to deliver value.
T a s k 3	Select and tailor the team's process based on project and organizational characteristics as well as team experience in order to optimize value delivery.

Avoid Potential Downsides

T a s k 4	Plan for small releasable increments by organizing requirements into minimally marketable features/minimally viable products in order to allow for the early recognition and delivery of value.
T a s k 5	Limit increment size and increase review frequency with appropriate stakeholders in order to identify and respond to risks early on and at minimal cost.
T a	Solicit customer and user feedback by reviewing increments often in order to confirm and enhance business value.

s k 6	
Prioritization	
T a s k 7	Prioritize the units of work through collaboration with stakeholders in order to optimize the value of the deliverables.
T a s k 8	Perform frequent review and maintenance of the work results by prioritizing and maintaining internal quality in order to reduce the overall cost of incremental development.
T a s k 9	Continuously identify and prioritize the environmental, operational, and infrastructure factors in order to improve the quality and value of the deliverables.
Incremental Development	
T a s k 1 0	Conduct operational reviews and/or periodic checkpoints with stakeholders in order to obtain feedback and corrections to the work in progress and planned work.
T a s k 1 1	Balance development of deliverable units and risk reduction efforts by incorporating both value-producing and risk-reducing work into the backlog in order to maximize the total value proposition over time.

T a s k 1 2	Re-prioritize requirements periodically in order to reflect changes in the environment and stakeholder needs or preferences in order to maximize the value.
T a s k 1 3	Elicit and prioritize relevant non-functional requirements (such as operations and security) by considering the environment in which the solution will be used in order to minimize the probability of failure.
T a s k 1 4	Conduct frequent reviews of work products by performing inspections, reviews, and/or testing in order to identify and incorporate improvements into the overall process and product/service.

Domain III Stakeholder Engagement

Understand Stakeholder Needs

- | | |
|-----------------------|---|
| T
a
s
k
1 | Identify and engage effective and empowered business stakeholder(s) through periodic reviews in order to ensure that the team is knowledgeable about stakeholders' interests, needs, and expectations. |
| T
a
s
k
2 | Identify and engage all stakeholders (current and future) by promoting knowledge sharing early and throughout the project to ensure the unimpeded flow of information and value throughout the lifespan of the project. |

Ensure Stakeholder Involvement

- | | |
|-----------------------|--|
| T
a
s
k
3 | Establish stakeholder relationships by forming a working agreement among key stakeholders in order to promote participation and effective collaboration. |
|-----------------------|--|

- | | |
|-----------------------|--|
| T
a
s
k
4 | Maintain proper stakeholder involvement by continually assessing changes in the project and organization in order to ensure that new stakeholders are appropriately engaged. |
|-----------------------|--|

- | | |
|-----------------------|---|
| T
a
s
k
5 | Establish collaborative behaviors among the members of the organization by fostering group decision making and conflict resolution in order to improve decision quality and reduce the time required to make decisions. |
|-----------------------|---|

Manage Stakeholder Expectations

- | | |
|---|---|
| T | Establish a shared vision of the various project increments |
|---|---|

a s k 6	(products, deliverables, releases, and iterations) by developing a high-level vision and supporting objectives in order to align stakeholders' expectations and build trust.
T a s k 7	Establish and maintain a shared understanding of success criteria, deliverables, and acceptable trade-offs by facilitating awareness among stakeholders in order to align expectations and build trust.
T a s k 8	Provide transparency regarding work status by communicating team progress, work quality, impediments, and risks in order to help the primary stakeholders make informed decisions.
T a s k 9	Provide forecasts at a level of detail that balances the need for certainty and the benefits of adaptability in order to allow stakeholders to plan effectively.

Domain IV Team Performance

Team Formation

T a s k 1	Cooperate with the other team members to devise ground rules and internal processes in order to foster team coherence and strengthen team members' commitment to shared outcomes.
T a s k 2	Help create a team that has the interpersonal and technical skills needed to achieve all known project objectives in order to create business value with minimal delay.

Team Empowerment

T a s k 3	Encourage team members to become generalizing specialists in order to reduce team size and bottlenecks, and to create a high-performing cross-functional team.
T a s k 4	Contribute to self-organizing the work by empowering others and encouraging emerging leadership in order to produce effective solutions and manage complexity.

T a s k 5	Continuously discover team and personal motivators and demotivators in order to ensure that team morale is high and team members are motivated and productive throughout the project.
-----------------------	---

Team Collaboration and Commitment

T	Facilitate close communication within the team and with
---	---

a s k 6	appropriate external stakeholders through co-location or the use of collaboration tools in order to reduce miscommunication and rework.
T a s k 7	Reduce distractions in order to establish a predictable outcome and optimize the value delivered.
T a s k 8	Participate in aligning project and team goals by sharing project vision in order to ensure that the team understands how their objectives fit into the overall goals of the project.
T a s k 9	Encourage the team to measure its velocity by tracking and measuring actual performance in previous iterations or releases in order for members to gain a better understanding of their capacity and create more accurate forecasts.

Domain V Adaptive Planning

Levels of Planning

T
a
s
k
1 Plan at multiple levels (strategic, release, iteration, and daily) creating appropriate detail by using rolling wave planning and progressive elaboration to balance predictability of outcomes with the ability to exploit opportunities.

T
a
s
k
2 Make planning activities visible and transparent by encouraging participation of key stakeholders and publishing planning results in order to increase commitment level and reduce uncertainty.

T
a
s
k
3 As the project unfolds, set and manage stakeholder expectations by making increasingly specific levels of commitments in order to ensure a common understanding of the expected deliverables.

Adaptation

T
a
s
k
4 Adapt the cadence and the planning process based on results of periodic retrospectives about characteristics and/or the size/complexity/criticality of the project deliverables in order to maximize the value.

T
a
s
k
5 Inspect and adapt the project plan to reflect changes in requirements, schedule, budget, and shifting priorities based on team learning, delivery experience, stakeholder feedback, and defects in order to maximize business value delivered.

Agile Sizing and Estimation

T | Size items by using progressive elaboration techniques in

a s k 6	order to determine likely project size independent of team velocity and external variables.
T a s k 7	Adjust capacity by incorporating maintenance and operations demands and other factors in order to create or update the range estimate.
T a s k 8	Create an initial scope, schedule, and cost range estimates that reflect a current high-level understanding of the effort necessary to deliver the project in order to develop a starting point for managing the project.
T a s k 9	Refine scope, schedule, and cost range estimates that reflect the latest understanding of the effort necessary to deliver the project in order to manage the project.
T a s k 1 0	Continuously use data from changes in resource capacity, project size, and velocity metrics in order to evaluate the estimate to complete.

Domain VI Problem Detection and Resolution

T a s k 1	Create an open and safe environment by encouraging conversation and experimentation in order to surface problems and impediments that are slowing the team down or preventing its ability to deliver value.
T a s k 2	Identify threats and issues by educating and engaging the team at various points in the project in order to resolve them at the appropriate time and improve processes that cause issues.
T a s k 3	Ensure issues are resolved by appropriate team members and/or reset expectations in light of issues that cannot be resolved in order to maximize the value delivered.
T a s k 4	Maintain a visible, monitored, and prioritized list of threats and issues in order to elevate accountability, encourage action, and track ownership and resolution status.
T a s k 5	Communicate status of threats and issues by maintaining a threat list and incorporating activities into the backlog of work in order to provide transparency.

Domain VII Continuous Improvement

(Product, Process, and People)

T a s k 1	Tailor and adapt the project process by periodically reviewing and integrating team practices, organizational culture, and delivery goals in order to ensure team effectiveness within established organizational guidelines and norms.
T a s k 2	Improve team processes by conducting frequent retrospectives and improvement experiments in order to continually enhance the effectiveness of the team, project, and organization.
T a s k 3	Seek feedback on the product by incremental delivery and frequent demonstrations in order to improve the value of the product.
T a s k 4	Create an environment of continued learning by providing opportunities for people to develop their skills in order to develop a more productive team of generalizing specialists.
T a s k 5	Challenge existing process elements by performing a value stream analysis and removing waste in order to increase individual efficiency and team effectiveness.
T a s	Create systemic improvements by disseminating knowledge and practices across projects and organizational boundaries in order to avoid re-occurrence of identified problems and improve the effectiveness of the organization as a whole.

It is important to understand the areas that PMI intend to cover in both Tools and Techniques and Knowledge and Skills required in Agile projects. Below you will find the list of topics that PMI provides as the guideline for these areas but at the same time the list is not intended to be exhaustive, but is more of indicative in nature.

TOOLS AND TECHNIQUES

	Area Focus
1	Com munic ations
2	Plann ing, Monit oring, . and Adopt ing
3	Agile Estim . ation
4	Agile Anal ysis . and Desig n
5	Produ ct Qualit . y
6	Interp ersona . l skills

7.	Value-Based Prioritization	ROI, NPV, IRR, Compliance, customer valued, Prioritization, Relative prioritization/Rankings, requirements reviews, minimal viable product (MVP) MMF MoSCoW, Kano analysis
8.	Risk Management	Risk Adjusted Backlog, Risk Burn-Down Graph, Risk-Based Spike, architectural spike
9.	Metrics	Velocity/throughput/productivity, Cycle time, Lead time, Earned Value management, Defects rate, approved iteration, work in progress
10.	Processes Improvement	Kaizen, the Five WHYS, retrospectives, introspectives, process tailoring/hybrid models, control limits, pre-mortem (rule setting, failure analysis), fishbone diagram analysis, Value Stream Mapping

AGILE KNOWLEDGE AND SKILLS

Each Knowledge and Skill mentioned below are important from exam perspective and should be understood clearly.

- Agile values and principles
- Agile frameworks and terminology
- Agile methods and approaches
- Assessing and incorporating community and stakeholder values
- Stakeholder management
- Communication management

- Facilitation methods
- Knowledge sharing/written communication
- Leadership
- Building agile teams
- Team motivation
- Physical and virtual co-location
- Global, cultural, and team diversity
- Training, coaching, and mentoring
- Developmental mastery models (for example, Tuckman, Dreyfus, Shu Ha Ri)
- Self-assessment tools and techniques
- Participatory decision models (for example, convergent, shared collaboration)
- Principles of systems thinking (for example, complex adaptive, chaos)
- Problem solving
- Prioritization
- Incremental delivery
- Agile discovery
- Agile sizing and estimation
- Value based analysis and decomposition
- Process analysis
- Continuous improvement
- Agile hybrid models
- Managing with agile KPIs
- Agile project chartering
- Agile contracting
- Agile project accounting principles
- Regulatory compliance
- PMI's Code of Ethics and Professional Conduct

CHAPTER-WISE COVERAGE OF EXAM CONTENT

This book is organized in a fashion that each and every Domain along with the *Tools and Techniques* and *Knowledge and Skills* mentioned in PMI guidelines are covered. Below is the mapping of the topics in each chapter of the book that will help you find out which Tools and Knowledge areas are covered in what chapter.

Domains and Tasks

Domain I Agile Principles and Mindset		Covered in Chapter
Task 1	Advocate for agile principles by modeling those principles and discussing agile values in order to develop a shared mindset across the team as well as between the customer and the team.	Chapter 2
Task 2	Help ensure that everyone has a common understanding of the values and principles of agile and a common knowledge around the agile practices and terminology being used in order to work effectively.	Chapter 2
Task 3	Support change at the system or organization level by educating the organization and influencing processes, behaviors, and people in order to make the organization more effective and efficient.	Chapter 2
Task 4	Practice visualization by maintaining highly visible information radiators showing real progress and real team performance in order to enhance transparency and trust.	Chapter 5
Task 5	Contribute to a safe and trustful team environment by allowing everyone to experiment and make mistakes so that each can learn and continuously improve the way he or she works.	Chapter 2

k 5		
T a s k 6	Enhance creativity by experimenting with new techniques and process ideas in order to discover more efficient and effective ways of working.	Ch apt er 2
T a s k 7	Encourage team members to share knowledge by collaborating and working together in order to lower risks around knowledge silos and reduce bottlenecks.	Ch apt er 2
T a s k 8	Encourage emergent leadership within the team by establishing a safe and respectful environment in which new approaches can be tried in order to make improvements and foster self-organization and empowerment.	Ch apt er 9
T a s k 9	Practice servant leadership by supporting and encouraging others in their endeavors so that they can perform at their highest level and continue to improve.	Ch apt er 9
Domain II Value-Driven Delivery		
Define Positive Value		
T a s k 1	Define deliverables by identifying units that can be produced incrementally in order to maximize their value to stakeholders while minimizing non-value-added work.	Ch apt er 10
T a s	Refine requirements by gaining consensus on the acceptance criteria for features on a just-in-time basis in order to deliver value.	Ch apt er 8

k 2		
T a s k 3	Select and tailor the team's process based on project and organizational characteristics as well as team experience in order to optimize value delivery.	Ch apt er 8
Avoid Potential Downsides		
T a s k 4	Plan for small releasable increments by organizing requirements into minimally marketable features/minimally viable products in order to allow for the early recognition and delivery of value.	Ch apt er 11
T a s k 5	Limit increment size and increase review frequency with appropriate stakeholders in order to identify and respond to risks early on and at minimal cost.	Ch apt er 5
T a s k 6	Solicit customer and user feedback by reviewing increments often in order to confirm and enhance business value.	Ch apt er 8
Prioritization		
T a s k 7	Prioritize the units of work through collaboration with stakeholders in order to optimize the value of the deliverables.	Ch apt er 10
T a s	Perform frequent review and maintenance of the work results by prioritizing and maintaining internal quality in	Ch apt

k 8	order to reduce the overall cost of incremental development.	er 10
T a s k 9	Continuously identify and prioritize the environmental, operational, and infrastructure factors in order to improve the quality and value of the deliverables.	Ch apt er 7
Incremental Development		
T a s k 1 0	Conduct operational reviews and/or periodic checkpoints with stakeholders in order to obtain feedback and corrections to the work in progress and planned work.	Ch apt er 10
T a s k 1 1	Balance development of deliverable units and risk reduction efforts by incorporating both value-producing and risk-reducing work into the backlog in order to maximize the total value proposition over time.	Ch apt er 5
T a s k 1 2	Re-prioritize requirements periodically in order to reflect changes in the environment and stakeholder needs or preferences in order to maximize the value.	Ch apt er 8
T a s k 1 3	Elicit and prioritize relevant non-functional requirements (such as operations and security) by considering the environment in which the solution will be used in order to minimize the probability of failure.	Ch apt er 5

T a s k 1 4	Conduct frequent reviews of work products by performing inspections, reviews, and/or testing in order to identify and incorporate improvements into the overall process and product/service.	Chapt er 8
--------------------------------	--	---------------

Domain III Stakeholder Engagement

Understand Stakeholder Needs

T a s k 1	Identify and engage effective and empowered business stakeholder(s) through periodic reviews in order to ensure that the team is knowledgeable about stakeholders' interests, needs, and expectations.	Chapt er 2
T a s k 2	Identify and engage all stakeholders (current and future) by promoting knowledge sharing early and throughout the project to ensure the unimpeded flow of information and value throughout the lifespan of the project.	Chapt er 2

Ensure Stakeholder Involvement

T a s k 3	Establish stakeholder relationships by forming a working agreement among key stakeholders in order to promote participation and effective collaboration.	Chapt er 4
T a s k 4	Maintain proper stakeholder involvement by continually assessing changes in the project and organization in order to ensure that new stakeholders are appropriately engaged.	Chapt er 4
T a s	Establish collaborative behaviors among the members of the organization by fostering group decision making and	Chapt er 4

k	conflict resolution in order to improve decision quality and reduce the time required to make decisions.	
---	--	--

Manage Stakeholder Expectations

T a s k 6	Establish a shared vision of the various project increments (products, deliverables, releases, and iterations) by developing a high-level vision and supporting objectives in order to align stakeholders' expectations and build trust.	Ch apt er 4
T a s k 7	Establish and maintain a shared understanding of success criteria, deliverables, and acceptable trade-offs by facilitating awareness among stakeholders in order to align expectations and build trust.	Ch apt er 4
T a s k 8	Provide transparency regarding work status by communicating team progress, work quality, impediments, and risks in order to help the primary stakeholders make informed decisions.	Ch apt er 4
T a s k 9	Provide forecasts at a level of detail that balances the need for certainty and the benefits of adaptability in order to allow stakeholders to plan effectively.	Ch apt er 4

Domain IV Team Performance

Team Formation

T a s k 1	Cooperate with the other team members to devise ground rules and internal processes in order to foster team coherence and strengthen team members' commitment to shared outcomes.	Ch apt er 9
T	Help create a team that has the interpersonal and	Ch

a s k 2	technical skills needed to achieve all known project objectives in order to create business value with minimal delay.	apt er 9
Team Empowerment		
T a s k 3	Encourage team members to become generalizing specialists in order to reduce team size and bottlenecks, and to create a high-performing cross-functional team.	Ch apt er 9
T a s k 4	Contribute to self-organizing the work by empowering others and encouraging emerging leadership in order to produce effective solutions and manage complexity.	Ch apt er 9
T a s k 5	Continuously discover team and personal motivators and demotivators in order to ensure that team morale is high and team members are motivated and productive throughout the project.	Ch apt er 9
Team Collaboration and Commitment		
T a s k 6	Facilitate close communication within the team and with appropriate external stakeholders through co-location or the use of collaboration tools in order to reduce miscommunication and rework.	Ch apt er 9
T a s k 7	Reduce distractions in order to establish a predictable outcome and optimize the value delivered.	Ch apt er 9
T	Participate in aligning project and team goals by sharing	Ch

8	<p>a project vision in order to ensure that the team s understands how their objectives fit into the overall goals k of the project.</p>	apt er 9
---	--	-------------

T a s k 9	<p>Encourage the team to measure its velocity by tracking and measuring actual performance in previous iterations or releases in order for members to gain a better understanding of their capacity and create more accurate forecasts.</p>	Ch apt er 5
-----------------------	---	-------------------

Domain V Adaptive Planning

Levels of Planning

T a s k 1	<p>Plan at multiple levels (strategic, release, iteration, and daily) creating appropriate detail by using rolling wave planning and progressive elaboration to balance predictability of outcomes with the ability to exploit opportunities.</p>	Ch apt er 5
-----------------------	---	-------------------

T a s k 2	<p>Make planning activities visible and transparent by encouraging participation of key stakeholders and publishing planning results in order to increase commitment level and reduce uncertainty.</p>	Ch apt er 6
-----------------------	--	-------------------

T a s k 3	<p>As the project unfolds, set and manage stakeholder expectations by making increasingly specific levels of commitments in order to ensure a common understanding of the expected deliverables.</p> <p>***</p>	Ch apt er 5
-----------------------	---	-------------------

Adaptation

T a s k 4	<p>Adapt the cadence and the planning process based on results of periodic retrospectives about characteristics and/or the size/complexity/criticality of the project deliverables in order to maximize the value.</p>	Ch apt er 9
-----------------------	--	-------------------

T a s k 5	Inspect and adapt the project plan to reflect changes in requirements, schedule, budget, and shifting priorities based on team learning, delivery experience, stakeholder feedback, and defects in order to maximize business value delivered.	Chapt er 10
-----------------------	--	-------------------

Agile Sizing and Estimation

T a s k 6	Size items by using progressive elaboration techniques in order to determine likely project size independent of team velocity and external variables.	Chapt er 6
T a s k 7	Adjust capacity by incorporating maintenance and operations demands and other factors in order to create or update the range estimate.	Chapt er 6
T a s k 8	Create an initial scope, schedule, and cost range estimates that reflect a current high-level understanding of the effort necessary to deliver the project in order to develop a starting point for managing the project.	Chapt er 6
T a s k 9	Refine scope, schedule, and cost range estimates that reflect the latest understanding of the effort necessary to deliver the project in order to manage the project.	Chapt er 6
T a s k 1 0	Continuously use data from changes in resource capacity, project size, and velocity metrics in order to evaluate the estimate to complete.	Chapt er 6

Domain VI Problem Detection and Resolution

T a s k 1	Create an open and safe environment by encouraging conversation and experimentation in order to surface problems and impediments that are slowing the team down or preventing its ability to deliver value.	Ch apt er 9
T a s k 2	Identify threats and issues by educating and engaging the team at various points in the project in order to resolve them at the appropriate time and improve processes that cause issues.	Ch apt er 11
T a s k 3	Ensure issues are resolved by appropriate team members and/or reset expectations in light of issues that cannot be resolved in order to maximize the value delivered.	Ch apt er 11
T a s k 4	Maintain a visible, monitored, and prioritized list of threats and issues in order to elevate accountability, encourage action, and track ownership and resolution status.	Ch apt er 11
T a s k 5	Communicate status of threats and issues by maintaining a threat list and incorporating activities into the backlog of work in order to provide transparency.	Ch apt er 11
<h2>Domain VII Continuous Improvement (Product, Process, and People)</h2>		
T a s	Tailor and adapt the project process by periodically reviewing and integrating team practices, organizational culture, and delivery goals in order to ensure team	Ch apt er 4

k 1	effectiveness within established organizational guidelines and norms.	
T a s k 2	Improve team processes by conducting frequent retrospectives and improvement experiments in order to continually enhance the effectiveness of the team, project, and organization.	Ch apt er 8
T a s k 3	Seek feedback on the product by incremental delivery and frequent demonstrations in order to improve the value of the product.	Ch apt er 4
T a s k 4	Create an environment of continued learning by providing opportunities for people to develop their skills in order to develop a more productive team of generalizing specialists.	Ch apt er 9
T a s k 5	Challenge existing process elements by performing a value stream analysis and removing waste in order to increase individual efficiency and team effectiveness.	Ch apt er 8
T a s k 6	Create systemic improvements by disseminating knowledge and practices across projects and organizational boundaries in order to avoid re-occurrence of identified problems and improve the effectiveness of the organization as a whole.	Ch apt er 8

Tools and Techniques

Area	Focus (Tools and Techniques)	Covered in chapter
1	Communications	Information radiator Team space Agile tooling Osmotic communications for co-located and/or distributed teams Two-way communications (trustworthy, conversation driven) Social media-based communication Active listening Feedback methods Brainstorming
2	Planning, Monitoring, and Adopting	Reviews Kanban board Task board Timeboxing Iteration and release planning Variance and trend analysis WIP limits Daily stand ups Burn down/up charts Cumulative flow diagrams Backlog grooming/refinement Product-feedback loop
3	Agile Estimation	Relative sizing/story points/T-shirt sizing Wide band Delphi/planning poker Affinity estimating Ideal time

4	Agile Analysis and Design	Product roadmap User stories/backlog Story maps Progressive Elaboration Wireframes Chartering Personas Agile modeling Workshops Learning cycle Collaboration games	Chapter 7
5	Product Quality	Frequent verification and validation Definition of done Continuous integration testing, including exploratory and usability	Chapter 8
6	Interpersonal Skills	Emotional intelligence Collaboration Adaptive leadership Conflict resolution Servant leadership negotiation	Chapter 9
7	Value-based Prioritization	ROI/NPV/IRR Compliance Customer-valued prioritization Requirements reviews Minimal viable product (MVP) Minimal marketable feature (MMF) Relative prioritization/ranking MoSCoW Kano analysis	Chapter 10
	Risk Management	Risk-adjusted backlog	Chapter

8	.	Risk burn down graphs Risk-based spike Architectural spike	ter 11
9	Metrics	Velocity/throughput/productivity Cycle time Lead time EVM for agile projects Defects rate Approved iterations Work in progress	Chap ter 6
10	Process Improvement	Kaizen The Five WHYs Retrospectives introspectives Process tailoring/hybrid models Value stream mapping Control limits Pre-mortem (rule setting, failure analysis) Fishbone diagram analysis	Chap ter 8

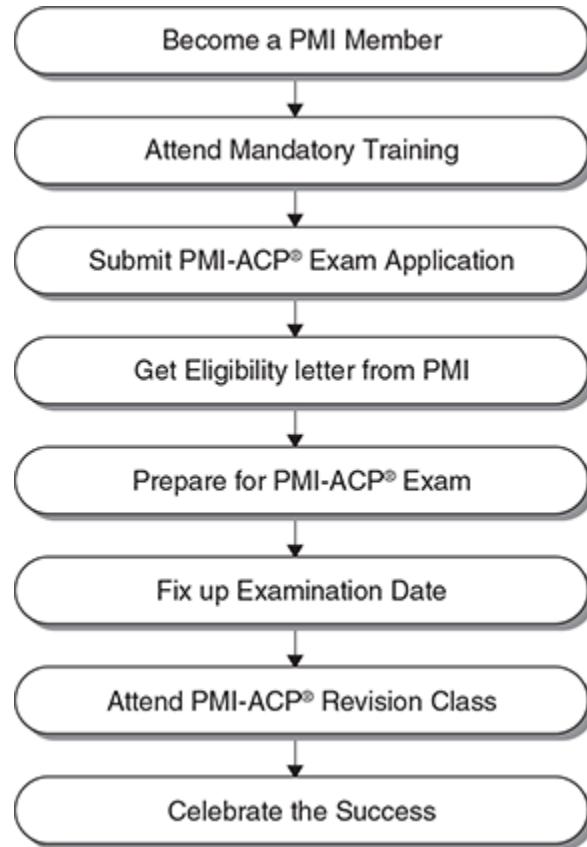
Knowledge and Skills

<i>Knowledge and Skills</i>	<i>Covered in Chapter</i>
• Agile values and principles	Chapter 2
• Agile frameworks and terminology	Chapter 2
• Agile methods and approaches	Chapter 2
• Assessing and incorporating community and stakeholder values	Chapter 4
• Stakeholder management	Chapter 4
• Communication management	Chapter 4
• Facilitation methods	Chapter 9
• Knowledge sharing/written communication	Chapter 9
• Leadership	Chapter 9
• Building agile teams	Chapter 9
• Team motivation	Chapter 9
• Physical and virtual co-location	Chapter 4
• Global, cultural, and team diversity	Chapter 9
• Training, coaching, and mentoring	Chapter 9
• Developmental mastery models (for example, Tuckman, Dreyfus, Shu Ha Ri)	Chapter 9
• Self-assessment tools and techniques	Chapter 8
• Participatory decision models (for example, convergent, shared collaboration	Chapter 7
• Principles of systems thinking (for example, complex adaptive, chaos)	Chapter 9
• Problem solving	Chapter 9

• Prioritization	Chapter 10
• Incremental delivery	Chapter 2
• Agile discovery	Chapter 11
• Agile sizing and estimation	Chapter 6
• Value based analysis and decomposition	Chapter 10
• Process analysis	Chapter 8
• Continuous improvement	Chapter 8
• Agile hybrid models	Chapter 3
• Managing with agile KPIs	Chapter 6
• Agile project chartering	Chapter 2
• Agile contracting	Chapter 5
• Agile project accounting principles	Chapter 5
• Regulatory compliance	Chapter 10
• PMI's Code of Ethics and Professional Conduct	Chapter 12

EIGHT STEPS FOR PMI-ACP® EXAM PREPARATION

We strongly recommend the following eight-steps process for passing the PMI-ACP exam.



Step 1—PMI Membership and Benefits

- Reconfirm the need to become a PMI-ACP
- If yes, become a PMI member by making a simple online enrollment at www.pmi.org
- PMI membership is not mandatory for the PMI-ACP certification, but recommended. You get a CD-ROM of PMBOK latest version + monthly magazine—*PM Network*, a quarterly research journal and monthly newsletter (for a year) + discounts on PMI books.
- Membership is annual (\$129 for the initial membership and \$119 for renewals).
- We get \$60 benefit (discount) while applying for PMI-ACP exam if you are a PMI member.

- Keep the e-receipt for reimbursement purpose, if any.

Step 2—Mandatory Training/Agile PM Education

- It is mandatory to have 21 hours of Agile project management training.
- You can attend classroom training or complete 21 hours through e-learning offered by various education providers.
- While submitting the application kit for PMI-ACP exam, the details of this training are to be furnished.
- It is better to get this training through registered education provider (REP) of PMI.

Step 3—Submit PMI-ACP Examination Application

- Fill up the details in the online application for examination, which includes your education, experience and training.
 - PMI membership ID
 - A detailed split-up for the 2000 hours of general project management experience project-wise (name of the project, reporting line, brief description and hours split up according to initiation/planning/execution/controlling and closure). These 2000 hours should be from the last five years of experience
 - 1500 hours of experience working in Agile project team/Agile methodologies. These 1500 hours should be earned from the last two years of experience, over and above 2000 hours of general project management experience described above.
- You can save the information and also continue later
- After submission, you will get acknowledgment through e-mail

Step 4—Receipt of PMI-ACP Examination Eligibility Letter from PMI

- After submitting the application, within a week, you will get the eligibility letter through e-mail.
- This eligibility letter is valid for one year, and within one year you need to pay the examination fees.
- After getting the eligibility letter, you need to submit the credit card details for PMI-ACP exam fees.
- The application fee is \$435 for a PMI member and \$495 for a non-PMI member.
- Keep the e-receipt for reimbursement purpose.
- After submitting the credit card details you can take the examination.

Step 5—Prepare for the Exam

- Key step: requires investing substantial quality TIME.
- Prepare systematically through various references.
- The main focus should be on PMI-Agile exam content outline, with supporting information from other references.
- Map with real-life experiences while reading.

Step 6—Contact the Local Pro-metric center and Fix Up the Examination Date

- Once confident of taking up the examination, contact the nearest pro-metric center and fix up an appointment for the examination (10–15 days in advance).

Step 7—Attend PMI-ACP Revision Classes

A lot of education providers conduct one-day crash courses for PMI-ACP revision. Try to attend this as it will be helpful for you to pass the exam and also boost your confidence level to appear for the exam.

Step 8—Execute, Succeed and Celebrate

RELAX! You have full 180 minutes to answer 120 questions. You have plenty of time. Certification documentation will be sent to you within 3–4 weeks of passing the exam. Enjoy the benefits of PMI-ACP.

TIPS AND TRICKS FOR PMI-ACP® EXAM

Preparing for the Exam

1. Take at least three months for the whole preparation. Do not try to attempt the exam without preparation.
2. Lock and freeze your exam date well in advance (at least three months before). We can prepare seriously only when the exam date is out (Habit from college days!). The preferable exam dates may not be available as there is always a huge demand for PMI exam.
3. Read the *PMBOK Guide* at least once to understand the traditional project management concepts because PMBOK

is aligned with Agile certification. Read the introductory chapters of this book to understand this relation better.

4. Order the domain as per your interest.... For example, (1) Stakeholder engagement (2) Value-driven delivery, etc.
5. For each domain areas, practice as many questions as possible before going to the next domain. (This will definitely increase confidence in that particular domain). Practice the questions from the exam viewpoint rather than just reading the questions and answers.
6. Take at least one week to study each domain along with practicing questions.
7. Even if you take one week for each domain, you can finish your preparation in six weeks (two months approximately). Another one month for revision and going through the full question papers.
8. Create a study note for yourself in each domain, which will be helpful for you to revise during the exam period.
9. Remember your own success formula from your past academic successes and try to follow a similar study pattern.
10. Be aware of tools and techniques. 50% of the exam is based on tools and techniques. Remaining 50% is based on knowledge and skills required to execute Agile projects.
11. Don't try to memorize anything as it will not be helpful. Try to understand the concepts well.

The content of the PMI ACP Examination and the coverage of the same in this book are discussed in Chapter 1.

Exam Pattern

1. The Exam has 120 multiple choice questions, each having only four choices.
2. Total examination time is three hours (180 minutes).
3. None of the above, all of the above, choices may not usually be there.
4. You can expect some TRUE/FALSE type of questions.
5. READ the questions fully before trying to answer.
6. There are no negative marks for wrong answers, so, do not leave any question unanswered.
7. You can see only one question on the screen at a time. You can answer a question and/or mark it for later review. You can move back and forth through the questions at any point of time before submission.
8. This exam not only tests the knowledge you have but also the application of knowledge. You must understand how the various terminologies used in this book are used in the real world.
9. You need to understand and answer each question based on the PMI viewpoint and not on what your company/you follow.
10. Understand risk calculations (using decision trees, expected monetary value, etc.) and categories of risk.
11. Concentrate more on questions with often, but not always, key words such as Except, Not, At Least, At the Most, Not True, Not False, Except.
12. You need not use all the information provided in the question to answer.
13. You need to select the answer which is close to a given situation.
14. Answer the questions based on the PMI concepts first, and then consider your own experience.

Types of Questions in the Exam

Try to understand and be prepared for the following patterns of questions.

1. Straight from PMI-Agile Certification Content

Usually, there will be a few questions in this pattern where the questions and answers will be straight from the latest edition of Agile exam content outline. For example:

Question: All of the followings are major Agile domain practice prescribed by PMI except:

- Value-driven delivery
- Boosting project performance practices
- Adaptive planning
- Problem detection and resolution

Answer: B

2. Situational Questions

Most of the questions are expected to be in this pattern, where a situation will be described to give context to the problem. Based on the situation, we need to choose the best answer.

Question: You are working in a training organization and the proposal which you submitted two months back to start and conduct a new course to increase the revenue got approved. This is an example of which of the following?

- Strategic Opportunity
- Customer Request
- Legal Requirement
- Employee Request

Answer: A

3. Questions with Unnecessary Information

Not all information included in a question will be used to answer the question. Sometimes, there will be multiple paragraphs of information for each question on the exam.

Question: You are discussing with another project manager who is also preparing for PMI-Agile exam and there were arguments about the number of domains and major tools, techniques discussed by PMI. Which of the following is TRUE?

- 4 domains and 12 major tools and techniques
- 6 domains and 6 major tools and techniques
- 4 domains and 10 major tools and techniques
- 6 domains and 10 major tools and techniques

Answer: D

The exam will have a lot of questions like this and you need to aware of these type.

4. TRUE/FALSE Questions

Question: You are a program manager in discussion with another program manager. You generally discuss project and program relationship. He is telling that if a relationship between projects is only that of a shared client or services, the efforts should be managed as a portfolio of projects rather than as a program. Is it true?

- True
- False

Answer: A

Program management focuses on project interdependencies and helps to determine the optimum approach to managing the projects.

One Day Before the Exam

1. Know the route for the exam center clearly; if possible, try to visit it.
2. Go to bed early (Before 9:30 pm)
3. Don't eat oily foods the day before the exam, don't eat heavy food.
4. Don't drink alcohol the day before as it may reduce your concentration.
5. Relax a lot the day before the exam.
6. Re-visit your exam notes in all the knowledge areas.
7. Visualize success of the next day.

In the Exam Hall

1. Go to the exam hall at least one hour in advance.
2. Carry some snacks, if possible, as it may stop hunger pangs at the end of the exam.
3. Report your presence to the coordinator immediately and do not wait to be called.
4. You must bring your authorization letter from the PMI, as well as two forms of ID with exactly the same name you entered on the exam application.
5. Relax and be cool in the exam hall.
6. Do not get tensed with the surroundings as there will be people who write other types of exams.
7. Last minute preparation may be not helpful in this exam.

8. You will need only two hours to finish the PMI-ACP exam. Take the remaining time for review.
9. Do not spend more than two minutes on the same question, just leave it for review and move ahead. You can come back and review these questions later.
10. Do not expect to answer all the questions, there may be some questions that you have not heard of, or for which you may not know the answers. Do not worry about this kind of questions, leave it for review and move ahead.
11. Follow time management strictly in the exam hall. You should have crossed 40 questions in the first hour and next 40 questions in the second hour. If you have completed anything less than this, try to speed up immediately and you need to use this as a speed indicator.
12. Control your anger, energy level, frustrations in the exam as you may not like some of the questions and may not agree with some of the questions and answers. Don't worry; you need to just move ahead with the exam.
13. You need to keep your energy level up always, from the beginning until the end. Try to take a five-minute break in between, possibly at the end of 1.5 hours to refresh yourself.
14. Read all the four answers completely before picking up the correct answer. Practice this from the beginning of your preparation. It is better to read the answers backward (choice D first, then C, etc.) as this will be helpful to pick the right answer easily.
15. If you do not know the answers correctly try to use elimination method. Eliminate the answers which you are sure is not the correct answer.
16. Try to find out common project management errors in the answers, which may be useful for the elimination purpose.
17. Look for words like 'next,' 'best,' 'never,' 'most,' 'always,' 'except,' 'most likely,' 'less likely,' etc. Take note of these words, and try to understand the real meaning of the questions using 'the keywords' above.

18. Sometimes two negative keywords will be given in the same question and you need to rephrase the question before answering it, otherwise the answer may go wrong. For example, “which of the following is not true” should be rephrased as “which of the following is false” before answering the question.
19. Understand the Agile project management concepts in real-world application before selecting the answers.
20. If you find more than one answer for any of the question, then try to choose the best answer out of these.
21. When you answer “fill in the blank” type of questions, the correct answer may not be grammatically correct.
22. When you are marking any question for review due to confusion in choosing the answers, just note in the paper given, the possible answers according to you. When you review you can straight away choose among the marked one- and also save time reading the other answers.
23. The exam will not be scored until you indicate that you are ready, or the 3-hour time allotted is over (There will be clock tickling at the top right hand corner of the screen indicating the remaining time).
24. Don’t try to click the finish button without answering all the questions. Even you do not know the correct answer in spite of applying all the logic, choose the best option you feel as it may go right and cost you nothing!!
25. A printed summary of test results will be given immediately after the exam.

Common Things an Agile Practitioner Should Know

Agile project managers are generally expected to know the following terminologies and concepts.

Understand these as it will be helpful for you to answer the questions in PMI-ACP exam.

- Deliver working software frequently
- Facilitate the team to solve their impediments
- Meetings expected to happen only with agenda
- Only relevant topics should be discussed in the meeting
- Agenda for the meeting should be set well in advance
- The team should be consulted in the planning
- Agile team should be empowered to take decisions

2

Introduction to Agile Principles and Mindset

KNOWLEDGE AND SKILLS LEVEL

Level 1: High

- *Agile Manifesto Values and Principles*

Level 2: Medium

- *Elements of Project Charter for Agile Project*
- *Agile Frameworks and Terminology*

INTRODUCTION

It has been a subject of long debate since the concept of Agile methodology arrived as to whether Agile is a better methodology than the traditional waterfall methodology. Each of this methodologies is applicable for particular types of projects, and each has its own pros and cons.

Limitations of Traditional Project Methodologies

1. Modern projects are constrained by several uncertainties in terms of what is expected/what is anticipated in the beginning of the project.
2. Even though we have the process ...

3

Agile Methodologies

KNOWLEDGE AND SKILLS LEVEL

Level 3: Low

- *Applying New Agile Practices*
- *Variance in Agile Methods and Approaches*
- *Scrum*
- *Extreme Programming*
- *Adaptive Project Framework*
- *Test-driven Development*
- *Feature-driven Development*
- *Kanban Method,*
- *Agile Unified Process*
- *Dynamic Systems Development Method*
- *Crystal Methods*
- *Lean Software Development. Hybrid Agile Model*

INTRODUCTION

Frameworks and processes that are based on Agile manifestoes are called Agile methodologies or Agile methods.

VARIANCE IN AGILE METHODS AND APPROACHES

In this chapter, we will discuss the most popular Agile methodologies and will discuss the situations where each of these is most relevant.

Examples of Agile methodologies are:

1. Scrum
2. Extreme programming (XP)
3. Test-driven development (TDD)
4. Feature-driven development (FDD)
5. Kanban methodology
6. Dynamic system development method (DSDM)
7. Crystal methods
8. Agile unified process (AUP)
9. Lean software development

APPLYING NEW AGILE PRACTICES

Note: The PMI-ACP exam is based on the concepts of Agile and not on a particular Agile methodology. But we should know the purpose

of each methodology and the context under which each of the methodologies is being used. For the PMI ACP exam, the most important methodologies to know are Scrum, XP, Kanban, and Lean. There may be only a few questions from the other methodologies. So, we will discuss Scrum, XP, Kanban, and Lean in more detail in this chapter, and some highlights of the other methodologies will be given.

SCRUM

Scrum is the most widely used framework of Agile. The term ‘Scrum’ is associated with the sport, rugby, which means “to restart the game after a short infringement when the players pack closely together head down trying to get possession of the ball”.

Scrum is characterized by short time-boxed iterative cycle, adjustable scope, customer satisfaction through quick turnaround time, responding to change quickly, customer collaboration throughout the project.

Figure 3.1 explains the flow in the Scrum framework. The incremental product development is achieved in small time-boxed iterative cycles called sprint. The duration of sprints are fixed, i.e. sprints are time-boxed. Even if all the features in the sprint backlog are not completed, a sprint is marked as completed if the sprint duration is over. The remaining work (if any) is moved back to the product backlog and re-prioritized to be taken up in the future sprints. Normally the requirements are kept unchanged inside the sprints. So, the main steps of scrum are:

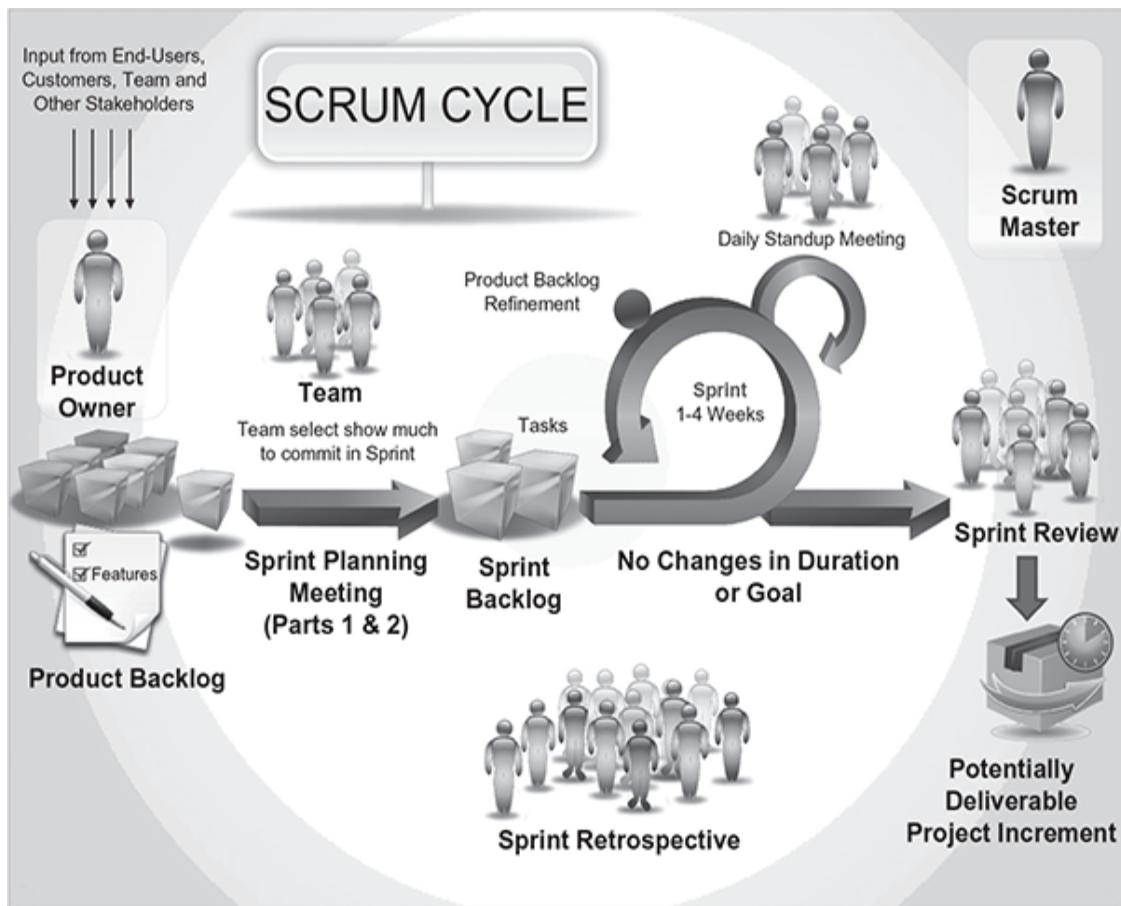


Figure 3.1 Scrum project execution

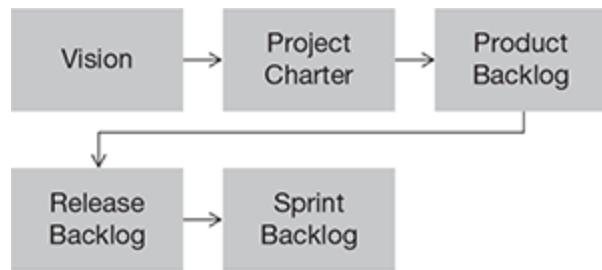


Figure 3.2 Scrum concepts

1. Step 1 : Collect product features (product backlog)

2. Step 2 : Get the product features (backlog) in order according to priority
3. Step 3 : Plan the number of releases
4. Step 4 : Plan the number of sprints in each release
5. Step 5: Sprint Planning Session - Part 1: Clarify the requirements
6. Step 6: Sprint Planning Session - Part 2: Estimate individual tasks
7. Step 7: Create work environment for executing the Sprint
8. Step 8: Execute Sprint
9. Step 9: Conduct stand-up meeting to track the Sprint
10. Step 10: Conduct a Sprint review and Sprint retrospection at the end of Sprint
11. Step 11: Go to Step 5 if there are more Sprints in the release
12. Step 12 : Conduct release retrospection at the end of the release Scrum
13. Step 13 : Go to Step 3 and 4 if there are more releases to execute

Scrum framework has three main concepts: Scrum Roles that define roles and responsibilities of the Scrum team members; Scrum Ceremonies – the meetings and group co-ordinations that help improving the communication; and Scrum Artifacts – the minimum set of artifacts that the scrum team should maintain to properly structure the work.

Scrum Roles are based on an imaginary story of a chicken and a pig who wanted to open a restaurant together. The chicken suggested that they could open a Ham and Eggs restaurant. But then the pig didn't agree as he mentioned 'in that case I would be COMMITTED but you would only be INVOLVED'. Only the people who are committed to the project's delivery are considered as part of the Scrum team. Everyone else who may be involved in the project but is not committed directly are NOT part of the Scrum team. For example, a tester who is involved in testing the product is part of the Scrum team, but a delivery manager who oversees the project and provides help if any resource-related issue arises is not considered as part of the Scrum team. The three main Scrum roles are – the product owner, the Scrum master and the team.

Product owner (PO): This person is the bridge between the end users and the development team. The product owner has the primary responsibility of translating the product vision to user requirements and of ensuring that the project's objectives are met by the end product.

This role is critical to the project's benefit realization and meeting the final goal largely depends on the product owner's ability to create, prioritize, and verify the requirements. The product owner

- creates the initial product vision – overall requirements
- decides on release date and content
- is responsible for the profitability of the product (ROI)
- prioritizes features according to the market value
- interacts with stakeholders and customers to define the product backlog
- adjusts features and priority every iteration, as needed
- accepts or rejects work results

Scrum master: This person is responsible for making sure that the scrum values and principles are followed in the team and more importantly is responsible for helping the team in removing the roadblocks in order to achieve the scrum goals. As the project manager's role in the traditional sense (that is performing planning, monitoring, and tracking activities) is not present in the Scrum framework, this person works as the bridge between the team and the management. The Scrum master

- represents management to the project

- responsible for enacting Scrum values and practices
- removes or helps in removing impediments
- ensures that the team is fully functional and productive
- enables close cooperation across all roles and functions
- shields the team from external interferences

Often the Scrum master role is played part time by one member of the project team. The most important characteristic of the Scrum master is that he/she should be a good communicator and is trusted by all the members of the team.

Team: This is the group of people who are executing the actual tasks of the project like design, development, testing, etc. Some of the key characteristics of an ideal Scrum team are

- Typically consists of 8–12 people. As the key Agile feature is a high level of communication among the team members, the team size is restricted to keep the number of communication channels under control.
- Cross-functional team: programmers, testers, user experience designers, etc. This means that the team has all the skills required for the project available within the team.
- Members should ideally work full time, perhaps with a few exceptions (e.g. database administrator, network support people).
- Teams are self-organizing: ideally, no titles, but rarely a possibility.
- Responsible for the quality

- Estimates the complexity
- Committed to developing functionality

Scrum ceremonies are designed based on the principle that all the team members should be aware of who is doing what, know about the challenges in the project, and work together to improve the way the project is run. The important Scrum ceremonies are Sprint planning, Sprint review, Sprint retrospective, and daily Scrum meeting.

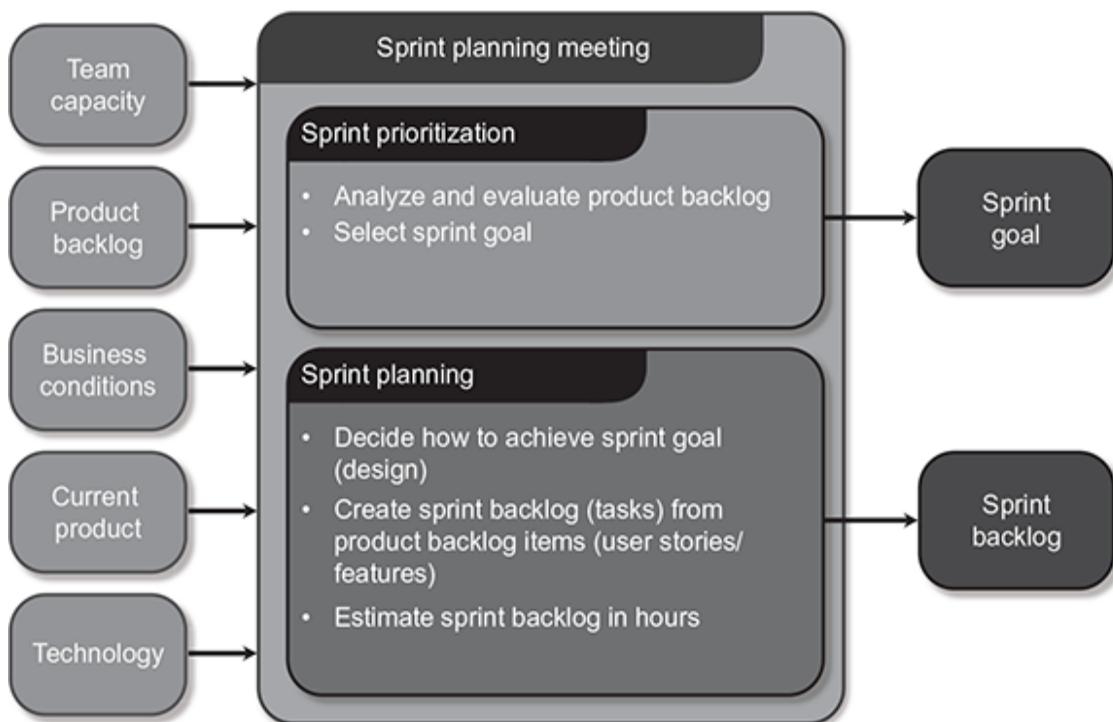


Figure 3.3 Sprint planning

Sprint planning: The preparation for a Sprint starts with the Sprint planning meeting. The input to the Sprint planning meeting is the prioritized product backlog, and the output is the Sprint backlog, which has task estimation and assignment for each backlog item that is decided to be taken up in the current Sprint for implementation. The team selects items from the product backlog they can commit to for completing. Once the Sprint backlog is created, the tasks are identified, and each is estimated (1–16 hours) collaboratively by the team. A high-level design may also be considered while discussing the task-level estimates and assignments.

Normally the Sprint planning meetings are broken into two sessions. Refer to [Figure 3.3](#) for the structure of a Sprint planning meeting. During the first session, the product owner explains the scope of the requirements in the product backlog, and the team gets clarification about what is involved in the implementation of each of these requirements. Based on the team's capacity (called velocity) and the story point estimate of the user stories in the product

backlog, the team will create the first cut Sprint backlog that they will aim to implement during the current Sprint. During the second session, the team tries to find out the tasks for implementing each requirement and also estimate those tasks in hours. The tasks should be small enough (1–16 hours) so that those can be correctly estimated and assigned. Then based on the total number of available days in the Sprint and total number of resources, the team finds out if all the user stories planned for the Sprint during the previous session can be accommodated. Also, the team members sign up for the tasks based on the familiarity and interest areas. At the end of the Sprint planning meeting, a well-defined Sprint backlog is created.

Sprint review: This is the most important meeting conducted at the end of the Sprint where the team presents what it accomplished during the Sprint. This typically takes the form of a demo of new features or underlying architecture. The idea is to show the incremental product or design and get feedback from the users for future enhancement. The ‘Done’ness of the user stories are determined during this review meeting. It is

important to understand that this review is mainly focused on showing the actual product, and thus the use of too much of slides/presentation is discouraged. A ‘2-hour preparation time rule’ signifies that the team should showcase whatever they have achieved during the Sprint without spending time for preparation. The whole team participates in this review, and it is important that the product owner and the end users take part as well. It is ok to invite the people outside the team as well, but the discussions mainly happen within the core group.

Sprint retrospective: Agile framework has a continuous improvement mechanism built within it. As Scrum is running in short iteration cycles, it is easy to take the opportunity to fine tune the processes followed in the Sprint so that the overall scrum effectiveness increases. Sprint retrospective is the team meeting held at the end of the Sprint to discuss what went well, what did not go well, and what improvements can be brought into the way the team is working. So, the Sprint review is about the product improvement,

and Sprint retrospective is about process improvement.

See Chapter ‘Planning, Monitoring, and Adapting’ for details on Sprint retrospective.

Daily scrum meeting/daily stand up: It is very important to make sure that all the team members are aware of the progress each member is making and also they get to know or let others know about any possible roadblock that may hamper the progress of the Sprint. Daily standup call is a short duration time-boxed call involving all team members where the members update each other about the progress, planned activity, and roadblocks. So, the focus of the meeting is to discuss three main points from each – what did you do yesterday? What will you do today? Is anything in your way?

See more details about daily stand up in Chapter ‘Communication’.

Introspective: In Sprint retrospective the main focus is on the immediately completed Sprint.

But there is also a need to look at the system level and find out the tools, processes, and rules that are set up for the whole project are providing the intended level of benefit or are still relevant for the project. This is done through introspective meetings. As we can understand, this is not required for every Sprint or release, but can happen at a lesser frequency. Normally introspective meetings are time boxed to 3 hours, and the first 2 hours are spent in discussing the overall process, tools, rules, and analyzing their benefits or suitability. The last 1 hour is used to create the implementation plan for the suggested changes.

Introspection provides the scope for continuous improvement at the system level, while retrospectives and daily standups incorporate continuous improvement at more granular levels subsequently.

Scrum artifacts are the items that help in maintaining the scrum goal and the final output. Mainly four artifacts are maintained in Scrum projects.

Product backlog: A product backlog contains the requirements that are nothing but a list of all desired work on the project. This is ideally expressed such that each item has value to the users or customers of the product. One important aspect of product backlog is that the backlog items are prioritized by the product owner and can be reprioritized at regular intervals.

Sprint backlog: A subset of the product backlog (a group of requirements) that is taken up for implementation within a Sprint is called Sprint backlog. There can be an intermediate backlog prepared between product backlog and Sprint backlog called the release backlog. The relation of these three types of backlogs are depicted in Figure 3.4. Changes to the Sprint backlog are kept to a minimum within a Sprint as too much change in the backlog can hamper the team's focus on the final outcome of the Sprint.

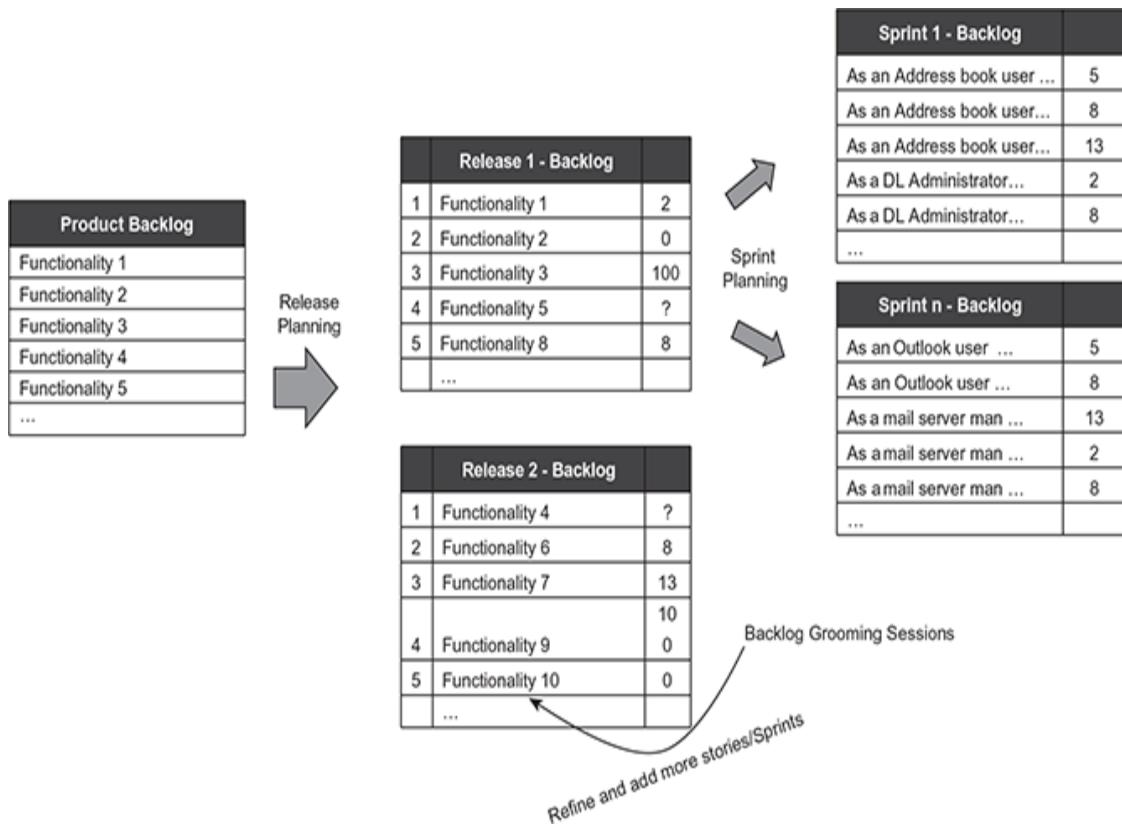


Figure 3.4 Product, Release and Sprint Backlog

See Chapter ‘Agile Analysis and Design’ for more details on product backlog and Sprint backlog.

Incremental product: Scrum expects that at the end of each Sprint there will be a potentially shippable product ready with some feature enhancement over the last version. This incremental product is the main output of the Sprint execution. It is important to understand

that during an early development cycle there may not be a fully working product, and it may very well be an increment on the product architecture that becomes the outcome of a Sprint. Moreover, potentially shippable does not mean that the product will be released to the end users. That may follow the normal release cycle. It is more of a business validation whether the incremental development done by the project team meets the customer's stated requirement. Definition of Done (DoD) is the criteria the customer sets for validating the incremental products and are checked during the Sprint review meeting.

Burn-down charts: Agile projects emphasize on making the progress of the project visible to all stakeholders in an easy way. Contrary to the traditional projects that track the work done, Agile projects focus more on work remaining. One of the powerful visual dashboards that are used in Agile projects is called the burn chart. There are different options like burn-up chart, burn-down chart, cumulative burn-down chart, etc. based on the way of representation. Also,

based on the span of representation, these charts can be at the Sprint or release level.

See Chapter ‘Planning, Monitoring, and Adapting’ for details on burn charts used for tracking Agile projects.

So, consists of three phases, namely, pregame, development phase and post game. Sprint planning happens in pregame phase. Actual sprint execution happens in the development phase. Integration, retrospection happens in the post game phase.

Few points to remember about Scrum are:

1. A product backlog is a list of all known requirements
2. A sprint backlog is a list of all known requirements which the team is going to work in the current sprint
3. A period of work is called sprint (It is actually the current phase of the project) and it is usually time-boxed
4. Daily stand-up meetings happen with the Scrum team
5. A burn-down chart, burn-up chart helps to track progress of the sprint
6. An incremental product is delivered to the customer at the end of each sprint
7. A Scrum master is a facilitator and leader and he is responsible for teaching others how to use the Scrum

process to deal with every new complexity encountered during a project.

8. Story point is the metric used for estimation
9. Concept of velocity is used to measure the productivity of the team

Scrum Challenges

1. Resistance and unlearning required for team members for working on Scrum, especially resources who are used to waterfall.
2. Managing expectations from clients—the clients may assume that any requirements change at any time can be easily accommodated in Scrum projects.
3. Adhering to standards—expectations on detailed documentation.
4. Code brittleness—In Scrum, the same piece of code is refactored multiple times after it is deployed to implement future requirements. This continuous refactoring may make the code brittle and thereby impact the code quality.

EXTREME PROGRAMMING (XP)

I recently came across a project where the customer changes the requirements more often than expected (almost every day) and also the domain and technology tool used was a new one for the company. Customer changes the requirement even after the entire coding gets over, but the customer wanted everything

quickly. The environments outside the project were in total chaos and there was total uncertainty. I felt this project is a good candidate for extreme project programming.

In the following scenarios XP works best:

1. Requirements changing almost every day
2. Total chaos in the project environment
3. Customer wants everything fast
4. Handling with new domain/technology
5. Total uncertainty in everything
6. Small projects that are more easily managed through informal methods

Extreme Programming is an Agile development methodology that was created by Kent Beck in the 1990s.

Extreme programming methodology is built on four guiding values—communication, simplicity, feedback and courage. XP programming has a key focus on improving the **communication**—it may be among the developers or between developers and customer. The concept of the onsite customer (discussed later in this chapter) as well as pair programming

is the way to improve the communication. XP relies on **simple** design and coding models. The code will be easily refactorable if it is kept simple. The small iterations in XP ensure the continuous **feedback** cycles. Thus, the customer gets what he/she wants. It inspires **courage** among the team members because the customer and the developers are working closely, and any bad news should be shared without fear. XP is designed for small collocated teams and tries to ensure quality and productivity as high as possible.

The following 2 principles form the basis of the XP methodology:

Planning Game: Only the immediate iteration is only planned so that we don't plan too far, which has too many unknowns. The three steps involved in the planning game are exploration, planning and steering. In the exploration phase, the customer defines the requirement for the system and the developers estimate the effort. Requirements are captured in the form of User Story. Based on this analysis in the planning phase, both these parties negotiate on the

finalized features to be implemented within the project scope. In the steering phase, the plan gets updated as more knowledge regarding the development and business are acquired.

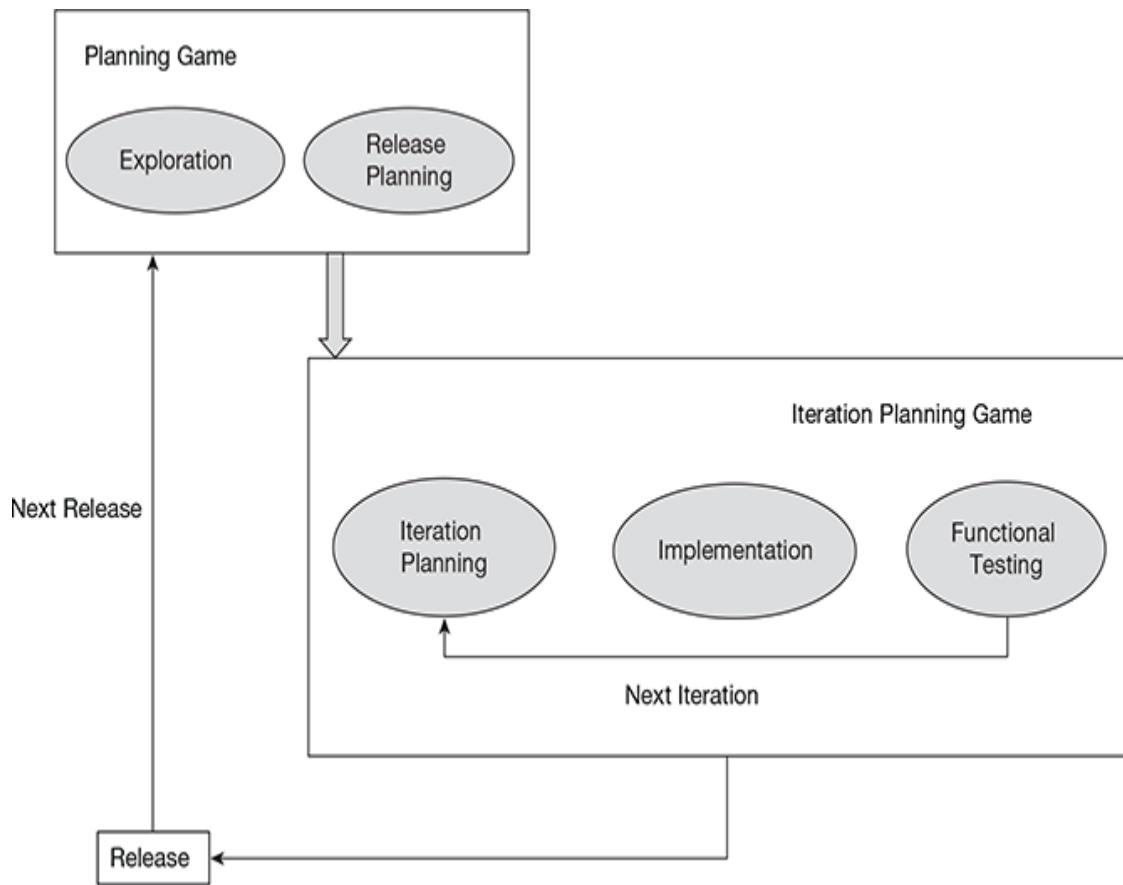


Figure 3.5 Planning Game

Simple design: The programmers should always ask themselves if there is any simpler way to implement the functionality. The best possible design approach which is the most simple one

should be chosen for implementation. This ensures that only the customer-required features are implemented and also it is easy to add incremental functionalities later on this code.

Refactoring: One main pitfall of the quick small releases is that the design and codes created during iteration focus on the immediate requirement and not on the full system. Thus, there is always a possibility of patchy and messy code getting generated after few iterations are completed. Refactoring is a process where the quality of the system is improved by changing the code after few iterations for consistency and maintainability, but making sure that it does not change the behavior or functionality of the system.

Pair programming: A powerful concept used in XP which aims at improving the product quality by introducing an extra pair of eyes while doing the development is called pair programming. Here two developers will work together using one computer to create the design, code and unit testing. The philosophy behind this approach is ‘two heads are better than one’. This ensures that

this two developers are continuously reviewing their work jointly and making sure that the best possible solution is created, and the test cases cover all the scenarios of the system.

Continuous integration: XP methodology goes to the extent of integrating the newly developed codes with the existing one even daily. This ensures that there is always a running version of the product and ensures small release cycles.

Collective code ownership: This concept promotes the idea that all developers own all the codes rather than just part of the system. This makes the refactoring easier and builds a sense of responsibility among the team members.

Coding standards: For the small and quick coding cycles, it is necessary that all the developers follow some standards in terms of naming convention, writing style, exception handling, etc. This will make sure less effort is needed during the refactoring as the developers need not try to decipher other programmer's

code and also reduces the need for internal commenting.

Onsite customer: Improved customer collaboration is one of the basic values of XP. The onsite customer presence ensures the continuous collaboration of the team and the customer and the continuous feedback by the customer. This makes sure what is required by the customer is actually getting developed, and there is a minimum waste due to expectation mismatch.

Focus on testing and automation: XP stresses on the test first approach (See next section Test-driven Development). This gives the developers a goal. The focus on unit testing is to ensure the quality of the product. As XP relies on small iterations, investing on the automated testing is essential.

System metaphor: A common set of terminologies used for the system are called the system metaphor. This ensures easy communication among the project team

members and with the customer and improves communication.

Sustainable pace (40-Hour work week): Team is the focus of the XP methodology. Thus, it makes sure that team works 8 hours per day for 5 days a week making a 40-hour week. Continuous overtime by the team actually reveals a problem, and it is not a long-term solution.

Small releases: Similar to other Agile methodologies, XP also focuses on small incremental releases to end user for getting frequent feedback and providing early customer ROI. This also enables the team to improve the processes in the project and provides the customer required confidence on the progress of the project.

Though XP is a powerful and popular concept in software development, there are few issues that have been highlighted like less documentation produced during development and the scalability due to its focus on small collocated team.

TEST-DRIVEN DEVELOPMENT (TDD)

TDD is a specific engineering practice from XP, which is a way of writing code and creating the design by making those pass the test scenarios. It actually recommends writing the automated unit test first before writing the code. The test will not compile itself at the first instance, so write the basic minimal code in order to make it compile. Run the test and see it fails because we did not implement the full code. Write the full code and make it pass. Refactor for clarity and repeat. This increases the speed and also improves the quality. ‘3 A’ model (Arrange, Act, Assert) is the principle behind TDD. Following are the steps in TDD model (Refer Figure 3.6):

1. Write a single test
2. Compile the test. It will not compile because the code is not written
3. Implement the just enough code to enable the test to compile
4. Run the test and see it fail because no content there inside
5. Implement just enough code to see the test pass
6. Run the test and see it passes
7. Refactor the code for clarity
8. Repeat the process

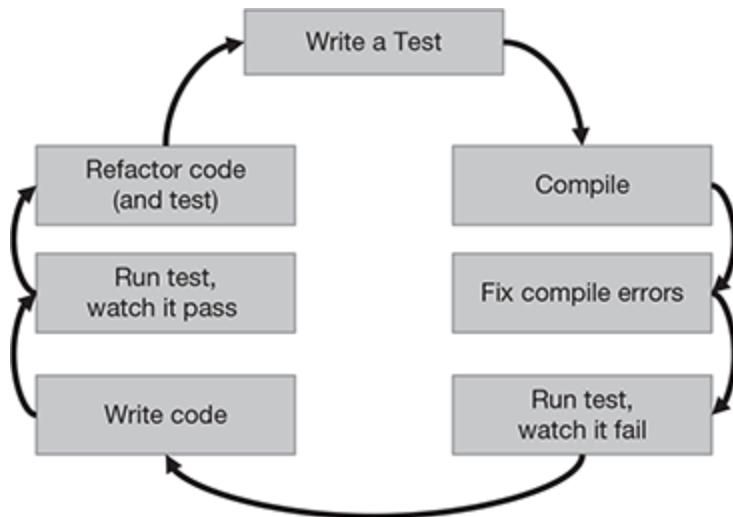


Figure 3.6 TTD

FEATURE-DRIVEN DEVELOPMENT (FDD)

FDD is a very simple methodology and consists of only five processes (Figure 3.7), namely, develop an overall model of the system, build the features list, plan by feature, design by feature and build by feature.

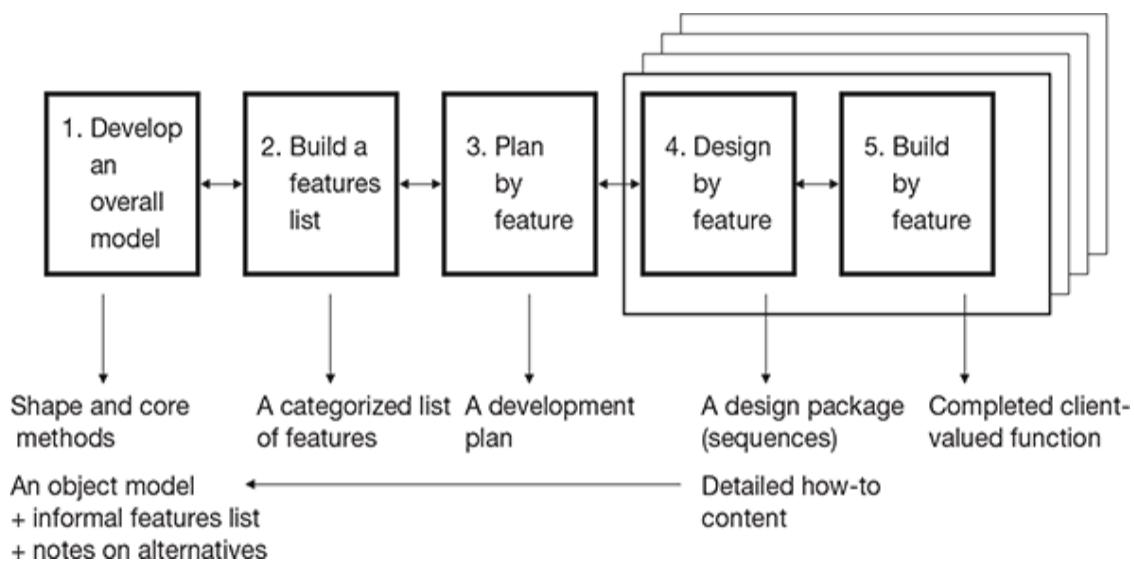


Figure 3.7 FDD five processes

The features to be built are small aspects of client valued functionality that can be expressed in the form of <action> <result> <object>. FDD refers individual code ownership and seeks to avoid refactoring by concentrating on domain knowledge.

FDD also defines the following six roles.

- Project Manager
- Chief Architect

- Development Manager
- Chief Programmers
- Class Owners (aka Developers)
- Domain Experts

KANBAN METHOD

Kanban means ‘signboard’. In early 1940s, the concept of Kanban was introduced by Japanese car manufacturers to ensure that the work in progress items (WIP) match with the capacity of the team, which helps in reducing the inventory and flexible planning and increasing productivity and transparency in the whole process. The Kanban framework later becomes a powerful tool in Lean methodology which focuses on Just In Time delivery and limiting the work queue. The main principles of Kanban are as below:

Reduce cycle time: The time gap between when the work on an item starts and when it is delivered to the end users is called the cycle time. This is the key metric followed in the Kanban framework. So, a Kanban team does not work on fixed time-boxed iterations like in Scrum, but they take up one item from the top

priority list of the product backlog, develop and deliver it, and then move to the next priority items. The team consciously avoids working on too many items in parallel because shifting focus continuously on multiple items reduces the team's efficiency. The product backlog can be reprioritized or changed at any time because the team is not working on too many items at a time. This gives tremendous flexibility to the customer compared to other frameworks like Scrum where the requirements are frozen for the duration of the Sprint.

Features	Tasks Queue	WIP (2)	Tasks Complete	Feature Complete
				

Figure 3.8 Kanban board sample

Overlapping skillset: A team with overlapping skill sets helps in reducing the cycle time because if a single person holds a particular skill set, then that person may become the bottleneck

in the workflow. By reducing the number of work in progress items, Kanban can make the workflow very transparent to all the stakeholders so that the bottlenecks in the process, either due to person dependency or infrastructure dependency, are surfaced very early.

Visual dashboards: Kanban has a high focus on continuous improvement. For ensuring that the team focuses on improvement, Kanban makes the current performance visible through the use of two key reports – control charts and cumulative flow diagrams. These diagrams are discussed in more detail in later chapters, but the main use of these diagrams in Kanban is to identify the issues early in the lifecycle and limit the work queue for optimizing the cycle time. A Kanban board ([Figure 3.8](#)) provides the visual input about how many items are moving within the workflow and which stage those are at.

Continuous development: This principle is one step ahead of the continuous integration principle. The customer does not have to wait for the release cycles to get the new features, but the team is flexible enough to deliver functionalities

continuously as it is focusing on a limited number of items at a time and completing those before taking up more items. This gives great flexibility to the customers to remain competitive in the marketplace throughout the development cycle.

So, as discussed, limiting the WIP items is a key focus for Kanban. In Figure 3.8, WIP (2) indicates that maximum 2 items are allowed to be in WIP at any stage indicating work in progress limit. If any step reaches its limit, then the preceding step must halt so that the bottleneck can be cleared first. For example, if the testing team reaches the WIP limit, then the development should stop sending the code as there is no point in producing more code as it won't reach the end customer without testing. This can reveal a potential bottleneck in the team that it needs more experienced testers to improve the cycle time or may be the testing environment is very slow, which becomes visible through this. Kanban focuses on the continuous flow of work rather than wasting time on time-boxed tasks.

Rules for using Kanban effectively are as follows:

1. Start with a plan. The PFEP (Plan for Every Part) is the cornerstone of the Kanban implementation.
2. Order only what is needed.
3. Make only what is ordered.
4. Kanban is official. No items are made or moved without a Kanban.
5. Quality is of the top order. Defective parts and incorrect amounts are never sent to the next process.

See more content about the Kanban Board in Chapter 5 of this book.

AGILE UNIFIED PROCESS (AUP)

This is the simpler version of the rational unified process (RUP). It is summarized by Scot Ambler as ‘serial in large, iterative in small, delivering incremental release over time’. Inception, elaboration, construction, transition are the four phases of AUP and is risk-driven which contains various optional artifacts.

DYNAMIC SYSTEMS DEVELOPMENT METHOD (DSDM)

It combines the project management life cycle and product development life cycle. Pre-project, project life cycle and post project are the phases of DSDM. Project life cycle phase is broken into five parts, namely, feasibility, foundation, exploration, engineering and development. Roles of DSDM team are developers, testers, technical coordinators, ambassador users and the visionary. Visionary is the person who actually initiated the project and knows about the whole project.

CRYSTAL METHODS

Colour principle is used to denote the set of standards to employ. Maroon is for heavy projects followed by red, orange and yellow and crystal clear (lightest). The above colour grid is used to define the process to be employed.

LEAN SOFTWARE DEVELOPMENT

Lean 5S Tool for Improvement

5S tool (Figure 3.9) is a foundation tool for continuous improvement in lean. 5S are called

housekeeping standards. As per the lean principle 95% of the lead time is non-value adding.

Lean Software Development is adopted from Toyota Production System which combines the principles of lean manufacturing and Lean IT principles. The seven principles that form the Lean system are:

- Elimination of waste (discussed in Chapter 10)
- Create knowledge—the learning from each phase of the project should be propagated to the whole team and take necessary action based on the learning. For example, if there is a high number of a defect found after the release, then the learning in the unit testing needs to be more stringent which is a learning that needs to be propagated to the team.
- Quality is in-built—one of the ways of building the quality is refactoring. The customer must see the quality in the delivered system.
- Defer commitment—commit only the portion for which you have sufficient visibility. There is no point committing something far off in the future as there may be many changes by the time you reach that point. So, defer your commitment as late as possible. This is the same concept as adaptive planning.
- Deliver fast—create small release cycles so that the values are delivered faster to the customer.

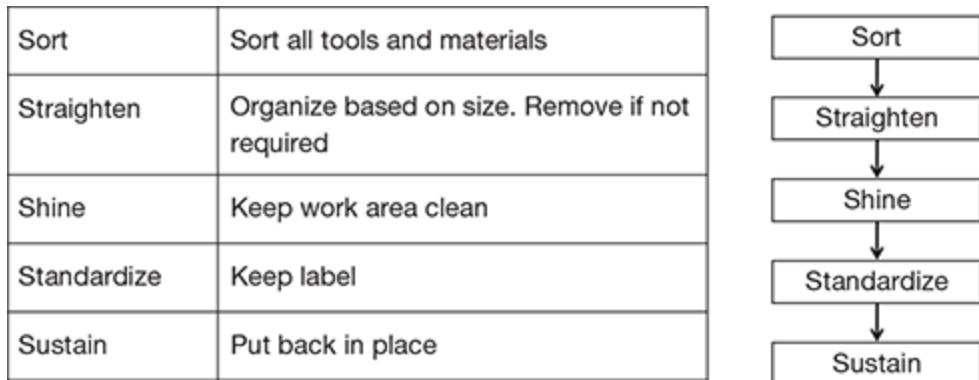


Figure 3.9 LEAN tools–5S

- Empower the team—in software development, people are the key. So, take the time to motivate the team and build the team over time.
- Optimize the whole—see the whole system as an integrated view. There may be different teams working on different parts of the system, but all the team members should have the overview of the whole system and know the purpose of the system. Without this knowledge, the team members can't produce an effective deliverable.

Agile Hybrid Models

In Agile, incremental codes are developed in iterations that build the whole software in steps. In each iteration, the same code base is touched over and over again. It has been observed for large applications that the lack of good architecture upfront makes the product quality poorer at the end. So, there is a group of large

organizations, which believe that moving away completely from the traditional waterfall way of development is not prudent after all. Rather, the new study shows that following a Hybrid approach where the architecture is developed in the waterfall at the beginning and then functionalities are added in iterations yield better architectural stability and end product. Requirements can change throughout the project life cycle in an agile project – but robust architecture platform and framework are important.

The Hybrid approach uses Sprint 0 to build a strong foundation for the project. In Sprint 0, start designing the system architecture and also check the architecture early by creating some features using that architecture. Don't wait till the architecture is frozen to start the feature development as it may cause large rework on architecture later. Also, finalize the NFR (Non-Functional Requirement) list and address those in the system architecture design and build the re-usable components, which can improve velocity in subsequent sprints. See Figure 3.10 for how Sprint 0 is used before the actual

development sprints to create a strong foundation for the project. The duration of Sprint 0 can be larger or smaller than actual sprints based on the project scenario.

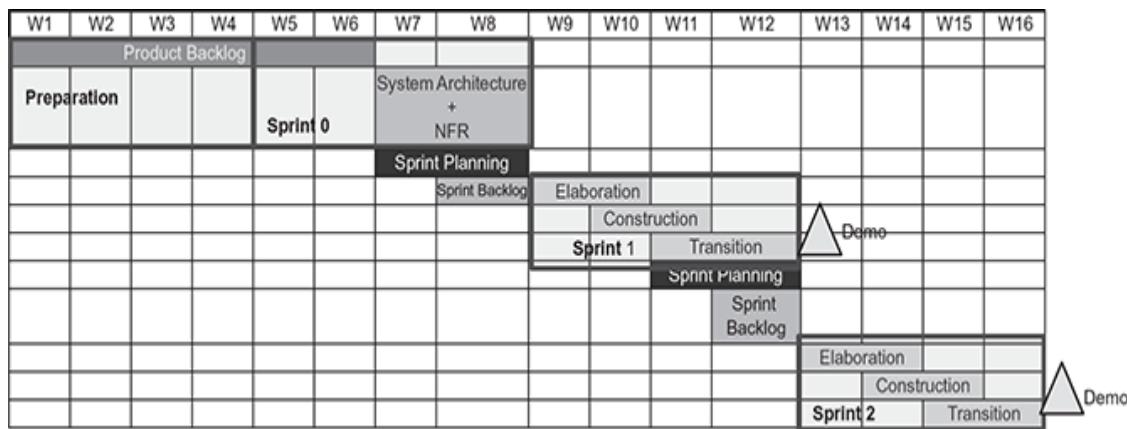


Figure 3.10 Hybrid Agile Model

Summary

Agile methodology is more people oriented. Agile methodology helps us to increase productivity and reduce risks. This paper discussed popular agile methods that are prevalent in the market such as Extreme Programming (XP), Dynamic System

Development Method (DSDM), Scrum, Crystal Methods, Agile Unified Process (AUP), Feature Driven Development (FDD), and Lean software development. From the examination perspective, we need to understand the concepts of these methodologies and the context under which they are applied into a Project.

Chapter 3 Questions and Answers

- Please set yourself a time clock of 1 hour to take this test
- Mark your answers using pencil in the answer sheet provided at the end of this question set
- The correct answers are provided at the end of this question set (after the answer sheet)
- Give one mark for each correct answer for evaluation purpose
- There is no negative marking for the wrong answers
- Practice this test multiple times for better results
- All the very best!

Question 1. Following are the concepts used in Scrum methodology except:

- Vision
- Product backlog
- Velocity
- Colour principles

Question 2. Following are the colours used in crystal methodology except:

- Maroon
- Black
- Red
- Yellow

Question 3. Following are the phases of Scrum methodology

- Pregame, development phase and post game
- Pre-project, project life cycle, post project
- Inceptions, elaboration, construction, transition
- Initiation, planning, execution

Question 4. Following are the Phases of DSDM methodology

- Pregame, development phase and post game.
- Pre-project, project life cycle, post project
- Inceptions, elaboration, construction, transition
- Initiation, planning, execution

Question 5. Following are the phases of AUP methodology

- Pregame, development phase and post game
- Pre-project, project life cycle, post project
- Inceptions, elaboration, construction, transition
- Initiation, planning, execution

Question 6. A list of known requirements in Scrum on which the team is going to work in the current iteration is called

- Sprint backlog
- Release backlog
- Product backlog
- Functional requirement

Question 7. Sprint planning in Scrum happens in

- Pregame phase
- Development phase
- Post game
- Inceptions

Question 8. In Scrum methodology, a period of work is called as (choose the one most appropriate)

- Time-box
- Sprint
- Iteration
- Story point

Question 9. The concept used to increase the team productivity in Scrum methodology is called:

- Velocity

- Sprint
- Iteration
- Efficiency

Question 10. The concept of high speed, high change, high uncertainties is the characteristics of

- Scrum
- Extreme programming
- Crystal
- Traditional project management

Question 11. Different types of kanbans are

- Production kanban, WIP kanban
- Production kanban, Deposit kanban
- Production kanban, withdrawal kanban
- Kanban 1, kanban 2

Question 12. In FDD methodology, the features to be built are small aspects of client valued functionality and that is expressed in the form of

- <action> <result> <object>
- <action> <object> <result>
- <object> <result> <action>
- <object> <action> <result>

Question 13. In FDD developers are referred as

- Team
- Class owners
- Chief architect
- Domain experts

Question 14. Choose the LEAN 5S tool in order

- Sort, shine, straighten, standardize, sustain
- Sustain, straighten, shine, standardize, sort
- Sort, straighten, standardize, shine, sustain
- Sort, straighten, shine, standardize, sustain

Question 15. Project life cycle phase in DSDM is broken into

- 5 parts
- 6 parts
- 3 parts
- 4 parts

Question 16. Choose the order of projects life cycle parts of DSDM

- Foundation, feasibility, exploration, engineering and development.
- Feasibility, foundation, exploration, engineering and development.
- Feasibility, foundation, engineering, exploration and development.
- Feasibility, foundation, exploration, development and engineering.

Question 17. Scot Ambler summarized AUP as

- Serial in large, iterative in small, delivering incremental release over time’.
- ‘Serial in small, iterative in large, delivering incremental release over time’.
- ‘Serial in large, iterative in small, delivering documents over time’.
- ‘Serial in large, iterative in small, delivering product at the end’.

Question 18. PFEP in Kanban stands for

- Part of essential process
- Process for essential part
- Plan for essential part
- Plan for every part

Question 19. A project started following the Kanban methodology. The time gap between when they start working on an item and when they deliver to the end users is called

- Cycle time
- Sprint
- Time box
- Release

Question 20. A team member recently moved from a Scrum-based project to a Kanban delivery model. He asked the other team members what is the normal time-boxed duration for each

development cycle. What will be the response from the team members?

- Normally 2-4 weeks
- In Kanban there is no time-box concept, it is driven by the cycle time
- Depends on the project manager and team availability
- Each item is estimated at the beginning and the timeline is fixed before work begins

Question 21. Kanban uses the following system

- Push system
- Pull system
- Interactive system
- Hub and spoke system

Question 22. The number of process and the number of roles in FDD

- 5, 6
- 6, 5
- 4, 6
- 4, 5

Question 23. Choose the right order of process in FDD

- Build the features list, develop an overall model of the system, plan by feature, design by feature, build by feature

- Develop an overall model of the system, build the features list, plan by feature, design by feature, build by feature
- Develop an overall model of the system, plan by feature, build the features list, design by feature, build by feature
- Build the features list, plan by feature, design by feature, build by feature and develop an overall model of the system

Question 24. Which methodology recommends writing the automated unit test first before writing the code?

- FDD
- TDD
- AUP
- DSDM

Question 25. Following is a type of TDD

- 3A model
- 3C model
- 3B model
- 6P model

Question 26. 3A model of TDD stands for

- Act, arrange, assert
- Arrange, act, assert
- Assert, act, arrange
- Assert, arrange, act

Question 27. One of the powerful features of the Kanban methodology is Continuous

Development. It can be explained as below

- Team members are under continuous pressure to reduce the development time
- Same item is developed simultaneously by multiple team members to check consistency
- Developers are rotated continuously in multiple teams to improve the learning process
- Team is focusing on the development of a limited number of items at a time and thus the customer can get delivery of new features continuously

Question 28. Which one of the below is one of the 12 principles of XP?

- Experienced team
- Strong project management
- Simple design
- Flexible timing

Question 29. In the Hybrid Agile Model, what is the role of Sprint 0?

- To create robust architecture and reusable components
- To give time for the development team to get up to speed
- To do some work of Sprint 1 so that the team can relax during Sprint 1
- To teach the team how a sprint should happen

Question 30. You are a part of the Scrum team for which the implementation is very complex

and has scope for future expansion. The architect cautioned that the product's architecture should be designed very well so that it can handle future enhancement. What would you suggest before starting the development sprints?

- Scrum is not recommended for this project and should move to waterfall
- There should be a Sprint 0 planned in which the product architecture is created and then tested with a few sample features
- There should be two different tracks - one for architecture and another one for feature development
- More testing should be done

Question 31. In XP one of the main values is feedback. What type of feedback is encouraged in XP?

- Customer feedback only as they are the main users of the system
- System only
- Both from the system (through testing) and the customer
- Project manager feedback is most important

Question 32. Which one of the below is not a responsibility of the Scrum master?

- Represents management to the project
- Responsible for enacting Scrum values and practices
- Removes or helps in removing impediments

- Allocates and tracks the tasks of the project

Question 33. All of the followings are the shared values of extreme programming except:

- Late failure
- People first
- Courage
- Quality of life

Question 34. Number of shared values defined in extreme programming is:

- 4
- 5
- 12
- 10

Question 35. Following are the critical success factors defined for extreme project except:

- Keep it simple
- Leadership by commitment
- Real-time communication
- Agile organization

Question 36. Following are the accelerators defined for extreme programming except:

- Keep it simple
- Create ownership for results

- Make change your friend
- Flexible project model

Question 37. ‘Create ownership for results’ in extreme programming is an

- Accelerators
- shared value
- Critical success factors
- Business case

Question 38. ‘Result Orientation’ in extreme programming is an

- Accelerators
- shared value
- Critical success factors
- Business case

Question 39. Vision, product backlog, sprint backlog are related to

- RUP
- Scrum
- AUP
- DSDM

Question 40. Scrum master is (Choose the most appropriate answer)

- Facilitator

- Mentor
- Teacher
- Facilitator and leader

Question 41. A team is following the XP methodology. After working for a few iterations they found that the code has become very patchy and brittle. What step should they take to fix this problem?

- Refactoring
- Recoding
- Reviewing
- Code rewriting

Question 42. A team is trying to decide which Agile methodology they should be using based on the project parameters. They found that the requirements will be changing very often and the solution is not also known and thus need some R&D during implementation. Which methodology will be best suited for them?

- Scrum
- XP
- Kanban
- FDD

Question 43. The meaning of Kanban

- Signs board
- information board
- Message board
- Action board

Question 44. Which agile methodology is being indicated by ‘Customer pulls out what he wants from the supplier. Supplier just supplies what the customer wants’.

- DSDM
- AUP
- Scrum
- Kanban

Question 45. All of the followings are the characteristics of extreme programming except:

- Requirements changing almost every day
- Total chaos in the project environment
- Customer wants everything fast
- Total certainty in everything

Question 46. Chickens and Pigs are the characters in

- DSDM
- AUP
- Scrum
- Kanban

Question 47. Which of the following is not a Scrum cycle activity?

- Weekly inspection
- Daily scrum
- Sprint retrospection
- Sprint planning

Question 48. Lightest project in crystal is denoted by

- Maroon
- Red
- Orange
- Crystal clear

Question 49. Which of the following is a role in Scrum?

- Delivery manager
- Project manager
- Scrum master
- Scrum manager

Question 50. Pair programming is related with

- DSDM
- AUP
- Scrum
- Extreme programming

Answer Sheet for Chapter 3 Questions

Question Number	Answer	Question Number	Answer
-----------------	--------	-----------------	--------

Question 1	Question 26		
Question 2	Question 27		
Question 3	Question 28		
Question 4	Question 29		
Question 5	Question 30		
Question 6	Question 31		
Question 7	Question 32		
Question 8	Question 33		
Question 9	Question 34		
Question 10	Question 35		
Question 11	Question 36		
Question 12	Question 37		
Question 13	Question 38		
Question 14	Question 39		
Question 15	Question 40		
Question 16	Question 41		
Question 17	Question 42		
Question 18	Question 43		
Question 19	Question 44		
Question 20	Question 45		
Question 21	Question 46		

Question 22	Question 47
Question 23	Question 48
Question 24	Question 49
Question 25	Question 50

Answers for Chapter 3 Questions

Question Number	Answer	Question Number	Answer
-----------------	--------	-----------------	--------

Question 1	D	Question 26	B
Question 2	B	Question 27	D
Question 3	A	Question 28	C
Question 4	B	Question 29	A
Question 5	C	Question 30	B
Question 6	A	Question 31	C
Question 7	A	Question 32	D
Question 8	B	Question 33	A
Question 9	A	Question 34	D
Question 10	B	Question 35	A
Question 11	C	Question 36	D
Question 12	A	Question 37	A
Question 13	B	Question 38	B
Question 14	D	Question 39	B
Question 15	A	Question 40	D
Question 16	B	Question 41	A
Question 17	A	Question 42	B
Question 18	D	Question 43	A
Question 19	A	Question 44	D
Question 20	B	Question 45	D
Question 21	B	Question 46	C

Question 22	A	Question 47	A
Question 23	C	Question 48	D
Question 24	B	Question 49	D
Question 25	A	Question 50	C

Explanations for Chapter 13 Answers

1. **Answer D**
2. **Answer B**
3. **Answer A**
4. **Answer B**
5. **Answer C**
6. **Answer A**
7. **Answer A**
8. **Answer B**
9. **Answer A**
10. **Answer B**
11. **Answer C**
12. **Answer A**
13. **Answer B**
14. **Answer D**
15. **Answer A**
16. **Answer B**
17. **Answer A**
18. **Answer D**
19. **Answer A**
20. **Answer B**
21. **Answer B**
22. **Answer A**
23. **Answer C**
24. **Answer B**

- 25. **Answer A**
- 26. **Answer B**
- 27. **Answer D**
- 28. **Answer C**
- 29. **Answer A**
- 30. **Answer B**
- 31. **Answer C**
- 32. **Answer D**
- 33. **Answer A**
- 34. **Answer D**
- 35. **Answer A**
- 36. **Answer D**
- 37. **Answer A**
- 38. **Answer B**
- 39. **Answer B**
- 40. **Answer D**
- 41. **Answer A**
- 42. **Answer B**
- 43. **Answer A**
- 44. **Answer D**
- 45. **Answer D**
- 46. **Answer C**
- 47. **Answer A**
- 48. **Answer D**
- 49. **Answer C**
- 50. **Answer D**

Key Terms

Scrum Scrum uses standard project management concepts but with different terminology and best Practices. Scrum is the most widely used methodology framework of agile.

Extreme programming Extreme Programming is a method that is based upon agile concepts and the supporting XP principals of rapid development, flexibility, team empowerment and customer based quality management.

Test-driven development (TDD) TDD can be used without Extreme Programming also. It actually recommends writing the automated unit test first before writing the Code. Yes, of course the test will not compile itself at the first instance, so write the basic minimal code in order to make it compile. Run the Test and see it fail because we did not implement the full code. Write the full code and make it pass. Refactor for clarity and repeat. It actually increases the speed and also improves the quality.

Feature-driven development (FDD) FDD is a very simple methodology and consists of only 5 Process (Figure 3.6) namely Develop an Overall model of the System, Build the features list, Plan by feature, design by feature, Build by Feature.

Dynamic systems development method

(DSDM) It combines the project management life cycle and Product development life cycle. Pre Project, Project life Cycle, Post Project are the phases of DSDM.

Crystal methods Color principle is used to denote the set of standards to employ. Maroon is for heavy projects followed by Red, Orange, and Yellow and crystal clear (lightest).

Agile Project Communications

KNOWLEDGE AND SKILLS LEVEL

Tools and Techniques: High

- Team Space
- Information Radiator
- Agile Tooling
- Osmotic Communication
- Daily Stand-ups

Level 1: High

- Communications Management
- Stakeholder Management
- Assessing and Incorporating Community and Stakeholder Values

Level 2: Medium

- *Collocation/Distributed Team*

COMMUNICATIONS IN AGILE PROJECTS

The agile framework gives high importance to communication — both internal and external. Immediate and quick feedback from the customer is the fundamental principle of any Agile project.

Agile project systems are themselves designed in such a way that they handle communication requirements effectively. Communication happens more frequently in Agile projects than in ...

5

Planning, Monitoring, and Adapting

KNOWLEDGE AND SKILLS LEVEL

Tools and Techniques: High

- Iteration and Release Planning
- Progressive Elaboration
- Time Boxing
- Cumulative Flow Diagram
- Task/Kanban Boards
- WIP Limits
- Burn Charts
- Retrospectives
- Process Tailoring

Level 2: Medium

- Self-assessment

Level 3: Low

- Agile Contracting Methods
- Agile Project Accounting Principles
- Innovation Games

INTRODUCTION

Bob is a veteran traditional project manager with 15 years of industry experience. He has handled large multi-vendor programs successfully and is very proud of his knowledge and ability. Today One day he had a long discussion with his supervisor regarding a very big and new strategic account which they were bidding to win. The customer told that, it is a very big and long release implementation and the scope of the project will evolve over a period of time. He quoted 'We don't know the full requirements yet as we can't think of each and every feature now, but as and when we see the product evolving, our user group can provide more inputs and they may even ask to fine tune some of the features already implemented'. But Rob's confusion was how to plan the project if the end user is not

clear? This is a common scenario for the Agile projects.

It is impossible for a human being to visualize a large system completely just by thinking about it. On the similar line, the market conditions that affect the outcome of the project are also ever changing. The requirements will always evolve over a period of time in order to adjust to these changing conditions!

In this dynamic project environment shifting from the traditional *Predictive Planning* model to more flexible *Adaptive Planning* model is the need of the hour. Preparing a detailed plan (predictive planning) too far into the future can be wasteful because changes will inevitably happen and they cannot be predicted. The plan can be created with confidence only up to the point where predictability moves into uncertainty. It is safe to plan in detail up to this point, but beyond it, you should plan with a decreasing level of detail and precision. Figure 5.1 explains the difference between predictive planning and adaptive planning. We follow

predictive planning in traditional project management.

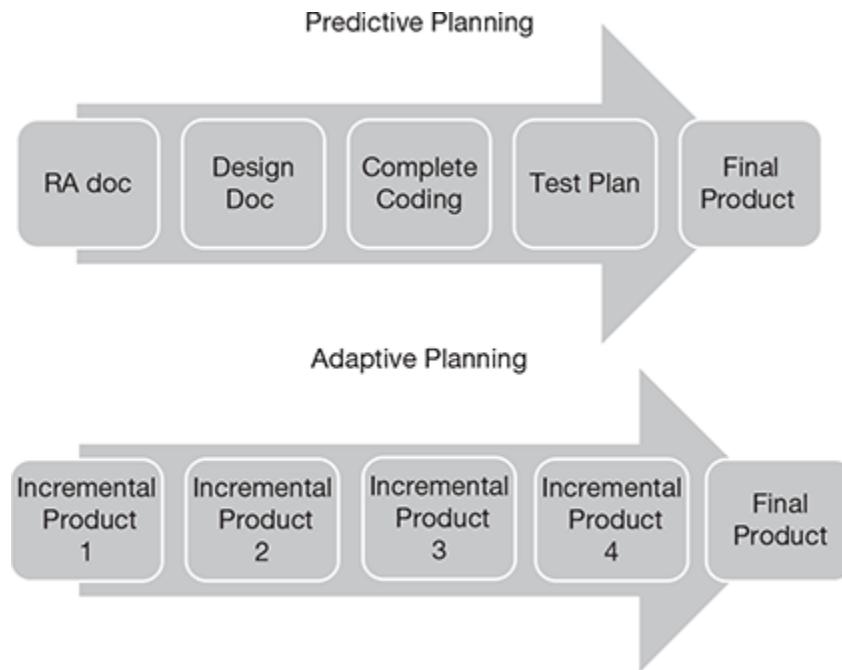


Figure 5.1 Differences between predictive and adaptive planning

WHAT IS ADAPTIVE PLANNING?

Adaptive planning expects the project manager to define only a high-level plan for the far-end features but a detailed plan for the next immediate iteration. This is always a more comfortable and realistic situation to live in, right? This is also referred to as *rolling wave planning* as the process of planning for a project

is happening in waves as the project becomes clearer and complexities unfold. Only key milestones are highlighted in the initial stages for reference.

Rolling Wave Planning is another name of Adaptive Planning

Rolling wave planning is based on the fact that we can see more clearly what is in close proximity, but looking further ahead our vision becomes less clear. So this is like waves—a multi-step, intermittent process—because we cannot provide details very far out in our planning. While creating a detailed well-defined work breakdown structure or task breakdown structure we may be able to plan as much as a few weeks or even a few months in advance with a fair amount of clarity based on the level of complexity and the length of the project. But

beyond this point, we may be able to just highlight the milestone for the rest of the project.

PROGRESSIVE ELABORATION

The terms ***rolling wave planning*** and ***progressive elaboration*** are used interchangeably as both the processes acknowledge the fact that the scope is variable within specified time and cost constraints and thus speculating too much into the future may be a wasteful effort. So an Agile plan should be prepared in such a way that it can be changed easily.

Common reasons for adapting in Agile projects are as follows:

1. Business needs changes
2. Feature larger than expected
3. Customer refinement of requirements
4. Technical constraints
5. A team member is unavailable
6. Third party delay
7. Team throughput lower than expected

Concept of User Story

Before we explore the concept of adaptive planning in detail, we need to understand a basic concept called ‘User Story’ which makes the basis of the planning activity. A user story describes what the user does with the software and how the software responds. In Agile terminology product features are otherwise called as user stories. It is a widely used format for describing features on the product backlog. User stories focus on the consumer of the feature, what they need it to do, and why. User stories are much smaller than the usage requirement artifacts such as use cases or usage scenarios; User stories will be captured in dialogue format and anybody can understand it easily.

Format of User Stories

As a [role] I want to do [feature] so that I can do [reason/benefit]

It is actually the format of [Who] [What] [Why]

[Why] part is optional but it is better to describe that

User Story Components

The basic components each user story should contain are:

- Brief description/Heading – To identify the story.
- Story description (full) – The requirements entailed in the story in detail; this can also include screen mockups, documentation, etc., as needed.
- Acceptance criteria - The confirmation points that indicate the full delivery of the story; this forms the basis of the test case preparation.

Figure 5.2 shows the example of a good user story and its components.

User Stories Example

It is important to recognize that each of the statements below represents a single user story.

1. As an end user, I want to purchase model question papers online so that I can avoid delay in purchase
2. As an end user, I want to chat with the moderator online so that I can clarify my doubts quickly
3. As a researcher, I want to track the research paper status online so that I can remove impediments, if any
4. As a loan applicant, I want to track the status of my loan online
5. As a loan applicant, I want the loan application to be filled online as per my convenience

6. As an author, I want to track the number of books sold by logging into the Web site

See the example of a user story in Figure 5.2

<i>Implementable User Story</i>	
Description	<u>As an end-user, I need to</u> find a car service center in my network in my local area <u>so that</u> I may take my car for repairing.
Acceptance Criteria	<ul style="list-style-type: none">• This story begins when the end user is on the "search for service center" page, enters a zip code and clicks on the "find a service center" button.• This story ends when the end user selects the checkbox adjacent to the center's name.• Display the table and map it as per the attached screenshot.<ul style="list-style-type: none">✓ Show a table of car service centers for the entered zip code, including their address, the types of the cars serviced and a point on an interactive map (allow the user to zoom in, zoom out and move the map with their mouse).✓ Service centers "in-network" should be in bold font in the table and correspond to the red pins on the map. "Out of network" centers should be in regular font in the table with gray pins on the map.✓ The table is sortable by any column. By default, it should be sorted by the name of the pediatrician.
Attachments	Screenshot.pptx; MockData.xlsx

Figure 5.2 User Story Example

3 C's of a User Story

There are three mandatory components of a user story, which are described as 3 C's of user stories as below (refer Figure 5.3)

1. The Card: *the who, what, and why*

1. This explains the user and what he/she wants to do with the system/feature
2. An action that someone in a specific role needs to accomplish and why
3. It usually does not include references to specific numbers, systems, or process steps

Example: As an end-user, I need to find a car service center in my network in my local area so that I may take my car for servicing.

2. The Confirmation: the acceptance criteria

1. What the story author expects to find in terms of completing the action, seeing information on the screen in a specific way, and (possibly) performance
2. Could include attachments if appropriate
3. Notes include negative statements, constraints, or links to material that provide contextual information

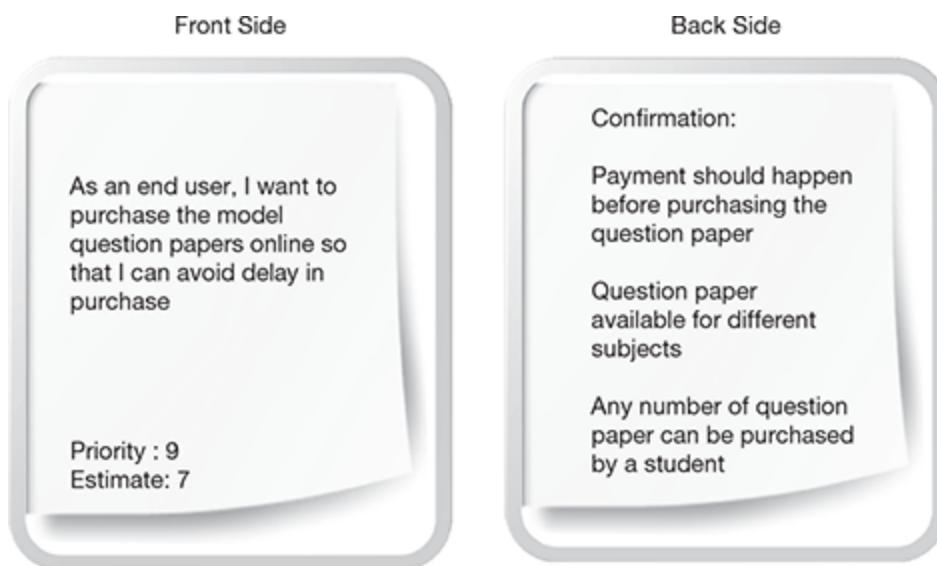


Figure 5.3 3 Cs of a story

Example: Show a list of service centers in my zip code, including their address, the names of the service stations at each location and a point on a map. Those “in-network” should stand out.

3. The Conversation: the story review

1. The story author may leave out some information that the developers and testers may deem as relevant or pertinent, and that could only be discovered via a conversation.
2. The verbal conversations between story authors and developers and testers are the most important step in this process.
3. *Multiple* conversations should take place
4. It is very important to capture the important communications regarding the requirement clarifications or design decisions, which can change the system behavior or are important for testing the final product.

Writing a good user story is not an easy task. Though user stories form the backbone of any Agile project, it is still one of the weakest areas and causes a lot of rework at the later stage. The INVEST principle (coined by Bill Wake) as described below provides a useful guideline of what constitutes a good user story.

INVEST

- Independent
- Negotiable
- Valuable
- Estimateable
- Small-sized appropriately
- Testable

INVEST Principle

- **Independent** – User stories should be independent of each other. The more the dependency among the user stories, the more difficult it will be to estimate and plan them during development. In the case of dependencies, stories can be combined or split and dependent user stories can be marked in the user story card.
- **Negotiable** – The user story details should be worked out after discussions between stakeholders and teams. The more the discussion that happens between the story author and the development team, the better will it help in unearthing the implicit requirements and lowers the chance of rejection.
- **Valuable** – Each story should deliver a business value upon implementation. Agile projects always focus on delivering values to the customer right from the beginning of the project. Prioritization of the user stories is an important step to ensure that more valuable user stories are implemented first.
- **Estimable** – The user story should be estimable so that they can be prioritized and planned. In the case of a story being too big for implementation in a single Sprint, it may be broken down into smaller sub-stories.
- **Small** – A good story should be small in effort, typically no more than two person weeks of effort, including analysis,

design, development, and test.

- **Testable** – A story needs to be testable and satisfy the acceptance criteria.

Guidelines to write good user stories

Apart from the above, there are a few other notable guidelines and characteristics for authoring good user stories:

- Engage with different user types – create personas so that one gets a complete understanding of the possible interactions with the users of the system.
- If the user is not available, consider user proxies. They could be domain experts or former users. In some capacity, business analysts, trainers and technical support persons could also fill the role although with a word of caution as their role could be very focused in a single direction.
- Slice the cake – Instead of splitting stories along technical components, consider one that includes end-to-end functionality and touches each layer. For example, if the team schedules Design, Coding, Testing phases one after the other for a group of user stories then the customer will realize the value from the product late as the outcomes of each of these phases are not usable itself. But if the team focuses on completing the features one by one (by completing the design, development, and test feature wise) then the customer will be able to start using the product earlier.
- Closed stories – Instead of being an ongoing activity, a story should be such that it achieves a meaningful goal. For example, ‘The user can administer the site content’ is not closed when compared against ‘The user can read reviews of hotels before purchasing’.

- Write for one user and in the active voice.
- Use the simplest tool – While some sophisticated tools like Jira and VersionOne are available, user stories are often written on index cards, a paper with feature number, priority number, and story points. Index cards are very easy to work with and are therefore an inclusive modeling technique.
- Write it in simple language – Stories need to be described using simple and understandable language rather than using complex business and technical terms. For example, the user story ‘Students can track the research paper status online’ does not contain any technical or business term and is easily understood by any stakeholder.
- Defer user interface details as long as possible.
- User stories are distinct from use cases, which are descriptions of the set of interactions between a user and the system. User stories are much smaller in scope and detail, short-lived and generally devoid of any user interface specifications.

Acceptance criteria are very important information that should accompany the user stories as the developers and testers get a clear understanding of what the end users want. Some guidelines for clearly writing the acceptance criteria are as follows:

- Describe where the story begins and any initial automated action, like a pop-up screen
- Describe any follow-on actions during or after the user performs the activity, such as mouse overs, tool tips
- Does every activity/function described support the description?
- Is every line item testable and specific?
- Are there numbers (e.g. limits or percentages) in the description? Then move them to the acceptance criteria.

- Words such as all, any, every, etc., misc. are not allowed – provide a list of specific items instead
- Do not use branching logic. A statement that uses “If... then...” is a signal that you need multiple user stories to cover this topic.
- Attachments, “mock” data, etc. may be added
- Business rules that constrain or guide the user activity may be added if required
- Diagram/screenshot/wireframe if a user interface is referenced by the user activity can add good information
- Negative statements (“Nurses should not be able to...”), technical constraints (“this app should work on the windows phone”), and contextual information should not be in the acceptance criteria.

Important considerations for writing user stories are as follows:

1. **Customer focused:** User stories should always be customer focused. Stories are written in the card in the language of the customer. Basically the conversation is recorded in the card and is also validated by the customer. Confirmation also comes in the form of defining the acceptance criteria for the written user story.
2. **Domain experts write user stories:** An important concept is that it is your project stakeholders who write the user stories, and not the developers. User stories are simple enough that people can learn to write them in a few minutes, so it makes sense that the domain experts (the

stakeholders) write them.

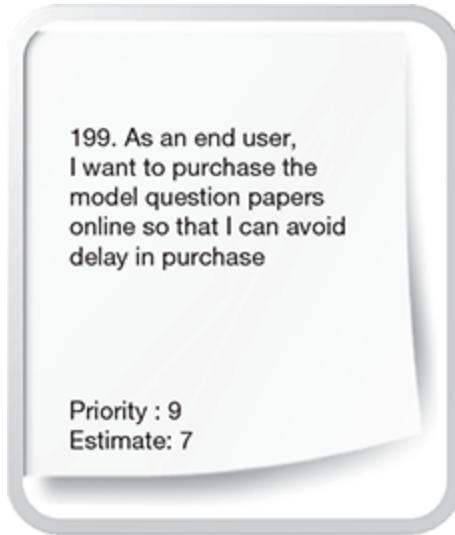


Figure 5.4 User story card

3. **Use the simplest tool:** User stories are often written on index cards, a paper with feature number, priority number and story points. Index cards are very easy to work with and are therefore an inclusive modeling technique.
4. **Write it in simple language:** Stories need to be described using simple and understandable language rather than using complex business and technical terms. For example, the user story 'Students can track the research paper status online' does not contain any technical or business term and is easily understandable by any stakeholders. Here the language does not necessarily mean English. The language needs to be understandable by all stakeholders and that is the only criteria here.
5. **Include unique story number:** The card described in Figure 5.4 includes a unique story number for the user story coded as 199. This unique story number helps us to maintain the traceability between the particular user stories with other artifacts. It also ensures that we do not miss any requirements due to human errors.

6. **Include the priority number:** In Agile, we always develop high priority user stories earlier. Priority also changes over a period of time. Qualitative priority number tags the user stories as high level/medium level/low level, but this is not useful when the number of user stories is more and users may always tend to classify everything (all the stories) as a high priority. Quantitative priority mechanism assigns a unique number to each user story but we need to find out a mechanism to assign a unique number to each story. Choose the best priority based algorithm which works best for you.
7. **Indicate the estimated size:** User stories are being estimated using story points (unit of measurement). The simplest user story is assigned a story point 1. All other story points are measured by comparing it with this simple user story and the comparative score is given. As the story point is assigned based on the relative complexity, a story with story point 2 is twice as complex as the story with story point 1.
8. **Achievable:** User story should be achievable in a single iteration itself and it should be sized appropriately. Well-defined user story follows INVEST model (coined by Bill Wake)
Stories should be independent of each other and it is also sized appropriately and is estimatable and testable.

The detailing of the User Stories also follows the progressive elaboration approach. The high level requirements which normally come from the business users are captured first. These are called the Epics. The Epics are a big chunk of the requirement which may take 1-3 months to

implement. An example of Epic is - The customer should be able to make operations on the Savings Account of the bank through an online process. As you can understand, this requirement is at a very high level and can be broken down into multiple granular level requirement. The next level of granular requirement is termed as Feature. If we use the same example, the Feature lists of the Epic discussed can be to - Allow the customer to check the balance, Allow the customer to make fund transfers etc. the Features can be implemented typically in 2 to 4 weeks. Further detailing is required to make the development Team to work with. The Features are further broken down into User Stories which can be implemented within a week. The User stories derived from the above Features can be - As a customer I want to be able to login into my account, As a customer, I should be able to check the balance in my account etc.

From Figure 5.5, it is clear that features (themes) are the combinations of various related user stories. Epics are a collection of features.

How User Stories Fit into the Adaptive Planning Model

User Stories and Planning

User stories are the basic building blocks of the requirement in a project. So, the planning for the project largely depends on the User Stories and the clarity of the stories. There are two areas where user stories affect the planning process on Agile projects:

1. **Scheduling:** Scheduling depends on the priority. High priority stories are always executed first.
2. **Estimating:** This depends on the story points. A user story with story point 2 is twice as complex compared to the user story of story point 1.

As shown in Figure 5.6, features can be added, modified, deleted and re-prioritized in the system at any point of time by the stakeholders, of course, before the cycle execution. New features (stories, epics) can always get into the system and it is being prioritized. High-priority items are ready for the next immediate cycle of iteration.

The plan for the Agile projects gets detailed out in multiple stages. In each stage, more details are added so that the overall vision of the project is broken down into a daily workable plan. Below are the 5 stages of Adaptive Planning Model:

1. Product visioning (strategic planning)
2. Product roadmap
3. Release planning
4. Iterative cycle planning
5. Daily planning

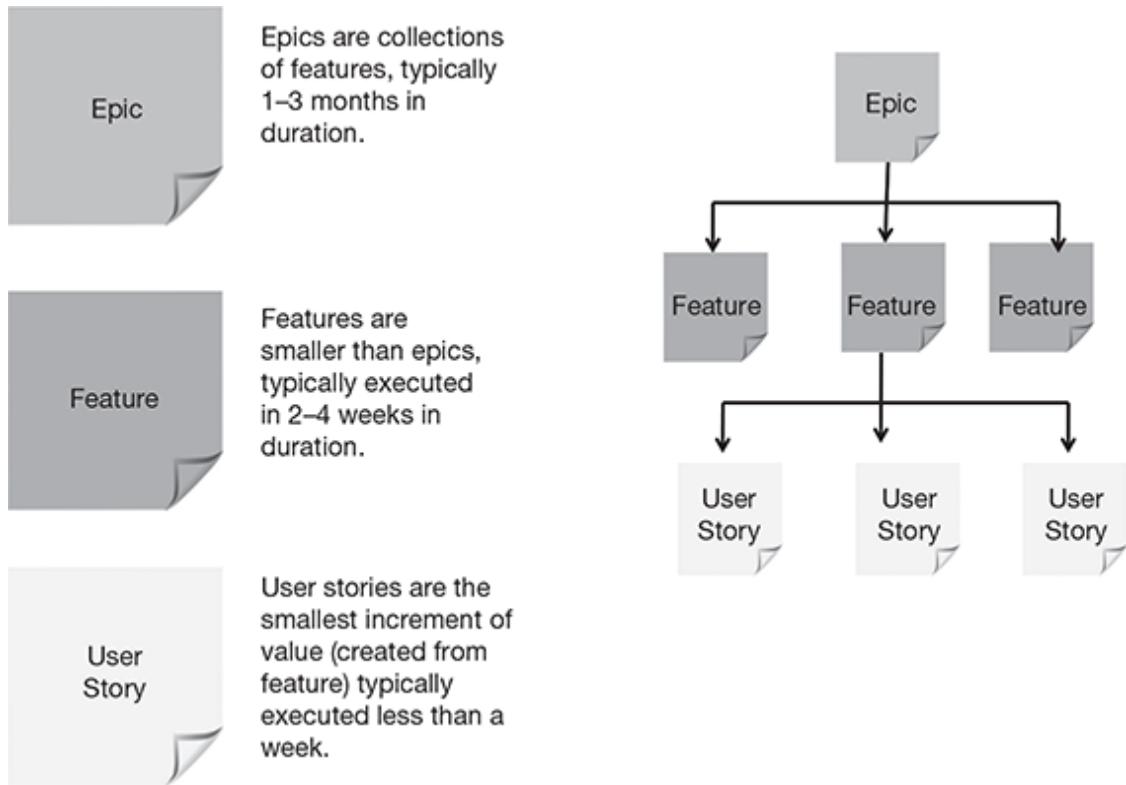


Figure 5.5 Relationships between epic, feature (theme) and story

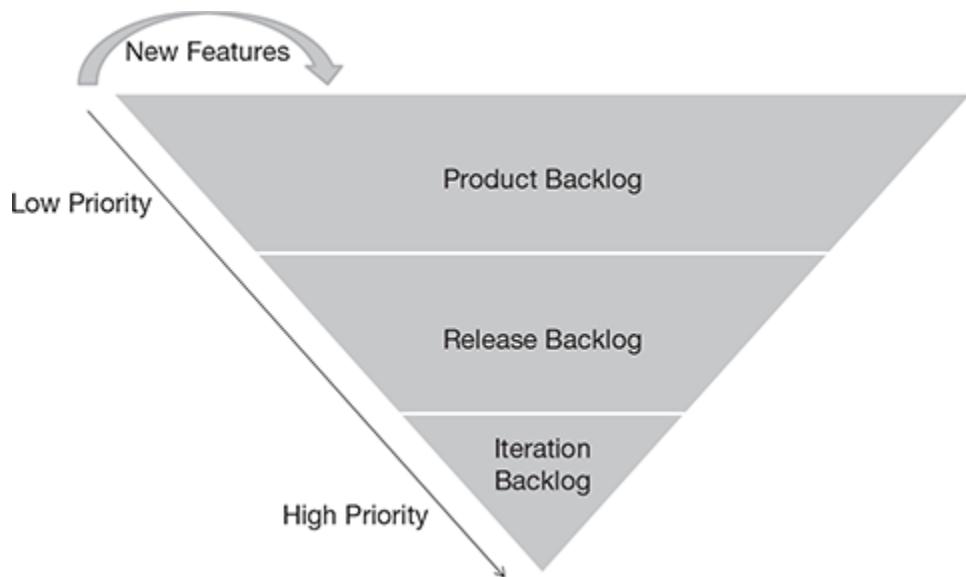


Figure 5.6 Agile project execution

Product Visioning

This planning helps to get a commitment from all the stakeholders involved in creating the product by determining money involved in each level of the iterative cycle. Product vision captures the purpose of the product. By trying to identify the users who will be using the product, their needs which will be addressed in the product and the budget allocated for the product, the overall goal for the project is set. Normally the product owner creates the product vision in discussion with the users and the sponsors so that the end product provides values to the stakeholders. The product vision should be clear and stable for the project team to remain focused. Also, it should be broad enough to engage the different stakeholders who directly or indirectly get affected by the project. It is also necessary to keep the vision short and crisp so that everyone understand the statements in the same way. The product owner what to be changed or created and the overall efforts involved in creating the product, the target customers, a key differentiator from the competitor. Rolling wave planning and

progressive elaboration are the techniques used in creating this plan which will be described in detail in the subsequent chapters.

Product Roadmap

From the overall product vision, a roadmap is created which deals with how the user needs will be addressed by the product. At this stage, the user needs are prioritized and a high-level roadmap is created on features that will be rolled out in a sequence to maximize the ROI. A product roadmap helps in marking the features on a timescale from which the customer and the project team gets a clear idea of which feature is needed at what point in time. An important input to this plan is the minimally marketable feature.

Minimally Marketable Features (MMF)

MMF is the basic minimum feature of the product so that people can start using it rather than waiting for something else. There will be a release after MMF. It may contain single user story or combination of multiple stories. It actually enables incremental delivery of the

product. Multiple MMF makes the whole product. MMF has value to the end user on its own. The term minimal denotes that the feature cannot be split further as in that case it won't be usable. The idea of MMF is to provide the customers maximum value as early as possible. The other terms used for the same concept are Minimum Viable Product (MVP) and Minimum Viable Feature (MVF).

ITERATION AND RELEASE PLANNING

Release and iteration planning happen in phases because the goal becomes clearer only when we drill down to the detail level. There are three broad phases in planning

1. Release planning
2. Iteration planning
3. Daily planning

Each of these phases is executed in cycles with intermediate client checkpoints and post iteration review to align the progress continuously with the plan or to refine the plan, if required based on the progress. The adaptive planning is also

termed Planning Onion because the detailed/inner layer is discovered once the planning of outer layers is complete. See Figure 5.7 for the elaboration of the concept of Planning Onion.

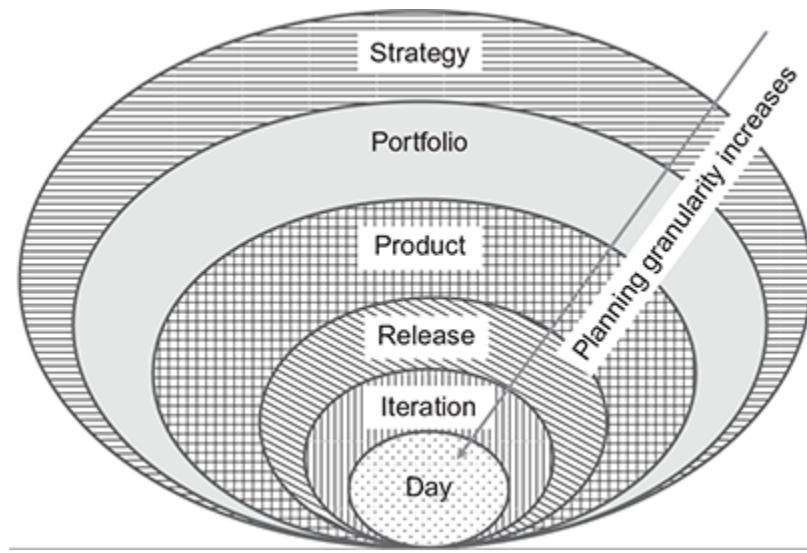


Figure 5.7 Planning Onion

Release Planning

Release depends on the following factors:

1. Deadline from competitor
2. Supporting the contract
3. To meet predetermined schedule
4. Supporting a financial deadline
5. When there is enough value
6. To test the product

The release is actually a set of iterative cycles, and it consists of multiple-related sprints (iterative cycles). Sometimes separate iterative cycle will be executed by the separate team. This is particularly applicable when multiple teams are involved. A typical release happens in 2 to 4 months of time. The release plan is also being defined by the product owner. The delivery team and the product owner need to agree on the number of iterative cycles (Sprints) in each release. Release planning usually happens one full day. The purpose of release planning is to establish a sense of how big a release might be. Release plan actually determines the timeplan (time box) for each release and in turn for each iterative cycle by determining the current team capacity. During release planning, the set of selected user stories is split into much smaller user stories that focus on narrower features.

Types of Release Plan

Two basic types of release plans that are widely used are *scope-boxed release plans* and *time-boxed release plans*. In a scope-boxed plan the work what the team will do is defined in

advance, but the release date (time) is uncertain. In a time-boxed plan, the time and release date is defined in advance, but the specific work what people will do is uncertain and is being elaborated with time.

Time-boxed plans are the most preferred way of release planning. This will constrain the amount of work and people will automatically start doing the prioritization which is the fundamental concepts of Agile.

Information radiator helps to draw the release plan and this plan should always be visible to the team. Practice visualization by maintaining highly visible information radiators showing real progress and real-team performance in order to enhance transparency and trust.

Iteration (Cycle) Planning

Iterations are *time-boxed* and ranges from 1 to 4 week and follow a strict schedule. Iterations are the heartbeat of an Agile project. It is inside the iteration that the actual work is achieved. See the

flow of activities inside an iteration in Figure 5.8.

Iteration plan consists of iteration backlogs, assumptions, risks, actions, dependencies and communication.

The team will determine what features can be accommodated in the current iterative cycle (Sprint). It also orders (priorities) the features of the current immediate iterative cycle. Features are decomposed into tasks (minute level). Now the team can determine the capacity of the team and the tasks on hand in the current iterative cycle (Sprint) and both should be mapped. Done definitions (which will be discussed in detail in the subsequent chapters) will be considered while decomposing the features into engineering tasks. *Engineering tasks* are concrete tasks that the team members will complete. Unlike stories, engineering tasks do not need to be customer-centric. Instead, they are programmer-centric.

The team is responsible for coming up with an exhaustive list of tasks. The team brainstorms the

tasks that they need in order to finish all of the iteration's stories. Some tasks will be specific to a single story; others may be useful for multiple stories. Take the opportunity to brainstorm the high level design or the ideas on how to develop the software during this session without going too much into details. This sometimes brings out the perspectives that a single developer or tester would have missed during the actual software development. Take advantage of your on-site customer's presence to ask about the detailed requirements for each story. What do customers expect when the story is done? This will make the acceptance test criteria for your iteration and is the most important piece for getting the customer sign-off.

After you have finished brainstorming tasks, list those out in a sequential manner and spread them out on the table and look at the whole picture. The team should discuss now: are these tasks enough to finish all of the stories and are there any duplicates or overlaps? If anybody is uncertain about how the plan works with the way the software is currently designed then discuss that and clear out the ambiguity. Discuss and fix

any problems. Finish the brainstorming before starting the estimation.

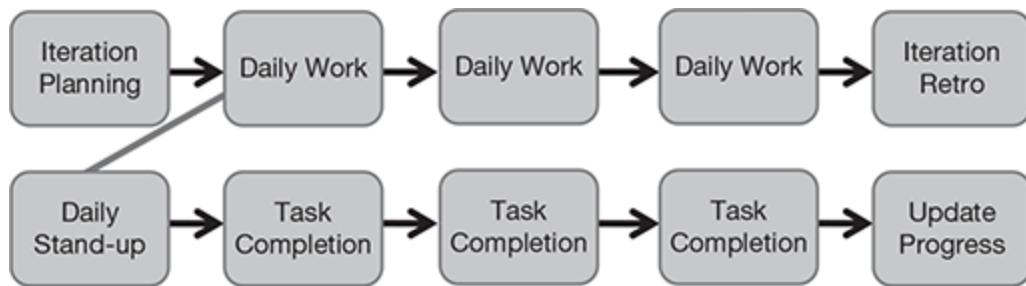


Figure 5.8 Iteration flow

Estimate the Tasks

There are several ways followed of estimating the tasks. But the philosophy is common—the estimate should not be done by one individual, but it is arrived at by the team. The common techniques used for estimating the tasks are discussed in the following.

Feature card modeling is focussed around user interaction and screen design which helps to identify the tasks. Once the tasks are identified iteration planning becomes similar to traditional planning.

By the end of the iteration, there will be the testable/incremental software for each story and are ready to begin the cycle again. Within the iteration, every week the team stops, looks at what it accomplished, and shares those accomplishments with the stakeholders. This brings huge transparency to the whole program and the client organization as well as the software development organization works in such a rhythmic collaboration that it is almost like a single team. See in Figure 5.9 how the user stories are broken down to task level and then resources are identified for each task.

After the iteration demo and retrospective are complete for the current iteration, the iteration planning begins for the next one. Note as the work continues, we need to revise the iteration plan to reflect the changing situation.

Team commitment is to deliver the stories and not tasks, and ask whether your current plan will succeed in that goal. As the project unfolds, set and manage stakeholder expectations by making increasingly specific levels of commitments in

order to ensure a common understanding of the expected deliverables.

A very important parameter that is needed in iteration planning is the velocity, thus start by measuring the velocity of the previous iteration. To do this, take all of the stories that are ‘done done’ and add up their original estimates. This number is the total story points which can be reasonably expected to be completed in the upcoming iteration. Encourage the team to measure its velocity by tracking and measuring actual performance in previous iterations or releases in order for members to gain a better understanding of their capacity and create more accurate forecasts.

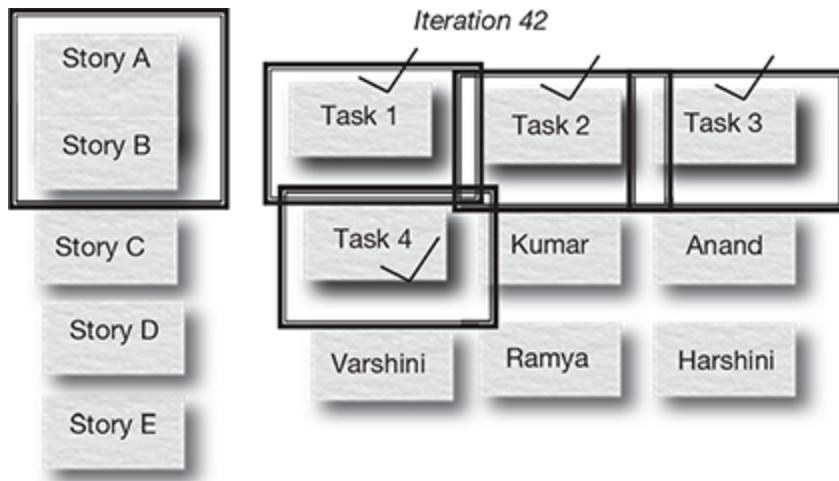


Figure 5.9 Iteration planning board

When you start working on a particular task, take it off from the whiteboard and clip it in the workstation. Mark your initials on the corresponding place at the whiteboard so that people know who is doing that particular task.

After finishing the task put it back on the board and mark it with a green marker (a tick) to indicate that the particular task was done already.

Time-boxing

In traditional project execution models, the main reason for the delay actually comes as an accumulation of small delays over a long period

of time and over a large number of tasks. If you ask a developer how far she is from finally releasing the code at first you may get the answer: ‘Almost done, just have to finish the test’. After a couple of days, you may hear ‘This will be done as soon as I fix this bug’. Then again after few days you may find that ‘This will be complete as soon as I research this cause of error... no, really’. Before you know it, you have lost about a week and your schedule is under pressure.

If you accumulate this scenario for the entire team then each delay of only a few hours gets multiplied across the thousands of tasks in a project. The cumulative effect is devastating.

Iterations provide you an avenue to avoid this surprise. Iterations are exactly 1 to 4 week long and have a strictly defined completion time. This is a time-box: work ends at a particular time regardless of how much is finished. The iteration time-box does not prevent problems but it reveals them early so that you get the opportunity to correct the situation. In Figure

5.10, the time boxing of different activities inside an iteration is highlighted.

Normally the iteration demo marks the end of the iteration followed by the team retrospective.

Iterations follow a consistent, unchanging schedule in which each activity is time-boxed:

1. Demonstrate previous iteration (up to an hour)
2. Hold retrospective on previous iteration (one hour)
3. Plan iteration (half an hour to 4 hours)
4. Commit to delivering stories (15 minutes)

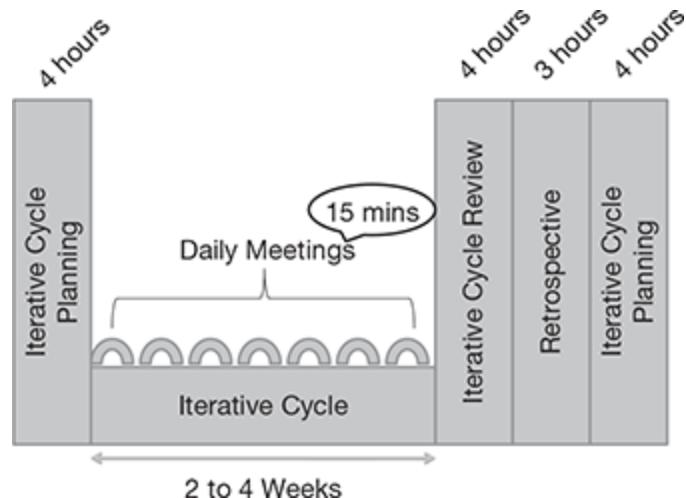


Figure 5.10 Time boxing

5. Develop stories (remainder of iteration)
6. Prepare release (1 hour)

This helps the team keep focused on the main activities and avoid spending time on the low priority tasks.

DAILY PLANNING

During the daily stand-up meeting (scrum meeting), the team provides an update on the status of the previous day's work, reports blocks, if any for executing the tasks and commits on the next work. This happens for a maximum duration of 15 minutes only.

In daily meetings, questions are answered quickly. Issues if any are identified and addressed quickly. Team members will get a common understanding of how the sprint is progressing. Remember that they are not reporting the status to the project manager but they are reporting it to themselves.

The daily Scrum meeting is held at the same time each day, at a time which works best for the team, it mostly happens at the beginning of the day. The Scrum master facilitates daily Scrum meeting, attended by all team members. It is the

responsibility of the Scrum master to take a note of the impediments the team members are facing during this meeting and get those resolved either himself/herself or with the help of the management.

1. Questions are answered quickly
2. Issues are identified and addressed quickly
3. Team members have a common understanding of how the Sprint is progressing

Each team member describes these three things:

1. What was done since the last daily Scrum meeting
2. What is the to-do plan before the next daily Scrum meeting
3. What issues faced

Issue resolving meeting will happen afterward and it is not part of this meeting as otherwise finishing the meeting within the time-box is difficult and moreover it will also waste other people's time who are not involved with the issue.

The product owner may attend this meeting if it being requested by the team and if there is a

necessity for the same. If he attends this meeting he will take notes of the issues/clarifications/doubts of the team members and will be addressed by the product owner at the later point of time.

Iterative cycle backlog (sprint backlog) is a to-do list of that particular iterative cycle (sprint). Iterative cycle backlog is being created by the team based on the high priority list of release backlog. A release consists of multiple iterative cycles (sprints).

ADAPTIVE PLANNING AND ORGANIZATIONAL CULTURE

Generally, doing a detailed estimation and prioritizing stories only for the immediate cycle of iteration is more than enough. This works best for small and frequent releases.

The organization needs to define the project success in terms of value rather than based on the triple constraint of the project.

We need to revisit the plan at least once per iteration. Checkpoints also ensure that we deliver what is required.

Effective Planning

Following are the guidelines for effective planning through which we can maximize ROI (Return on Investment):

1. Release early and frequently
2. Adapt your plan
3. Keeping your options open
4. JIT planning
5. Empower the team to take part

Agile manifesto's highest priority is to satisfy the customer through the early and continuous delivery of valuable software. So, try to increase the value of each release. The plan needs to be considered for learning which will give more opportunity to improve. Releasing early not only helps to give value to the customer earlier but also helps to correct the mistake in the plan quicker and so we need to plan for an early release.

Agile adapts quickly to the changes occurring in the business environment at the customer side. 30 to 50% of the requirements are expected to change during the execution of the projects. Since the Agile project is customer driven, if the customer needs any new requirements, it can be added into the system quickly without any trouble (in the next iteration/iterative cycle or in the next release). Project is able to fine-tune/adapt/change automatically as per the prevailing market conditions.

Just in Time (JIT) planning ensures no re-work in the plan and also helps to accommodate changes in the system, whereas in traditional project management we need to create change request for every change and we also need to follow change management process even for minor changes.

The team needs to take part in the planning in order to make it realistic and as a result, the team commits for execution. In fact, the planning is being owned by the team rather than the project manager. If the team is not involved in the planning process, they may not take ownership

of the project which will be a burden on the project manager. Planning actually should be team-based rather than machine or process based. In a traditional project, planning is done by following a set of process in which the human factors mostly are not considered because of which there is always a gap between the planning and the execution part.

AGILE CONTRACTING METHODS

Agile projects by its nature of execution are entirely different from the traditional method, and so we need to find out the way our contracting methods are to be changed as per the execution model.

- 1. Price/fixed scope/fixed time contract:** Even though these kinds of contracts are not suitable for executing Agile projects, we need to add suitable changes/columns in our contract while managing Agile projects. Customer needs to work closely with the team and the contract needs to review every iterative cycle (Sprint) of work. If required both the parties should agree to replace the scope with other priority scope and of course both should be the same story point. The customer must give signoff at every stage (Sprint) of the project to proceed to the next sprint. In fact, the customer needs to do this continuously at the daily stand-up meetings. Team (executing organization) also

needs to find innovative ways of executing this kind of projects like using reusable components from other projects, using templates, etc. Both the parties should have the option to break the contract at any point of time when their objectives are not met. Both the parties should work closely to prioritize the requirements (Stories) and so at the end of the projects we have only the requirements which are not so important (prioritization) to the customer. The customer also should pay for every sprint of the projects rather than paying the entire amount at the end of the project. This will help both the parties. We also should have a provision to convert our fixed bid contract to T&M contract at any point of time.

2. **Time and material contract:** This is the most suitable method of contracting for Agile projects. In this type of contract, the vendor is being paid based on the hours spent by the team members.

For executing Agile projects the customer may seek a trial period (say 6 months) to understand the culture/structure/resource combination to execute the projects. This trial period can be increased slowly.

Projects at T&M are usually driven by the customer and he can decide what the team members need to do. This kind of contract requires mutual trust of both the parties. The customer should trust the vendors about the resource capabilities/skill level/experienced level. These kinds of contracts also should be reviewed every 6 months and that will be an opportunity for the customer to ask for the replacement and change of resources, if required. For vendors also this is an opportunity to increase the resource rate (per hour dollar value) to the customer. If the contracting is for multiple years then we can mention the fixed rate hike (say 5%) every year, but we need to negotiate this upfront with the customer for rate discount. At the end of the day, it is business for both the parties and

it needs to be worked out in a suitable way to benefit both the parties.

3. **Time and material contract with CAP:** Projects at T&M are usually driven by the customer and he can decide what the team members need to do. But the number of team members (CAP) can be fixed upfront. For example, the customer needs to decide how many resources he needs during peak hours and during average. Customer can say that he needs 3 to 5 resources for a sprint at any point in time and so he can change the number of resources between sprints, if required. This kind of contract is called T&M with CAP.
4. **Fixed price per story point:** As the pure fixed price contracts are not suitable for Agile projects, fixed price for every story point can be opted. But we should have proper method of defining the story point in this case. A number of story points can be determined later, but the price per story point between the customer and the vendor is being agreed on. For example, if the number of story points in the project are 10 and we agree to pay \$300 per story point then the customer needs to pay \$3000 once all 10 story points are accepted by the user. Agreeing on the common definition of the story point to a project by way of defining the base user story is important so that there is no confusion on the pricing part based on story points.
5. **Fixed price per release:** We can think of going for fixed price every release. This is something better than fixed price per story point as the period of a release will be greater than that of story point. Even in this case, we need to define proper plan for the release and both the parties should agree on all other dimensions of release. For example, the number of sprints in each release is to be decided upfront. Percentage of acceptable deviation in the story (requirement) also needs to be agreed upfront.

AGILE PROJECT TRACKING

Information Radiators

Agile project space should be an ‘informative workspace’ which is similar to war room of the traditional project team, where the team is displaying product backlogs, release backlogs, iteration backlogs which are usually in the form of story cards or task cards on the wall. Other graphs and charts prepared by the team are also being posted on the wall which is sometimes called ‘information radiators’ or ‘visible big charts’.

There is no hard and fast rule for the information radiators and the team displays the items, whatever they feel is useful for them and the execution of the projects.

Self-assessment

The beauty of the information radiators lies in the fact that these help in creating the self-correcting team who can assess their team’s progress themselves and then take necessary

corrective action through the team discussion. All the charts and graphs that are mentioned below are made so intuitive that each member will be able to use those for assessing the team's current performance, remaining work, required progress rate, etc.

Burn Charts

Burn charts help to track the project. It helps to showcase the amount of work completed, amount of work remaining and projected completion dates. Team members are actually motivated by the amount of work remaining steadily reduced.

There are three types of burn charts as shown in Figure 5.11:

1. Burn-down chart
2. Burn-up chart
3. Combined chart

Burn charts are used for sprints as well as for releases and so we have six different types of

burn charts, three types each for releases and sprints.

Burn-down Chart

- How much work remains to be completed?

Burn-up Chart

- How much work has been completed?

Combined Burn Chart

- How much work has been completed and how much work remains?

Figure 5.11 Types of burn charts

1. Release burn-down chart
2. Sprint burn-down chart
3. Release burn-up chart
4. Sprint burn-up chart
5. Release combined burn-chart
6. Sprint combined burn-chart

Charts are drawn using line chart or bar style chart.

Burn-down Charts

Burn-down chart is also called an estimate to complete chart. Since it is moving downwards it

is called burn-down chart. It actually draws the story point yet to be completed. As we move on, we complete some story points and thus pending story points start to come down and hence the chart points go down. The concept of burn-down chart is depicted in [Figure 5.12](#). A sprint burn-down chart (or sprint burn-down graph) compares actual and expected progress, and shows whether the team is ahead or behind expectations ([Figure 5.12](#)). Expected progress line is a line drawn between total story points to be completed and zero. Burn-down chart is drawn either with the story points or time estimate. It is up to the team to decide which one to choose for the project; whichever is comfortable with the team, can be chosen.

The team can draw the burn-down graph for sprint, release and multiple releases too. Time scale is longer for release and multiple release charts.

The diagonal line from top left to lower right using the maximum story point (200 in the above case) and zero. This is also called expected line. We draw one more line indicating the remaining

story points (to be completed story points). If that line (bar) is below the diagonal line, the project is ahead of schedule indicating good. If the line (bars) is above the diagonal line, the project is behind schedule indicating some problem.

In the above chart, the actual line is above the diagonal line (for day 2 and 3) indicating that we are going behind and some action needs to be taken.

A sprint burn-down chart also called as ‘sprint burn-down graph’ showcase the total task hours or total points remaining per day. The X-axis represents days in the iteration, while the Y-axis is story points remaining. Burn-down charts are displayed in the team location and it is an effective information radiator.

The same burn-down chart can be displayed using bar graph as shown in Figure 5.13 and you can notice that the chart burns down to zero at the end of the sprint. We should mark only the story point completed on a particular day; we

may be doing some user story on that day and if it is not completed then it should not be marked.

The physical task board is also being used to track the project.

Release Burn-down

Release burn-down chart is drawn between the sprints (X axis) and the corresponding story points (Y axis) remaining at the end of that sprint. It is actually being tracked against the release plan. It is being updated usually at the end of each sprint. In release burn down (Figure 5.14) the work remaining is being shown using story points but it can be shown using ideal days/team days. But it is always better to display in user story points.

	Story Point Completed	Remaining Story Point
Day 0	0	200
Day 1	10	190
Day 2	15	175
Day 3	38	137
Day 4	30	107
Day 5	29	78
Day 6	20	58
Day 7	28	30
Day 8	30	0

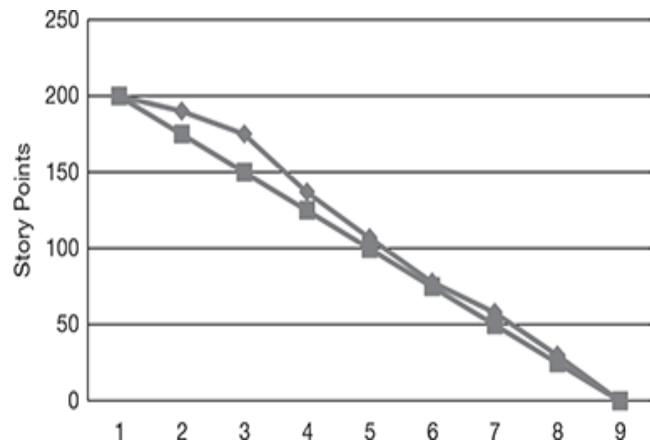


Figure 5.12 Burn-down chart

	Story Point Completed	Remaining Story Point
Day 0	0	200
Day 1	10	190
Day 2	15	175
Day 3	38	137
Day 4	30	107
Day 5	29	78
Day 6	20	58
Day 7	28	30
Day 8	30	0

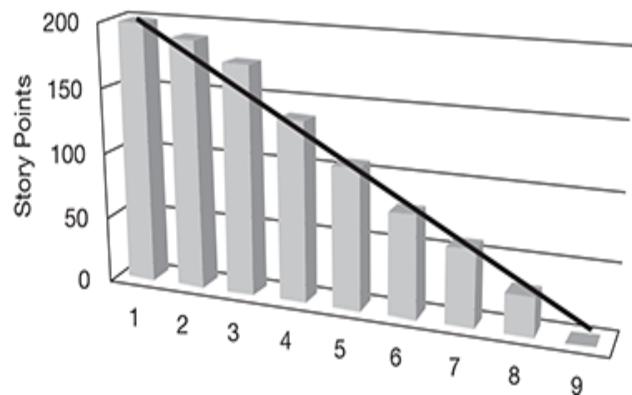


Figure 5.13 Burn-down chart using bar graph

	Story Point Completed	Remaining Story Point
Sprint 0	0	200
Sprint 1	10	190
Sprint 2	15	175
Sprint 3	38	137
Sprint 4	30	107
Sprint 5	29	78
Sprint 6	20	58
Sprint 7	28	30
Sprint 8	30	0

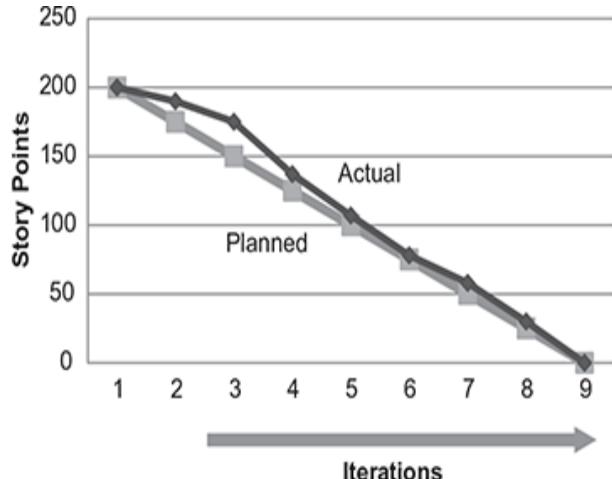


Figure 5.14 Release burn-down chart

Cone of Uncertainty Theory

Burn-down graph indirectly represents cone of uncertainty theory. In a typical project management, the cone of uncertainty actually describes the evolution of the amount of uncertainty during the execution of the project. There are lot of uncertainties in the beginning of a project because we know little about the end product and so estimates are subject to lots of uncertainties. As we move along the project, the uncertainties tend to decrease.

Burn-up Charts

Burn-down chart shows only how much work is pending (remains to be completed) after each iteration and it does not show up whether there is any change in the total story points and it does not show how much work is actually accomplished in each iteration.

Figure 5.15 is a burn-up chart. From the chart, it is very clear that every day the team is progressing towards the total story points and they have completed all the story points exactly at the end of Day 8.

The burn-up chart can be drawn using the bar graph as shown in Figure 5.16. The bar graph clearly indicates that on Day 2, 3, 4 it is not touching the expected line indicating some problem and action needs to be taken on it.

	Story Point Completed	Total Story Point Completed	Total Story Point
Day 0	0	0	200
Day 1	10	10	200
Day 2	15	25	200
Day 3	38	63	200
Day 4	30	93	200
Day 5	29	122	200
Day 6	20	142	200
Day 7	28	170	200
Day 8	30	200	200

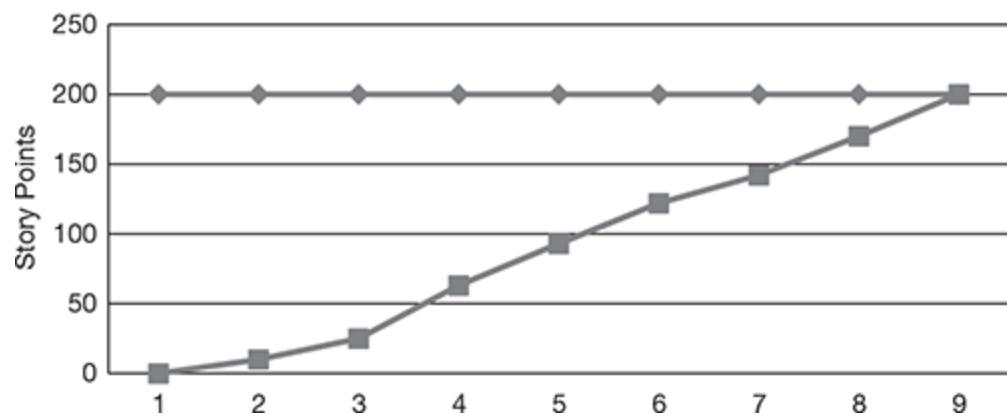


Figure 5.15 Burn-up chart

	Story Point Completed	Total Story Point Completed
Day 0	0	0
Day 1	10	10
Day 2	15	25
Day 3	38	63
Day 4	30	93
Day 5	29	122
Day 6	20	142
Day 7	28	170
Day 8	30	200

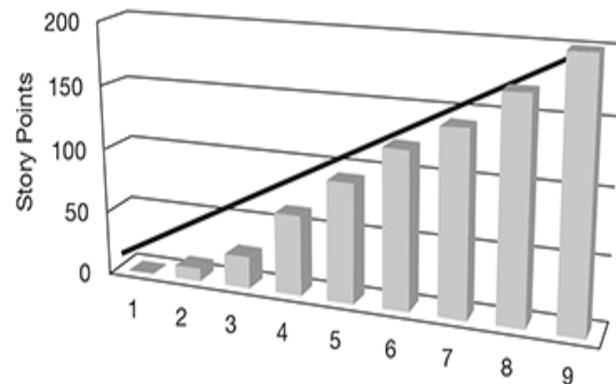


Figure 5.16 Burn-up chart using bar graph

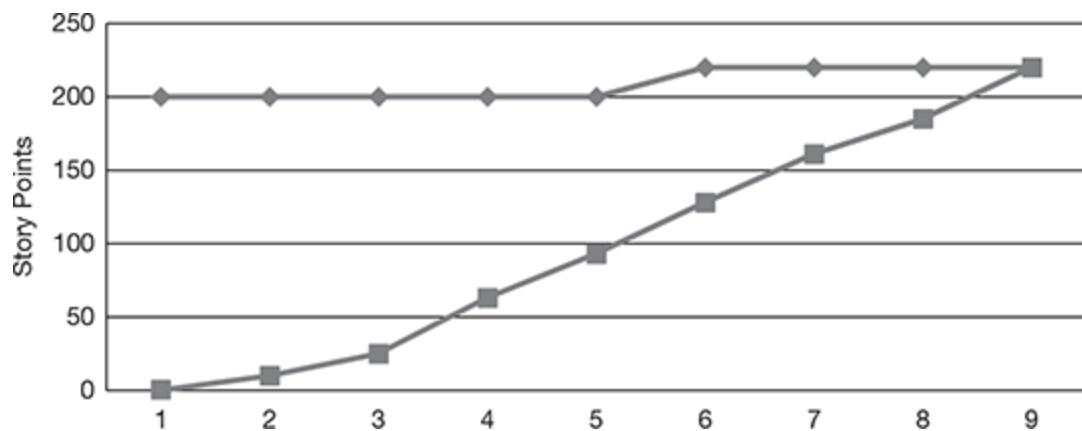


Figure 5.17 Burn-up chart with changes

In an ideal situation, the total story points will not change within each iteration (Sprint). If at all any change happens, then it will also get reflected in the chart as shown in Figure 5.17.

The total number of story points to be completed got increased after Day 5 in a cycle of 8 day iteration. The opposite also may happen whereby the total number of story points may also go down.

The burn-up chart shown in Figure 5.18 has a zigzag line indicating some problem throughout the sprint and this kind of graph needs to be examined at the end of the sprint (sprint

retrospective meeting). We need to do a root cause analysis to find out the real cause of the zigzag line. The reason may be a wrong estimation wrong calculation, wrong drawing, etc.

Combined Chart

Combined chart (Figure 5.19) is a combination of both burn-down chart and burn-up charts. Both the charts can be shown in the same graph.

The line which goes down indicates the burn-down graph and the line which goes up indicates the burn-up graph.

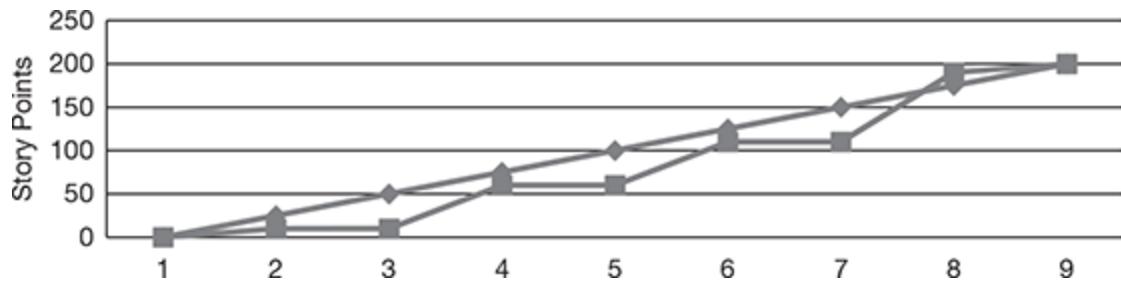


Figure 5.18 Burn-up chart indicating some problem

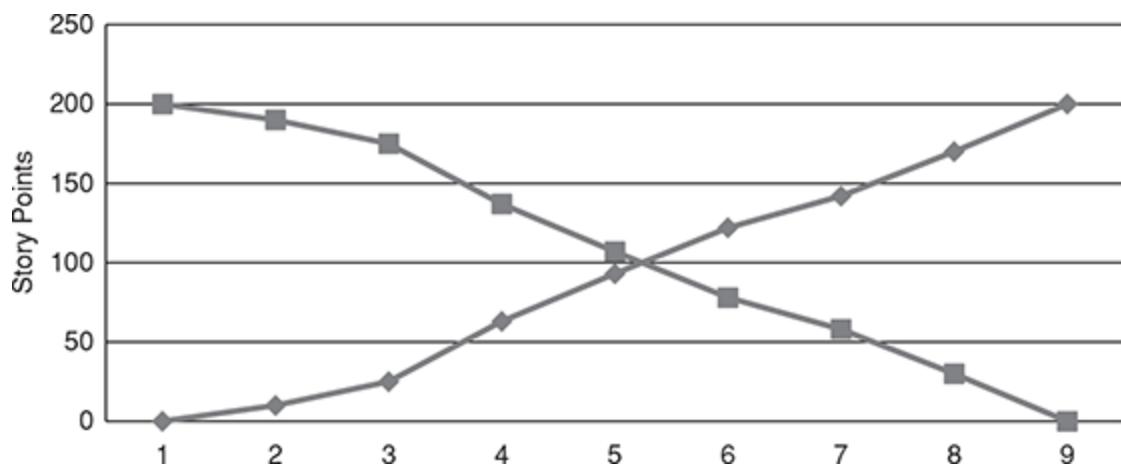


Figure 5.19 Sprint combined chart

Combined chart using the bar chart (Figure 5.20) is a better indicator of how much we have progressed and how much we are about to move ahead.

Cumulative Flow Diagram (Cumulative Chart)

Cumulative flow diagram (Cumulative Chart) as depicted in Figure 5.21 helps to understand the overall status of the items completed, finished and also in WIP items. It is a better means to find out the status of the project.

Lead Time

Calculation of Lead Time

Lead Time is the time required to deliver a given amount of work WIP is work in process - items started but not completed.

ACR is average completion rate.

Lead Time = Work In Progress (units) / Average Completion Rate (units per time period)

Work In Progress = Lead Time* Average Completion Rate (units)

Average Completion Rate = Work In Progress / Lead Time

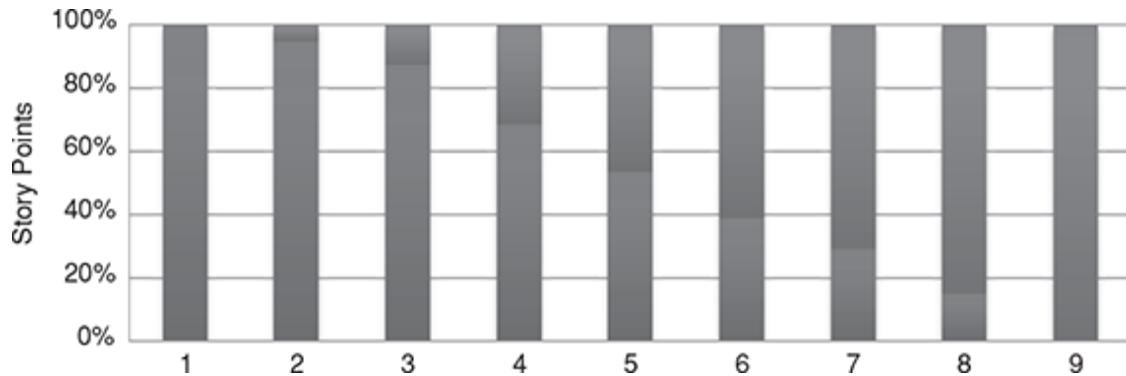


Figure 5.20 Sprint combined chart using bar graph

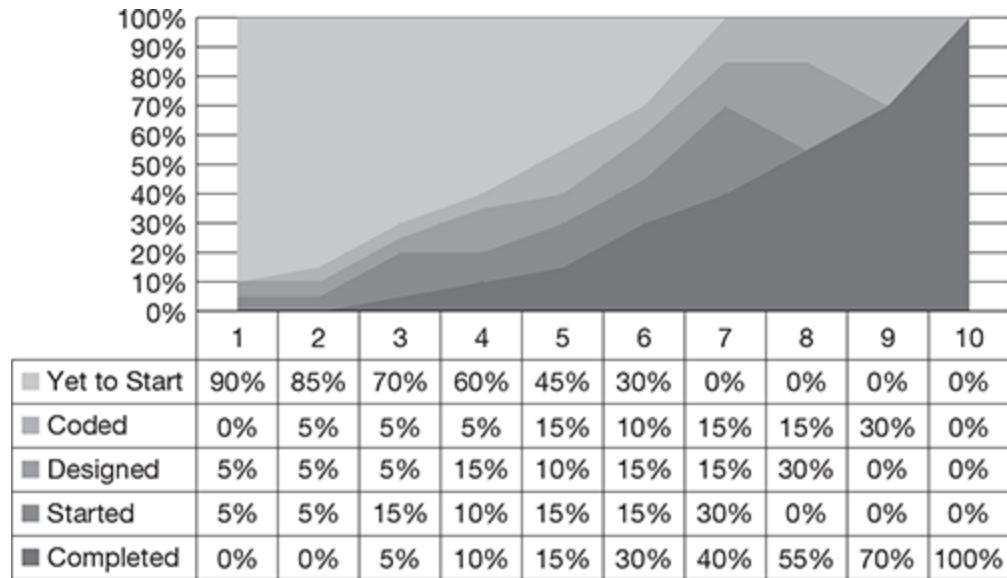


Figure 5.21 Cumulative flow diagram

Earned Value of Agile Projects

Basic concepts of Earned value of Agile project is very much similar to a traditional project but here no work is getting carried forward from one iteration to the next iteration cycle and the whole calculation is based on story points planned and story points completed in that iteration (sprint).

While burn charts provide the visual representation of the work remaining and the project schedule information, it does not give any information regarding the performance with respect to the project cost. For effective tracking, it is very important to understand how the project is doing in terms of the actual cost spent against the budgeted cost. The following three data points are required for establishing the initial baseline in Agile projects, which will be used in EVM calculation:

- Number of planned iterations in a release
- Total number of planned story points in a release
- Planned budget for the release

At any point in time during execution, the following measurements are required for

calculating the EVM:

- Total story points completed
- Number of iterations completed
- Total actual cost
- Total story points added to or removed from the release plan

Earned Value (EV) measures the value of work performed against the planned value for a project till the calculation date. In Agile projects, the story point is the basis of measurement for the value of work.

Let us take the example of the sample product backlog with the actual cost below to understand the EVM calculation in Agile projects:

Total project budget: \$1,75,000

Total number of iterations planned: 4

Completed iteration: 1

Feature	Estimate (story points)	Completed (story points)	Actual cost (USD)
Login screen	10	10	15
Retrieve personal information	20	20	30
Update personal information	10	10	20
Send e-mail notification	20		
....	20		
...	10		
Total	200	40	65,000

- As number of planned iterations = 4 and the number of completed iterations = 1, the expected percent complete = $\frac{1}{4} = 25\%$.
- Planned value (PV) for a given iteration = expected percent complete \times the total budget ($25\% \text{ of } \$1,75,000 = \$43,750$).
- Actual percent complete = total number of story points completed/total number of story points planned ($40/200 = 20\%$ complete).
- Earned value (EV) = actual percent complete \times total budget ($20\% \text{ of } \$1,75,000 = \$35,000$).

In this example, EV \$35,000 is less than PV \$43,750, which means that we were supposed to complete \$43,750 worth of work till the calculation date but we could complete only \$35,000 worth of work, implying that the project is in trouble.

The **Cost Performance Index (CPI)** shows how efficiently we are actually spending your budget compared to how efficiently we planned to spend it.

$\text{CPI} = \text{earned value/actual cost} = 35,000/65,000 = 0.53$. This means we will need more money to complete the project than initially planned, and the revised budget now stands at = initial budget/current CPI = $\$1,75,000/0.53 = \$330,188$.

So, $\text{CPI} = 1$ means on budget, $\text{CPI} < 1$ means over budget and $\text{CPI} > 1$ means under budget.

Similarly, the **Scheduled Performance Index (SPI)** compares the earned value with planned value and denotes if the project will be completed on time.

$\text{SPI} = \text{earned value / planned value} = 35,000 / 43,750 = 0.8$. This signifies that the project is 20% behind schedule and thus it will need total $4/0.8 = 5$ iterations to complete the job that we initially planned for 4 iterations.

The baselines for our calculation above can get changed in case there are user stories added to or removed from the product backlog. So, the EVM calculation for an Agile project can take into account the scope changes, which are inherent in an Agile project.

Task/Kanban Board

Task board is generally called Kanban board. Kanban board is used in Kanban methodology but it can also be used in other methodologies of Agile. It actually displays the status of various works under different stages at a point of time. A look at the Kanban board indicates how much we progressed and where we are, how much work needs to be done further. There are different types of Kanban boards, namely, sprint Kanban, release Kanban and task Kanban. Sprint Kanban shows various tasks of the sprint which are at different stages. Task Kanban is being updated as soon as the work is done. It always reflects the current status of the project.

Kanban board also gives the visibility to the team about how much tasks are pending on hand

at the current sprint. Team members help each other to move the status up so that finally the team performs well.

Kanban Board: It is named based on ‘Just-In-Time’ production method used in the Toyota production system. Figure 5.22 depicts a Kanban board.

Kanban means tickets to do or tasks to do. We can also draw an epic Kanban, feature Kanban, story Kanban and task Kanban.

Task Kanban Board: It shows the exact current status of the tasks within the iteration and in Figure 5.23, the tasks are represented by cards. Various tasks status used in the above chart are labeled as tasks queue, tasks in WIP and task complete. Some people use this as tasks in the design stage, coding stage and testing stage. It is up to the team to label it appropriately and thereby they understand the progress of what is next. It is indirectly self-directing the team.

Stories	Tasks Queue	WIP	Tasks Complete	Stories Complete
				

Figure 5.22 Kanban board

Features	Stories Queue	WIP	Stories Complete	Features Complete
				

Figure 5.23 Story Kanban board

Epics	Feature Queue	WIP	Feature Complete	Epics Complete
				

Figure 5.24 Feature Kanban board

Story Kanban Board: In this type of board, business stories are focused rather than the tasks, and this shows the flow of stories release-wise and it helps the business users, product owners and customers.

Feature Kanban Board: In a feature Kanban board, (Figure 5.24) the feature is of importance and shows the flow of features rather than the stories. This is usually being used by the high level teams including the marketing department. It may not be used to track the project but gives an overall idea of the progression of the project.

Main Features of the Kanban Board

As shown in Figure 5.25, Kanban boards organize three crucial parameters for an Agile project, namely, time, task and team (3 Ts of Agile project).

Kanban boards can be drawn at the release level, iterations level and days level for understanding the timescale.

Kanban boards can be drawn at the feature level, stories level and tasks level for understanding task scale.

WIP Limits

A big advantage of a Kanban board is that it makes bottlenecks easily visible. If story cards get piled up in one of the columns of your board, you know that either the next steps are not able to consume work as efficiently as you planned or it may mean that you are trying to do more than you are capable of. The possible solution in such a situation may be to do the following:

1. Stop working on other streams and focus solely on moving the stalled stories out of the bottleneck
2. Put a work in progress limit (WIP limit) in place for that column so that any bottleneck can be avoided in the future

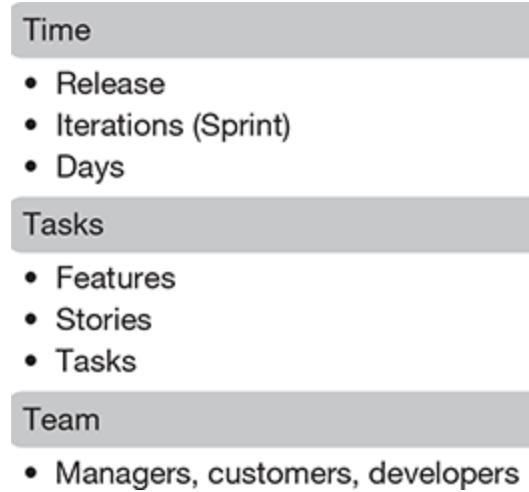


Figure 5.25 Kanban board features

A Work in Progress (WIP) Limit Helps You to Be Focused

It will help you focus on exactly few important tasks if you set your WIP limit to two or three for tasks being ‘in progress’. As you will be starting new stuff only when these tasks are completed, you have an explicit motivation for getting those tasks done which will empty a space on your Kanban board for another ‘in progress’ task.

It does not help to start new stuff if you already know that new tasks will ultimately get stuck in a bottleneck. Starting more and more

new tasks without finishing old ones will only lead to a situation where the team's time is completely consumed by task switching instead of getting anything done. This is where the WIP limit comes in.

Retrospectives

Agile is not a prescriptive methodology but rather it is a framework that should be adapted according to the given project, team and specific circumstances. The iteration retrospective is an important mechanism that allows a team to continuously evolve and improve the life of a project.

Key elements of the retrospective meeting are as follows:

1. Process improvements are made at the end of every iteration, which ensures that the project team is always in self-improving mode.
2. The retrospective is a collaborative process between all members of the team.
3. All team members identify what went well and what can be improved.
4. The actions and lessons learned are prioritized based on the team's direction.

5. The team works out a solution to most prominent problems; helps to build the team ownership and self-management.
6. Helps the team formation and bonding as the areas of conflict can be identified and dealt with.
7. Typically 15–30 minutes.

Figures 5.26 to 5.31 will give you guidance to structure your retrospective effectively:

As you are about to start the most important meeting in an Agile project iteration cycle, make the preparation well so that the retrospective meeting achieves its purpose. Start with setting the working agreement and the ground rules for the meeting if this is your first retrospection or there are new members of the team. A good example of a ground rule may be, ‘No negative criticism is allowed’. Try to gather as many data as possible regarding the last iteration as that will be the basis of your discussion on what went well and what can be improved. The team should get involved in analysing why few things did not go in the way those were supposed to and how these can be corrected. This will generate the action plan for the next iterations and also uncover the impediments that need to be

overcome to make the iterations more efficient. At the end do not forget to appreciate the input and participation from the team members.

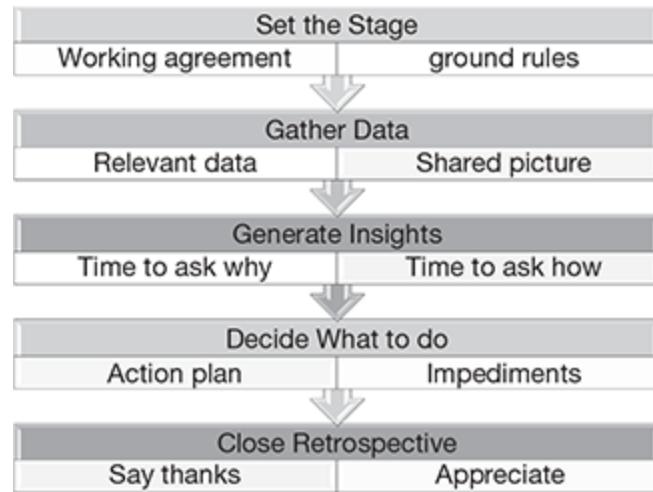


Figure 5.26 Structure of retrospection meeting

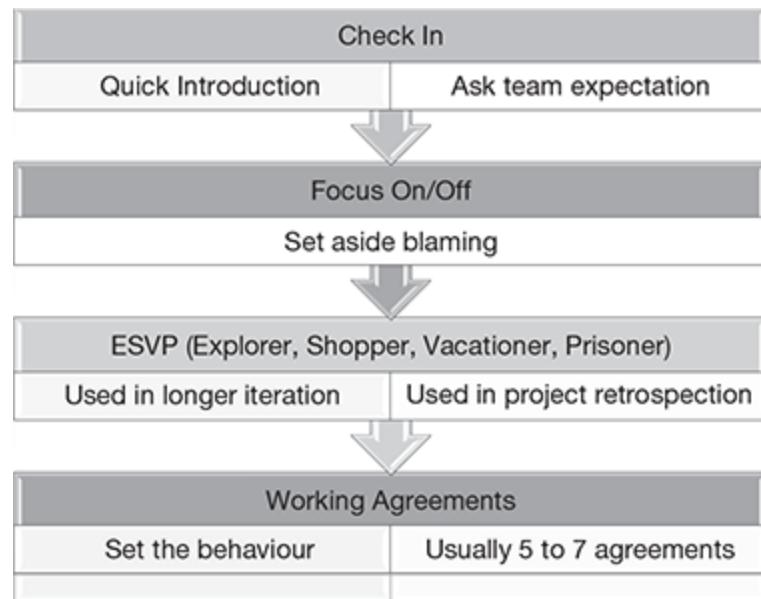


Figure 5.27 Activities to set the stage

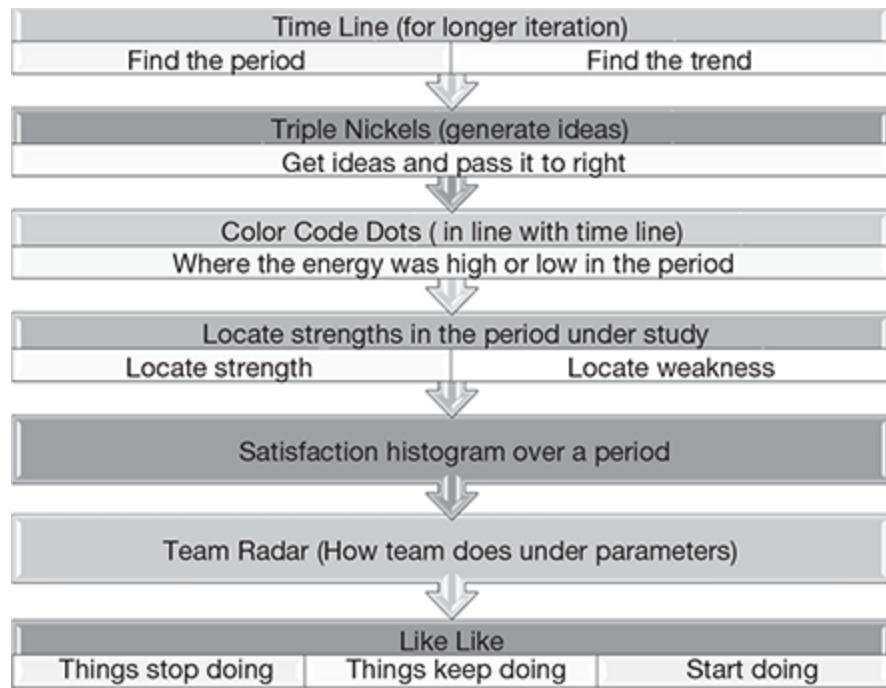


Figure 5.28 Activities to gather the stage

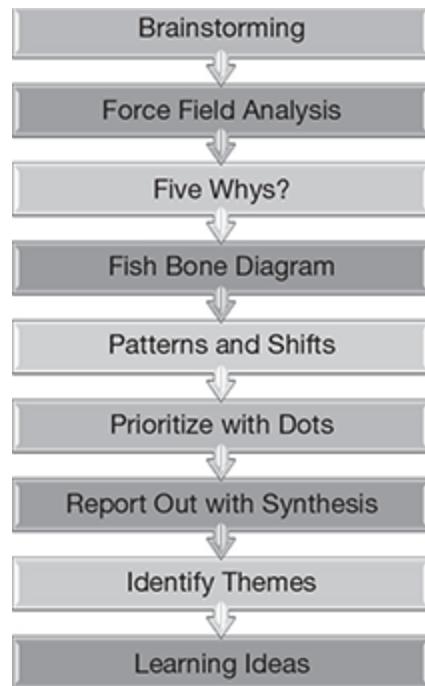


Figure 5.29 Activities to generate insights

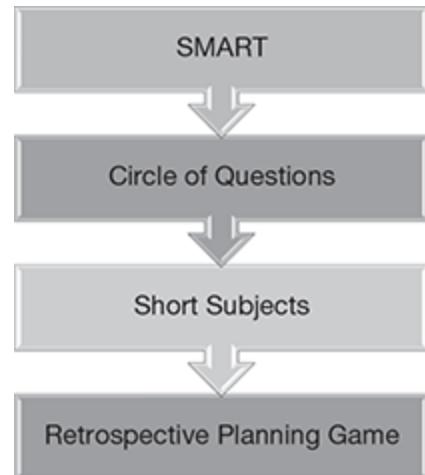


Figure 5.30 Activities to decide

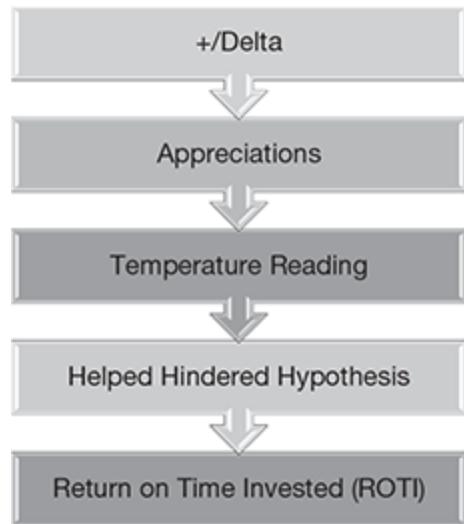


Figure 5.31 Activities to close the retrospective



Figure 5.32 Color coding the feelings

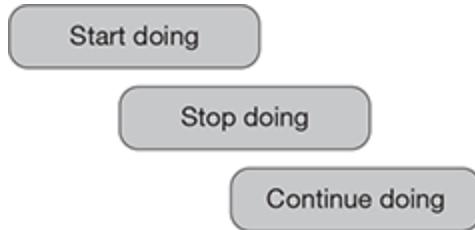


Figure 5.33

Based on your discussion with the team and their participation level, you will be able to judge the team's motivation level. Take necessary action for motivating the team if you find that due to some reason the motivation level is low.

The whole team gathers and discusses what it would like to.

This is the most important meeting that happens in the duration of Agile iteration. Do not assume that a retrospective is required only if the iteration has a problem, because unless you have this meeting how do you know whether the iteration had problems? This is a must for each iteration where it will be discussed what went well and what can be improved to find out and refine what best suits 'the team'.

AGILE PROJECT ACCOUNTING PRINCIPLES

PMI and accounting principles apply quite well in Agile projects. Let us discuss the accounting meaning of the common agile concepts that we discussed above.

1. **Story:** A story represents the smallest measurable unit of work from the external standpoint. Each story is associated with some unit of business value.
2. **Sprint/Iteration:** This represents the smallest unit of spending before any business value can be gained back. During this period, all of the team members are working for a fixed period of time, usually 1-4 weeks. By working in fixed time-boxes and with regular releases, the teams feed value back to the organization.
3. **Backlog:** This generally represents the total scope of the project known till date. In accounting terms, this represents the anticipatory revenue from the project. As the backlog items get refined over time, the revenue figures get fine-tuned.
4. **Task:** Once the stories are allocated to a sprint, the team breaks them into smaller engineering tasks. All the tasks may not necessarily carry business value.
5. **Burn rate:** In each sprint, the Agile team burns money in chunks. For example, if five team members have an average loaded rate of \$50/hour and work in 4-week sprints, the burn rate for the team is $50 \times 5 \times 40 \times 4 = \$40,000.00/\text{sprint}$. Agile theory says that at the end of any sprint, if necessary, the project can be canceled. This is because the backlog is prioritized in terms of greatest value to least value; and thus, the team should be delivering the pieces with the most value first. In each

sprint, the team is delivering the highest possible business value back to the customer.

6. Estimation of total cost: After the team delivers a couple of sprints resulting in tested chunks of business value back to the organization during each sprint, an average velocity is created. Using the velocity it is possible to estimate how many sprints it will take to complete the backlog (or to meet a certain release point). This will give the organization an idea of the total cost of the project. As with other items of planning, this should be revisited after completion of each sprint.

Innovation Games

The primary focus of the Agile methodology is the people and the team. Thus, it gives very high importance to collaboration and knowledge sharing inside the team. Innovation games are very popular in Agile projects for ensuring team building and team cohesiveness.

Following are examples of how you can reinvigorate retrospectives using innovation games:

The retrospective is one of the most important meetings that an Agile team holds after each cycle. It is expected that the retrospective brings out a lot of value from the collective discussion

of all the stakeholders in the Agile team. But after holding the retrospective for many cycles, Agile teams may reach the point where their retrospectives become ineffective or the team members stop seeing value in the meeting. It is also not uncommon for this to occur with newly formed teams as well if they just have not grasped ‘Inspect and Adapt’. On the other hand, a matured agile team might have lost sight of the need for continuous improvement and thus feel the retrospective to be kind of boring. There are multiple reasons why retrospectives may become ineffective, but here are the three main reasons that stand out in the crowd:

- 1. They have become monotonous:** The team has been asking the same three questions for the last 6 months: ‘What can we start? What should we stop? And, what should we continue?’ Although this format is tried-and-tested as to form an effective retrospective, it brings the sense of boring repetitiveness, especially when it happens every 2 or 4 weeks.
- 2. No actions taken:** ‘That’s a great idea’, ‘We should do that’, ‘We need to make it happen’. These are the phrases often used for items discussed during the retrospectives but many times these are the items that do not really get implemented. In fact, there is a worse situation when after the fourth retrospective performed, those actions and or issues mentioned in the first retrospective are reiterated and appreciated.

3. **Non-participating lot:** The non-participating lot are those in the room who do not speak up unless called upon. Usually, these folks often are very smart, but they will contribute only if the rest of the team is enthusiastic and involve them in conversation. But when most of the members have just given up on sharing their inputs because of the first two reasons (identified above), then the whole team suddenly finds the retrospective meeting to become a silent one.

So, can we think about holding the retrospective in a different form like introducing a game?

A Boat for Retrospective Findings

The concept behind this game is you are pictorially representing the problems and good practices in the project and making it visible to the whole team whether any action is taken on the items or not. The project is represented like a sailboat floating on the ocean. Though the boat is trying to move as fast as it can, there are few anchors that are holding it back. In a project scenario, the boat is the Agile cycle and the team is trying to move it as efficiently as possible, but the inefficient practices are adding weight to the anchor making it move slow and, on the other hand, the more wind to the sail which represents

the good practices within the agile cycle will help the boat to move faster. So, after the retrospective, the team will focus to remove as many anchors as possible and add as many winds to the sail.

Here's how it works (Figure 5.34):

1. You need a whiteboard (or sheet of easel paper), sticky notes and black pens.
2. Draw an ocean, a sailboat with big sails floating on the ocean, and a couple of anchors.
3. Give each team member in the retrospective meeting pens and sticky notes to write down 'what we should stop doing' (or issues) and 'what we should start and continue to do' (or efficiencies) that have been discovered since the last retrospective each point on a separate sticky note. Encourage each team member to put in their thoughts and ideas and let them know that their sticky notes will actually appear on the sailboat. Give everyone 5–10 minutes to write down the points.
4. After everyone is complete, each team member will go to the board and put the sticky notes on the sailboat—the good things represent the wind in the sails and the bad things represent the anchors that slow the boat.
5. While going through the above action, have the team organize the sticky notes into groups of like items (e.g. issues surrounding quality, etc) and eliminate the duplicates and consolidate wherever it is possible. You can also think of ranking the sticky notes in terms of priority which will help you to prepare the action plans better.

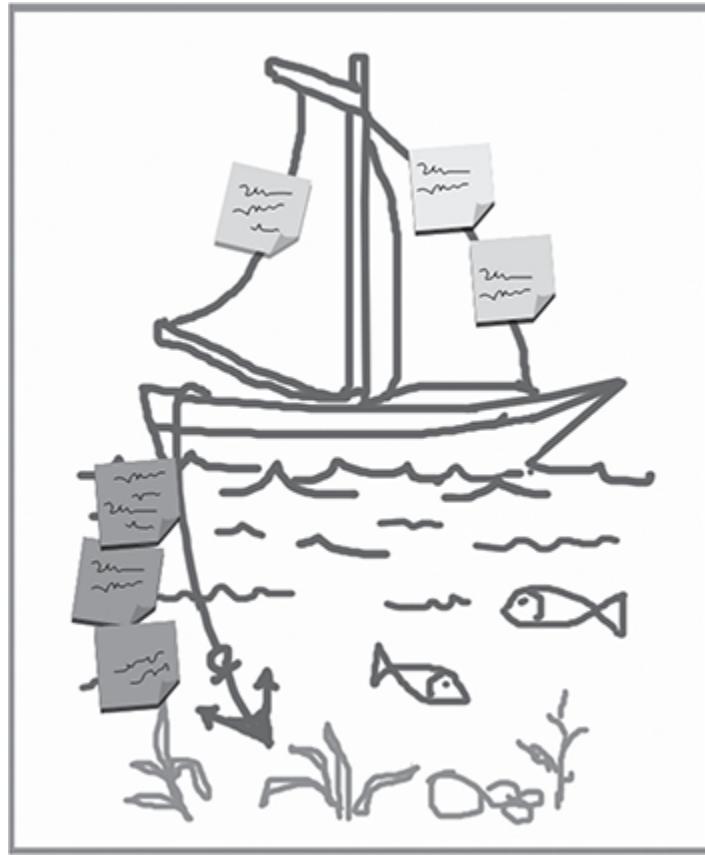


Figure 5.34 A boat for retrospective findings

1. 6. Now the team has a visual display of the retrospective findings that they can use to refine the backlog or plan how some tasks are to be performed. The final aim is to reduce sticky notes from the anchor and add more to the sails so that the boat (that means the project) runs more efficiently.

By adding to the fun part of the retrospective you are making the team members to do away with the monotonous meetings and capturing the retrospective finding innovatively. This is not

only helping your retrospectives to become more efficient but also is helping the team to think and work collaboratively to achieve a project goal.

There are other innovative games also which helps in improving the quality of the retrospectives. For example, you can take the below approach to get a list of what the team members liked, learned, lacked and longed for during the last sprint very quickly. This will also ensure participation from all the team members.

1. You will need four posters (easel paper), sticky notes and black pens.
2. Hang the four posters (easel paper) around the room and title the posters appropriately, one for each ‘L’—liked, learned, lacked and longed for.
3. Ask the team members to individually write down what they liked, learned, lacked and longed for since the last retrospective. They should write down one per sticky note. Give the team 5–10 minutes and ask them to put their sticky notes on the corresponding posters.
4. Divide the team up into four subgroups and assign an ‘L’ to each group for consolidating the notes.
5. The team should work on how the issues found from the above steps can be addressed. Identify issues that need to be worked and stories to be added to the backlog.

The innovation games focus on encouraging full team participation and add **fun** part to the retrospection. No matter which technique you use for your retrospective, a few key rules will always help:

1. **Issue resolution.** If there are issues between individuals, first give them chance to work it out and if necessary, involve a third party.
2. **Involve everyone.** The approaches in the above games help drive this; however, be sure folks are contributing and more importantly they feel their input is important.
3. **Do something about it.** The items identified should be tracked and acted upon; otherwise the team providing the input will get de-motivated.

Process Tailoring

The principle of Agile projects is to continually adapt to the processes that best suits the project based on the retrospective outcomes. That is the reason the topic of process tailoring is very important for Agile projects.

Figure 5.35 describes the steps to be followed for a project-specific process tailoring. Make sure to involve all relevant stakeholders for a successful roll out of the tailored process.

Some typical examples of project-specific process tailoring are as follows:

1. Adding/removing work products and tasks to the process
2. Changing milestones definitions like and what work products will be made available at each milestone and their state of completion
3. Tool and/or format which should be used for each work product

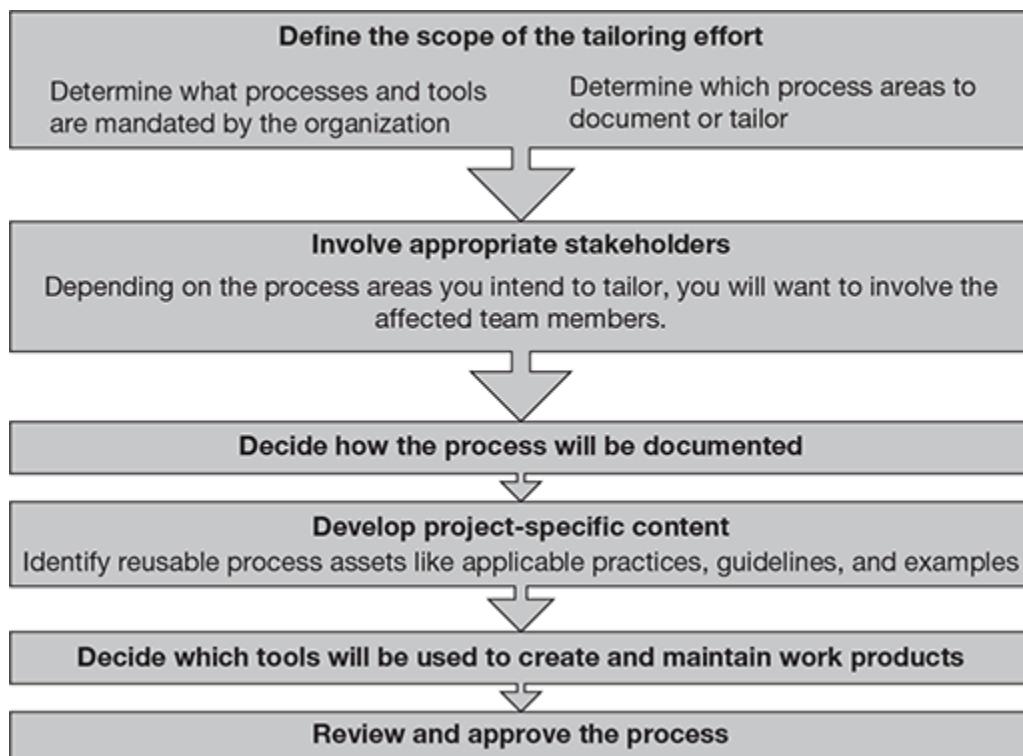


Figure 5.35

1. 5. Responsibilities for review and approval at each process stage

2. 6. Detailing the procedures for reporting progress, performance measurements, managing requirements, managing change requests, etc.

Once the process is tailored for a project it should be base-lined and institutionalized so that all the stakeholder involved start following the process without much ambiguity. This also helps in reusing the tailored process in future by similar projects.

Summary

In traditional project the detailed plan is base-lined before the start of work in the project, but the approach that Agile projects take is more like evolving the plan over time. Also the project monitoring and tracking approach is quite different from the traditional model as more focus is given in making the progress information available to the stakeholder and

making the team self-organized so that they can correct the problems themselves. In this chapter, we discussed the concepts of adaptive planning, progressive elaboration, user story, Agile planning, time-boxing, Agile contracting, Agile project tracking using the information radiators like burn-up chart, burn-down chart, Kanban board, the importance of retrospective and how innovation games can improve the quality of retrospectives.

Chapter 5 Questions and Answers

- Please set yourself a time clock of 1 hour to take this test
- Mark your answers using pencil in the answer sheet provided at the end of this question set
- The correct answers are provided at the end of this question set (after the answer sheet)
- Give one mark for each correct answer for evaluation purpose
- There is no negative marking for the wrong answers
- Practice this test multiple times for better results
- All the very best!

Question 1. Which of the following is the best approach for determining the iteration (time-box) length?

1. Iterations (time-boxes) should always be 4 weeks
2. The team calculates iteration (time-box) length by dividing the total number of story points by the average velocity of the team
3. Ideal iterations (time-boxes) is 2 weeks
4. The team will consider the size and complexity of the project and should agree on the length of the iteration (time-box) through discussion

Question 2. Choose the best reason for maintaining a consistent iteration (time-box) length throughout the project?

1. It helps to establish a consistent pattern of delivery
2. It helps the team to objectively measure progress
3. It provides a consistent means of measuring team velocity
4. All of the above

Question 3. What is the best approach for running an effective retrospective?

1. Presentation
2. Forcing team members to participate
3. Encourage team members to participate
4. Lecture

Question 4. Which of the statements best represents an Agile's approach towards planning?

1. Everything is ad hoc, there is no scope for planning in Agile

2. Planning should be done in detail at the beginning of the project and then should not be revisited
3. Planning is an iterative job and should involve the whole team
4. Planning is done by the project manager and the team members follow the plan

Question 5. Why is the presence of customer representative made mandatory in Agile projects?

1. The customer representative is the main judge for business value
2. The project leader should build rapport with the customer representative
3. The customer representative can dictate the project plan
4. All of the above

Question 6. Your project team was working on a set of prioritized user stories while a change in market condition made it necessary to re-prioritize the backlog before the next sprint is started. Who should get involved in the re-prioritization task?

1. The developers alone
2. The customer alone
3. The project leader
4. The whole team including customer and developers (together they can consider both business value and technical feasibility)

Question 7. During an iteration retrospective the team discussed the plan for the next sprint. The Scrum master is not happy about it because:

1. A retrospective meeting is to discuss what went well and what can be improved from the experience of the last sprint. This should not be used as sprint planning meeting.
2. The Scrum master does not like meetings
3. The Scrum master was unhappy with the quality of the product
4. The team was not working as an ideal team

Question 8. Which statement below is true regarding an Agile approach?

1. Get something as quickly as possible without thinking about quality
2. Get something simple up and running as quickly as possible
3. Get business-values delivered as quickly as possible, consistent with the right level of quality
4. Even half-done items can be delivered and approved

Question 9. A project team is busy in analysing the current organizational processes and making changes to the processes as per project requirements. Which activity are they performing?

1. Creating product backlog
2. Creating release plan
3. Process tailoring
4. Retrospective

Question 10. What is a Kan Ban?

1. A list of activities that should be banned for the team
2. The set of values defined for a project
3. A visible chart of work for the team for tracking purpose
4. A ban on team meetings

Question 11. In a burn-down chart the remaining work line is below the expected work line. What does this signify?

1. The project is ahead of schedule
2. The project is behind schedule
3. The resources are working their best
4. The project is managed well

Question 12. What is the advantage of burn-up chart over the burn-down chart?

1. The burn-up chart shows whether there is any change in the total story points which the burn down chart does not show
2. Burn-up charts are easy to maintain
3. Burn-up charts take less space than burn-down
4. Burn-down charts do not provide scope to track the project

Question 13. What are the points that normally are discussed in a retrospective meeting?

1. What team should stop doing
2. What team should start doing
3. What team should discuss tomorrow

4. What team should continue doing

Question 14. While measuring the EV of an Agile project it was found that the CPI is 0.8. This means the project will

1. Complete in right budget
2. Consume less money than budgeted
3. Need more money than that budgeted
4. More information is required to do the analysis

Question 15. While creating the release plan for a project the team is exploring the basic minimum features of the product so that people can start using it rather than waiting for something else. What is this set of features called?

1. Minimum marketable feature
2. Product backlog
3. Sprint backlog
4. Basic feature list

Question 16. Which of the below is not a part of the 3 Cs of user story?

1. Card
2. Conversation
3. Confirmation
4. Collocation

Question 17. In terms of size what is the ordering of the below items from biggest to smallest?

1. Epic, theme, story
2. Theme, story, epic
3. Story, theme, epic
4. These are not related at all

Question 18. According to Agile philosophy, who should own the development plan?

1. Project Manager
2. Team
3. Product owner
4. Customer

Question 19. Which of the below statement best describes the benefit of WIP limit?

1. Helps in avoiding the piling up of tasks at a particular stage
2. Makes developers freely choose tasks
3. Makes the testing more robust
4. Limits the product backlog size

Question 20. What is the time-boxed duration of daily standup meeting?

1. 1 hour
2. As long as the team wishes
3. 15 minutes

4. Half a day

Question 21. Which of the following is the key feature of an Agile plan?

1. It can be changed easily
2. It should be well-documented
3. The plan for the whole project should be made upfront
4. No planning required for Agile projects

Question 22. In a user story card where are the acceptance criteria documented?

1. At the front of the card
2. At the back of the card
3. In a separate document
4. This is not important information to capture

Question 23. What is the importance of mentioning the priority within a user story card?

1. These are just some serial numbers
2. The project manager chooses the priority of the stories
3. Priority measures the relative cost of the stories
4. The higher priority user stories are developed earlier

Question 24. Which approach is best suited for Agile planning?

1. Adaptive approach
2. Predictive approach

- 3. Definitive planning
- 4. Ad hoc planning

Question 25. During the iteration planning who creates the task list?

- 1. Scrum master
- 2. Product owner
- 3. The team
- 4. Project manager

Question 26. Which of the below statements is not true about the Agile estimation?

- 1. Estimation should be done by one individual
- 2. Estimate should be done by the team
- 3. Estimation is done as a group
- 4. Estimations are detailed out with time

Question 27. Which of the below is not a valid feature of the Agile planning?

- 1. Adapt your plan
- 2. Keeping your options open
- 3. Empower the team to take part
- 4. Make as detailed plan as possible at the beginning

Question 28. The X-axis of a Sprint burn-down chart represents which of the following?

- 1. Days in the iteration

2. Story points remaining
3. Days to the next release
4. Total estimation of the sprint

Question 29. While analysing a burn-up chart, you found that the total story point line has gone up on the 5th day of the iteration. Which might be the reason for this?

1. The graph was plotted wrongly
2. New user story added to the sprint increasing the total story point of the sprint
3. Some user stories have been removed from the sprint
4. Team is working harder to achieve more tasks

Question 30. You have an eight-member team which have an average loaded rate of \$50/hour and working in 2 week sprints. What is the burn rate for the team per sprint?

1. \$12000.00 per sprint
2. \$32000.00 per sprint
3. \$4000.00 per sprint
4. \$16000.00 per sprint

Question 31. Which of the following best describes the basic principle behind adaptive planning?

1. Planning too much in the future is a waste of effort as the future may change
2. Planning is a waste of effort
3. Team should do whatever they want, no planning is necessary
4. Plan the entire project at the beginning

Question 32. Which of the following is the most important feature of a user story?

1. Less information
2. More story than the feature
3. Customer focused
4. Written on a card

Question 33. Due to what reason it is a good practice to add a unique story numbers to each user stories?

1. To create some work for the team
2. Product owner has instructed to do so
3. For easy estimation
4. It ensures traceability so that nothing gets missed out during implementation

Question 34. Which of the statement below best describes the nature of a scope-boxed release plan?

1. The work what the team will do is defined in advance, but the release date is uncertain

2. The release date is fixed in advance, but the content of each release is uncertain
3. Both the content and the timeline for release is kept uncertain
4. The plan is provided by the customer

Question 35: An Agile project team is creating a release plan. They first defined the release dates but kept the decision of the specific work people will be doing pending for later. What type of release plan are they creating?

1. Time-boxed release plan
2. Scope-boxed release plan
3. Uncertain release plan
4. Adaptive release plan

Question 36. In Agile projects the task level breakup of the user stories is done by the team. What is one of the main benefits of this being done as a team rather than individually?

1. It keeps the team busy
2. This sometimes brings out the perspectives that a single developer or tester would have missed during the actual software development.
3. This ensures that the product owner knows each team member
4. This should have been done by individuals

Question 37. Agile principle strongly recommends daily stand-up meeting as this helps

in

1. Surfacing the problems early
2. Keeping the team together
3. Making the team disciplined
4. As status updation for the project manager

Question 38. What is the role of the Scrum master in the daily stand-up meeting?

1. Surfacing the problems early
2. Taking note and resolving the impediments faced by the team
3. Clarifying business requirements
4. Updating the project manager

Question 39. The product owner may attend the daily stand-up meeting if

1. If product owner is free
2. If the project manager requests for the same
3. If the team is not matured
4. It is being requested by the team and there is a necessity for the same

Question 40. An Agile coach asked a newly formed Agile team to focus on Just in Time (JIT) planning for a project the team is going to execute in Agile methodology. What is the main reason for the Agile coach to support JIT planning for Agile projects?

1. It takes less effort to create a Just in Time (JIT) plan
2. The team will be under constant pressure in Just in Time (JIT) plan
3. Just in Time (JIT) planning ensures no re-work in the plan and also helps to accommodate changes in the system
4. This is just another type of planning and has no extra benefit

Question 41. A customer wants to get his next project executed in Agile methodology. Your company is bidding for this project. You came to know that the customer wants to do it in fixed price contract only. What should be your approach for this contract?

1. Add additional conditions in the contract
2. Do not bid for the project
3. Quote very high price to mitigate the risk
4. Fixed price estimates are best suited for Agile projects

Question 42. What is the most suitable contract type for an Agile project?

1. Fixed price
2. Time and material
3. Fixed time
4. Fixed cost

Question 43. An Agile team is working in a workspace where the big information radiators

are kept on the walls. What is this type of workspace called?

1. Closed workspace
2. Agile workspace
3. Ideal workspace
4. Informative workspace

Question 44. What does the cone of uncertainty theory specify?

1. There will be lots of uncertainty at the beginning of the project which gets reduced as we progress
2. The project risk increases as we progress on the project
3. The project cost estimate should look like a cone
4. The uncertainty in a project should remain constant

Question 45. Feature Kanban board is used by

1. The daily Scrum meeting
2. The Scrum master
3. High level team including marketing
4. The project team

Question 46. Which step in team decision-making process generates feedback for team improvement?

1. The daily Scrum meeting
2. Retrospection

- 3. Release planning
- 4. Iteration demo

Question 47. What is the method of addressing the issue of too many WIP items?

- 1. Burn-down chart
- 2. Burn-up chart
- 3. WIP limits in a Kanban system
- 4. Feature Kanban board

Question 48. Which of the following ensures team coordination and collaboration through information exchange?

- 1. Burn charts
- 2. Daily stand-up
- 3. Product backlog
- 4. Brainstorming

Question 49. In a retrospective meeting, the team has raised few issues. Who is responsible to device the solution for these issues?

- 1. The team
- 2. Project manager
- 3. Scrum master
- 4. Customer

Question 50. Which of the below is used to ensure collaboration in a fun team environment?

1. Daily stand-up
2. Planning poker
3. Innovation games
4. Retrospectives

Answer Sheet for Chapter 5 Questions

Question Number	Answer	Question Number	Answer
-----------------	--------	-----------------	--------

Question 1	Question 26		
Question 2	Question 27		
Question 3	Question 28		
Question 4	Question 29		
Question 5	Question 30		
Question 6	Question 31		
Question 7	Question 32		
Question 8	Question 33		
Question 9	Question 34		
Question 10	Question 35		
Question 11	Question 36		
Question 12	Question 37		
Question 13	Question 38		
Question 14	Question 39		
Question 15	Question 40		
Question 16	Question 41		
Question 17	Question 42		
Question 18	Question 43		
Question 19	Question 44		
Question 20	Question 45		
Question 21	Question 46		

Question 22	Question 47
Question 23	Question 48
Question 24	Question 49
Question 25	Question 50

Answers for Chapter 5 Questions

Question Number	Answer	Question Number	Answer
-----------------	--------	-----------------	--------

Question 1	D	Question 26	A
Question 2	D	Question 27	D
Question 3	C	Question 28	A
Question 4	C	Question 29	B
Question 5	A	Question 30	B
Question 6	D	Question 31	A
Question 7	A	Question 32	C
Question 8	C	Question 33	D
Question 9	C	Question 34	A
Question 10	C	Question 35	A
Question 11	A	Question 36	B
Question 12	A	Question 37	A
Question 13	C	Question 38	B
Question 14	C	Question 39	D
Question 15	A	Question 40	C
Question 16	D	Question 41	A
Question 17	A	Question 42	B
Question 18	B	Question 43	D
Question 19	A	Question 44	A
Question 20	C	Question 45	A
Question 21	A	Question 46	B

Question 22	B	Question 47	C
Question 23	D	Question 48	B
Question 24	A	Question 49	A
Question 25	C	Question 50	C

Explanations for Chapter 5 Answers

1. **Answer D**

In Agile methodology, the team is encouraged to take decisions on the time-boxes themselves and commit to that.

2. **Answer D**

Consistent iteration time-box helps the team to plan their activities better as they will be clear in their mind regarding the timelines.

3. **Answer C**

An effective retrospection is possible only if all the team members take active part in sharing their views.

4. **Answer C**

Agile planning is done iteratively and that is why it is called adaptive planning.

5. **Answer A**

The role of the product owner [Customer Representative] is to provide input on the business value for effective prioritization of the requirements.

6. **Answer D**

7. **Answer A**

8. **Answer C**

9. **Answer C**

10. **Answer C**

11. **Answer A**

The remaining work line goes below the expected line when the team is able to complete more than planned, means ahead of schedule.

12. **Answer A**

13. **Answer C**

14. **Answer C**

15. **Answer A**

16. **Answer D**

17. **Answer A**

18. **Answer B**

Explanation: in Agile methodology the team comes up with the plan and owns it.

19. **Answer A**

20. **Answer C**

21. **Answer A**

22. **Answer B**

23. **Answer D**

24. **Answer A**

25. **Answer C**

26. **Answer A**

27. **Answer D**

28. **Answer A**

29. **Answer B**

30. **Answer B**

[$8 \times \$50 \times 40 \times 2 = \32000.00]

31. **Answer A**

32. **Answer C**

33. **Answer D**

34. **Answer A**

35. **Answer A**

36. **Answer B**

37. **Answer A**

38. **Answer B**

39. **Answer D**

40. **Answer C**

41. **Answer A**

42. **Answer B**

43. **Answer D**

44. **Answer A**

45. **Answer A**

46. **Answer B**

47. **Answer C**

48. **Answer B**

49. **Answer A**

50. **Answer C**

Key Terms

Adaptive planning/rolling wave

planning/progressive elaboration A multi-step, intermittent process of planning which is based on the philosophy that the plan can be detailed out only as more details are revealed with time and with the changing scenario in the project.

User story A user story describes what the user does with the software and how the software responds. A user story is a functional requirement that resembles a use case and test case.

Time-boxing It is a fixed timeframe within which the team tries to achieve the planned tasks

for the iteration and stops the iteration as soon as the timeline is over.

Burn charts The pictorial representation of the remaining work for an iteration or release which can be used to predict whether all the tasks can be completed for the iteration with the current pace.

Task/Kanban board It displays the status of various works under different stages at a point of time. A look at the Kanban board indicates how much we progressed and where we are, how much work needs to be done further.

Retrospectives The iteration retrospective is an important mechanism that allows a team to continuously evolve and improve through the life of a project.

Process tailoring Changing the existing organization processes to best fit the project scenario is called process tailoring.

6

Agile Metrics and Estimations

KNOWLEDGE AND SKILLS LEVEL

Tools and Techniques: High

- *Relative Sizing*
- *Story Points*
- *Wideband Delphi*
- *Planning Poker*
- *Affinity Diagram*
- *Ideal Time*
- *Velocity*
- *Cycle Time*
- *EVM*
- *Escaped Defects*

Level 1: High

- *Time*

- Budget
- Cost Estimation

Level 3: Low

- Variance and Trend Analysis

INTRODUCTION

Agile project evolves progressively and so does the estimate of an Agile project. During release planning, the epics are being estimated. Of course, the epics are the high level user stories and so they are estimated at a high level, with relative complexity measurement that uses the qualitative technique. Quantitative technique may not be possible at the high level.

But as we go deeper into the planning during iteration ...

Agile Analysis and Design

KNOWLEDGE AND SKILLS LEVEL

Tools and Techniques: High

- *Product roadmap*
- *User stories/backlog*
- *Story maps*
- *Progressive elaboration*
- *Definition of Done*
- *Agile modeling*
- *Wireframes*
- *Chartering*
- *Personas*
- *Workshops*
- *Learning cycle*
- *Collaboration games*
- *Backlog refinement*

Level 2: Medium

- *Participatory Decision Model*
- *Business Case Development*

INTRODUCTION

Agile analysis and design start with creating the roadmap for the product, which is often a high-level view of how the sponsor or the customer wants the features to be released. Based on this, the requirements are elaborated further to come to a point when all the doubts are clarified and it is ready for implementation. Successful gathering of requirements is a key to the success of any project. Unlike the traditional model where the requirements are gathered at the beginning of the project and tried to be kept frozen throughout the project life cycle, the Agile framework allows the flexibility to progressively elaborate the requirements throughout the life cycle. That is the reason why in Agile projects the different levels of requirement details are captured in the number of artifacts like Product roadmap, Product backlog, Release backlog and Sprint backlog as will be discussed further in this chapter. The different

techniques used for gathering the requirements are very important to know and that will be the starting point of the requirement analysis discussion.

COLLECTING AND VALIDATING REQUIREMENTS OF AGILE PROJECTS

Based on the project complexity, availability of the stakeholders, knowledge level of the end-users and the client, and the proficiency of the requirement analyst, the most suitable requirement gathering technique needs to be selected. The different requirement collection and validation techniques explained below are preferred for Agile methodology because all of these encourage and facilitate the direct interaction between the stakeholders of the project.

Interviews

Begin the interview with stakeholders who are believed to have a complete understanding of the requirements. Face-to-face contact with users through individual interviewing is the primary

source of requirements and an important way to gather and validate their requirements. Interview techniques are used when the requirements are detailed and differing opinions are likely sought. In the distributed Agile, the interview is the main technique used to gather requirements. Even if the team and the customer are collocated, we can use the interview technique to clarify doubts. During the execution of the project at any point of time interview may happen with the product owner to clarify the doubts.

Causes/Solutions

Write the problem that has been identified on the flip chart and then divide the paper into two halves, one half headed Causes and the other Solutions. Get team members to contribute to both columns. Each contribution should be discussed and agreed before being added to either column.

Future Visioning

Ask participants to imagine they are at some point in the future (e.g., 2010). They should

imagine that all their problems are resolved, dreams have been achieved, and goals have been achieved. Now step back and describe how we did it.

It is important to spend some time on placing the participants in the future. Use the past tense when asking how we did it. This technique can be useful when problems seem insurmountable or when the group has low morale.

Mind Map

Facilitate discussion with a group and record the output in a large mind map drawn for all participants to see—on a flip chart or large sheet of paper pinned to the wall. Mind maps are an expression of radiant thinking, a natural function of the human brain, and they allow creative ideas to bloom and flow. They provide a basic ordering of information and easily allow later additions as the session progresses.

Greenfield Site

Ask participants to imagine the problem and then imagine there were no history, rules, regulations, culture, or climate. If none of these things existed because we were just starting up what might we do, how might we approach the solution.

Celebrity Views

Split the group into smaller sub-groups. Each sub-group decides upon a celebrity (dead or alive, fact, or fiction). Each sub-group then explores the characteristics of their celebrity and decides how the celebrity would view the problem at hand. Each group then presents their solutions to the plenary. This method can be used in a variety of different ways, i.e., participants can imagine what someone they admire might do, etc. It is a useful technique for viewing a problem and solutions from a fresh angle.

Big Picture

Another way to take a fresh approach to solution finding is to facilitate discussion in the group and record the output as a big picture for all

participants to see. This provides a graphic representation of the problem and enables participants to visualize solutions, stimulating further ideas. This will appeal to participants for whom the visual channel is important and generate fresh ideas. It is also useful for building ownership.

Focus Groups

Focus groups are basically interviews, but it is more focused on the number of participants is restricted to 6 to 10 in one group at one point in time. We can get a lot of information during a focus group session.

Types of Focus Groups

1. **Two-way focus group**—one focus group watches another focus group and discusses the observed interactions and conclusions.
2. **Dual moderator focus group Model 1**—one moderator ensures the session progresses smoothly, while another ensures that all the topics are covered. This will ensure smooth progress of the session along with the entire topics of discussion.
3. **Dual moderator focus group Model 2**—two moderators deliberately take opposite sides on the issue under discussion. This is more helpful to find the pros and cons

of both the opposite views and will be helpful to take a final decision.

4. **Respondent moderator focus group**—Respondents are asked to act as the moderator temporarily and because of this, they take ownership of the session and hence we can collect the requirements successfully and the outcome will be helpful.
5. **Client participant focus groups**—Client representatives participate in the discussion and will ensure ownership from the client on the decision taken. Most of the projects fail at the last stage of the project lifecycle particularly in user acceptance testing phase. The reason being the client may not take the ownership of the initial requirements as the end-users have more ownership in it because they took the decision regarding the system in the beginning. But client participant focus group helps to overcome these kinds of problems.
6. **Mini focus groups**—Groups are restricted to four or five members rather than 8 to 12 and are helpful to avoid confusion; it will be easy for us to manage and gather the requirements quickly.
7. **Teleconference focus groups**—telephone network is used.
8. **Online focus groups**—computers connected via the internet are used.

Facilitated Workshops

Workshops can be used for rapidly pulling together a good set of requirements. Workshops are very quick and best way to gather requirements than all other techniques.

A workshop is expensive because it involves many people, but it saves a large amount of time for us.

Here we use the brainstorming technique to gather requirements from the people.

Requirements are discussed at the high level. Workshops are being conducted particularly when the requirements are focused on one area of business in which the participants have knowledge, and consensus is being sought.

Learning Cycle

Learning curve: This term refers to the decrease in time taken to perform a task because of the increase in efficiency in performing the task that results solely out of repeatedly performing the task. Therefore, with the passage of time, although total costs typically increase, the cost per unit drops. This pattern of cost reduction is given by

$$Y_x = (K_x) \log_2 b$$

where K is the number of direct work hours required to produce the first unit, Y_x is the number of direct work hours to produce the x th unit, x is the number of units produced and b is the learning percentage.

PARTICIPATORY DECISION MODEL

Agile projects encourage decision-making with the full participation of the relevant groups. In these groups, members meet face-to-face and rely on both the verbal and nonverbal interactions and brainstorming to communicate with each other.

Brainstorming is meant mainly to overcome pressures for conformity in the interacting group that retard the development of creative alternatives.

It does this by utilizing an idea-generation process that specifically encourages any and all alternatives, while withholding any criticism of those alternatives.

In a typical brainstorming session, about six people sit around a table and the group leader states the problem in a clear and understandable manner so that it is understood by all the participants involved. Members then can suggest as many alternatives as they can in a given length of time. No criticism is allowed strictly, and all the alternatives are recorded for later discussion and analysis. Brainstorming, however, is merely a process to generate ideas.

The Nominal Group Technique usually restricts discussion or interpersonal immunization during the decision-making process, hence, the term nominal. The members are all physically present but all members operate independently. A problem is presented and members do the following steps.

1. Members meet as a group and each member independently writes down his or her ideas on the problem before the discussion.
2. Each member presents at least one idea to the group. No discussion takes place until all ideas have been recorded by all the people.
3. The group now discusses the ideas in detail for clarity and evaluates each one of them with its pros and cons.

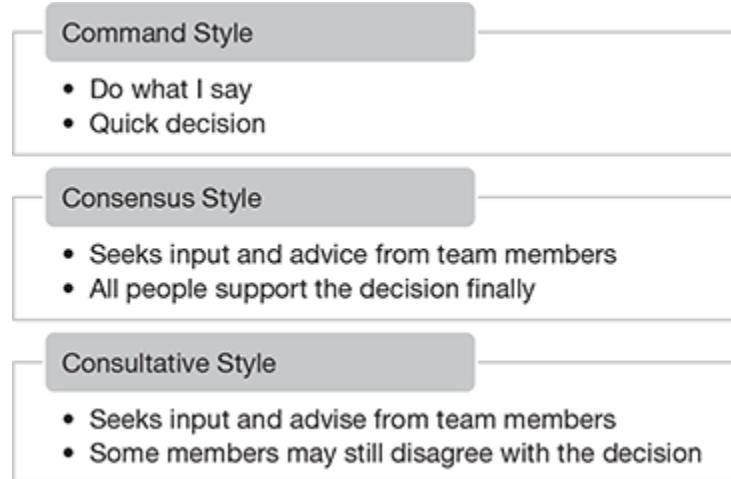
4. Each group member independently rank orders the ideas.
The ideas with the highest aggregate ranking go for the final decision.

The main advantage of the nominal group technique is that it permits the group to meet formally but does not restrict independent thinking.

The interacting group is good for building group cohesiveness, brainstorming keeps social pressures to a minimum, the nominal group technique is very cheap for generating a large number of ideas, and electronic meetings process ideas fast.

Styles of Group Decision-making

The figure below represents different styles of group decision making.



Command Style

Here the leader makes a decision for a group with little or no input from the members of that group. The group members may provide specific information on request, but are not asked to contribute toward finding the final solution. This is like an autocratic decision. The advantage with the 'command' style is that the decision is made quickly. The disadvantage of the 'command' style is that although group members may have been consulted, some or all of them may still disagree with the decision that has been made, and therefore may have little or no commitment to that decision.

We can use the ‘command’ style of group decision making when any of the followings applies:

1. the group perceives you to be the ‘expert’,
2. the group asks you to decide,
3. there is not enough time to use a consultative or consensus style of decision making (e.g., in a ‘crisis’ situation).

Consultative Style

In this case, the leader looks for input and advice from the group before making a final decision. The decision will meet the needs of the group. Group members have been consulted but some or all of them may still disagree with the decision that has been made.

We can use the ‘consultative’ style of decision making when any of the following applies.

1. The group does not have the information, education, skills or experience to make a high quality decision.
2. The group does not share the same goals or objectives that you hope to achieve by solving the problem.

Consensus Style

The ‘consensus’ style of group decision making refers to a situation where the leader seeks input and advice from the group, and works through the decision-making process with the group, until every member of the group can ‘live with’ the final decision that is made. We can call this a democratic solution as everybody takes part in the final decision. It leads to a high degree of ‘buy-in’ and commitment to the decision by all members and they are actively involved in the decision-making process and all of their interests are addressed, but it takes longer time as gathering a group of people is very difficult.

We can use the ‘consensus’ style of decision making when any of the following applies

1. The group shares the same goals or objectives.
2. All supports the decision reached irrespective of their own opinion.
3. Plenty of time available to take a decision.

There are multiple methods of reaching a group decision.

Unanimity	Everyone agrees on a single course of action
Majority	Support from more than 50%
Plurality	The largest block in a group decides even if a majority is not achieved
Dictatorship	One individual makes decision for the group

Thumb voting is a quick way to check for consensus. In this method, holding the thumb up means I agree, holding the thumb down means I disagree, and holding the thumb sideways indicate that the member will go along with the decision of the group.

Exercise 7.1 List Down Five Consequences of Not Collecting Proper Requirement

PRODUCT VISION

Before any project starts, be it an Agile project or a traditional one, all stakeholders should be on the same page with regards to the final goal. Its all the more important since in Agile projects, due to the small iteration cycles the team often concentrate on churning out more work in immediate iterations and as a result, lose sight of the overall project objective. The overarching goal of the project which clearly defines its success criterion is called the Product Vision. All the stakeholders – Product Owner, Scrum Master, Team, Management, Customers and other stakeholders should share this same goal. According to Ken Schwaber - “The minimum plan necessary to start a Scrum project consists of a vision and a Product Backlog. The vision describes why the project is being undertaken and what the desired end state is.” (Schwaber 2004, p. 68)

The product owner should lead the vision creation activity along with the team, because at the end the Product owner is the person who is responsible for driving the success of the project and the return on investment. The basis of the product vision is the customer needs and product attributes that will address those needs. It is important that the product vision is stable for a project as otherwise the team will get demotivated and confused leading to project failure if the vision itself is changed too often.

PRODUCT ROADMAP

A product roadmap is a plan where the features are planned for release on a timescale. By looking at the product roadmap the business users and marketing team of the customer will come to know when a particular feature will be made available for use.

This plan talks about the following:

1. Adding value to the customer in each release
2. When releases are needed
3. Features/Details Common about Each Release
4. Date/content/objectives of each release

5. Overall product features (Product Backlog, Business Story)
6. Technical backlog (technical story /non-functional requirements)

The product roadmap is drawn by the product owner. The product roadmap is actually a prioritized work, a list of all product features and non-functional items along with the plan for release. Product backlog and product roadmap are maintained by the product owner. Product backlog items can be added by anyone at any time provided the item has a business value attached to it.

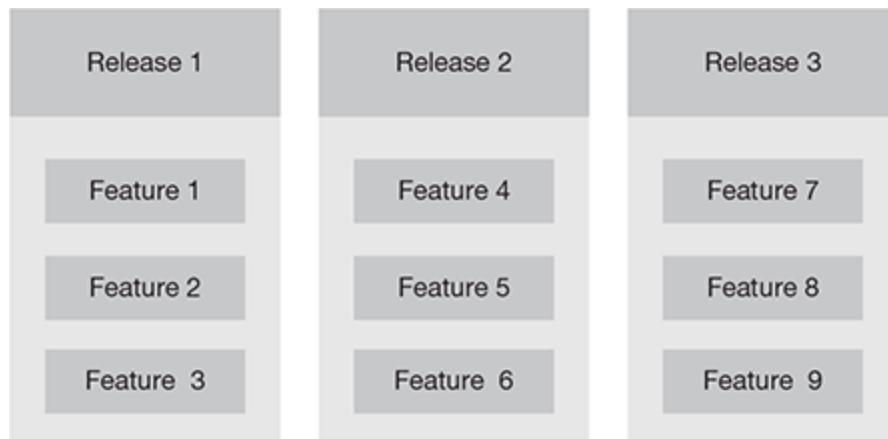


Figure 7.1 Product roadmap

We also need to continuously identify and prioritize

1. Environmental factors
2. Operational factors and
3. Infrastructure factors

in order to improve the quality and value of the deliverables.

BACKLOG

Product Backlog

Project Backlog: My login

<i>Requirement Number</i>	<i>Requirement Description</i>	<i>Requirement Estimation</i>
0 1	User should be able to login to the system	0 3
0 2	User should be able to retrieve their personal information	0 3
0 3	User should be able to send e-mail to other users	0 2
0 4	User should be able to update their personal information	0 4
0 5	User should be able to retrieve the password in case the user forgets the password	0 1
0 6	User can enter their bank details after successful login	0 8
0 7	User can retrieve and update their bank details in a secured fashion	0 6
...		
	Total project estimate in story points	9 5

The product backlog is cumulative of all the stories which are yet to be implemented. The features which were incorporated in the previous iterative cycle (sprint) are no longer part of the

current product backlog or are marked as ‘Done’. For marking a user story as done it is important that it passes all the acceptance tests associated with it. Otherwise, the status of the user story remains ‘Incomplete’. The overall product backlog gets adjusted based on the execution of the recent iterative cycle (sprint) which is reviewed during the checkpoint. The following is a list of the questions to be answered during the client review or team retrospective phase:

1. What was planned (scope) in the cycle?
2. What was done (scope) in the cycle?
3. Is the team working as expected?
4. What went wrong in the concluded cycle?
5. What went well in the concluded cycle?

The answers to the above questions are key inputs to the planning of the next cycle and the functionalities to be added into the cycle. So we have the original work (scope) of the next iterative cycle and thus we have the potential revised work (scope) to consider as a result of the checkpoint. In Agile, the change management process is embedded in the client checkpoint. Changes are accommodated in Agile only in between iterative cycles (sprints). No

changes to scope are incorporated within an ongoing iterative cycle (sprint).

The user stories in the product backlog are at high level and are elaborated as the project progresses. The estimation of the items is done based on the relative complexity model mostly with story points. The total of the story points gives a basic idea of the size of the project at a very early stage though this estimation is at a high level. Remember that the prioritization of the user story should take into account three parameters—the business value associated with the story, the cost of developing the story, and the risk involved. So, the high value low risk items are implemented first while the low value high risk items are prioritized for the end.

Product Backlog Refinement/Grooming

Though Product backlog Grooming or Refinement session is added as a formal Agile ceremony, this is a very important step to keep the backlog up-to-date and correctly prioritized. The reasons why product backlog update may be required are as follows:

1. Change in market condition, which added/deleted some features (stories)
2. Technical constraints or findings
3. Findings from the previous sprints
4. Product architecture change
5. Introduction to new competitor product
6. Discussion with the customer reveals additional requirements
7. Customer altered the internal network that will interfere with the functionality of the end product

The frequency of the backlog refinement session depends on how long the priorities can remain constant or how frequently new business requirements emerge or change existing ones. Some Agile teams fix a 30-min time slot each week when the team and the end users discuss the new or changed requirements, prioritize them to add in the product backlog, and quickly estimate those in story points. Even if weekly refinement is not required, the team should do it at least once 2-3 days prior to the iteration end date. Once the product backlog is updated the estimation is to be updated so that the team gets visibility of the pending works (Refer Figure 7.2).

The release backlog is also refined at the end of every Sprint and at the end of releases. Refinement includes addition, deletion, modification of requirements, and also prioritization of requirements based on the previous works completed. We also do the refinement of estimations based on previous actual data like velocity.

The feedback from the development team and the end users gets incorporated into the product and release backlog, which provides visibility of the remaining work in the project and any risk that has come out as a part of the new technical or functional requirements. This is termed as the Product Feedback Loop.

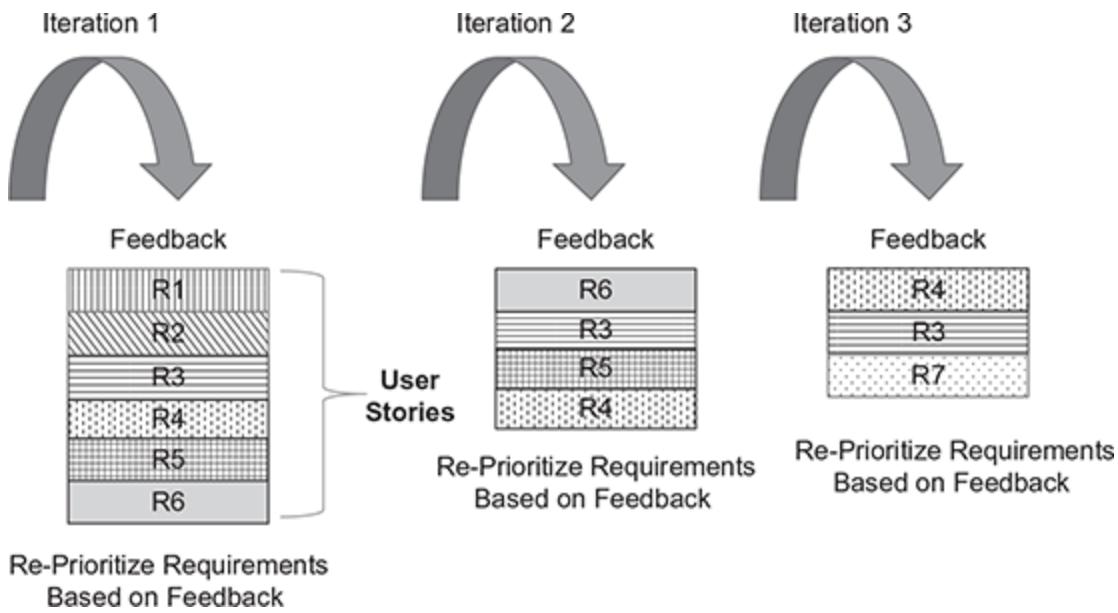


Figure 7.2 Backlog Refinement

Release Backlog

A release backlog is a subset of the product backlog that is planned to be implemented and released in the coming release. A release backlog would presumably contain the same type of items as on a product backlog—preferably user stories. So, during the release planning meeting, a product owner with help from the team and other stakeholders would select the high priority stories and move those items to the release backlog.

The main purpose of the release backlog is to provide the visibility to all stakeholders including the team members of what need to be worked on in next 1–6 months which is the normal release cycle.

Sprint Backlog

The sprint backlog is the list of work the team will work on during the current iteration/sprint. At this stage, the features are broken down into tasks, which, as a best practice, should normally be between 4 and 16 hours of work. As the level of details is very granular, the whole team understands exactly what to do, and potentially, anyone can pick a task from the list. In Agile projects, the tasks on the sprint backlog are never assigned; rather, tasks are signed up for by the team members as needed in accordance with the set priority and the team member skills. This helps to create the self-organizing team and ensures developer buy-in.

In the following example, a list of five requirements is selected from the product backlog to implement in the next sprint.

Sprint N

<i>Requirement Number</i>	<i>Requirement Description</i>	<i>Requirement Estimation (story point)</i>
0 1	User should be able to login to the system	0 3
0 2	User should be able to retrieve their personal information	0 3
0 3	User should be able to send e-mail to other users	0 2
0 4	User should be able to update their personal information	0 4
0 5	User should be able to retrieve the password in case the user forgets the password	0 1
	Total project estimate in story points	1 3

Each of these stories will be broken down into tasks as shown below and a more accurate estimation (in hours) is arrived at for the allocation of the tasks.

Below is a sample of the typical tracking sheet used by the Agile teams to track the work pending at the end of each day. So, the Day 1

figures under the column ‘Work Left’ represents the total estimated effort for the particular task which goes down as the work is done on them.

Requirement	Task	Who's Working	Status	Work Left (Hrs)					
				Day1	Day2	Day3	Day4	Day1	
User login	Database table creation	Vipul	Pending	4					
	UI design	Vipul	Pending		6				
	Authentication module writing	Vipul	Pending			8			
	Test the screen	Arun	Pending	4					
	Pending						
Retrieve info	Pending						
Send mail	Pending						
Update info	Pending						
Retrieve password	Pending						
Remaining work (Hrs)				26					

The above is just an example of how user stories are broken down into tasks and is not a representative of the real world scenario. Based on the complexity there might be more or fewer tasks. Ensure the following points regarding the sprint backlog:

1. The team has taken each requirement in the sprint backlog, broken it down into tasks and estimated the tasks

in hours.

2. Till the team starts working on some of the tasks, all the *tasks are pending* and the work remaining for each of the tasks is the same as the team has estimated.
3. In order to really mark all the requirements as *done* at the end of the sprint, it is important that the team should think of *all the tasks* it can for each requirement during this stage, note it down and estimate it as best as it can and then quickly move to the sprint.
4. This is not used for getting a precise estimate of hours but an *indication of effort/tentative plan* will be enough.
5. The team may find out additional tasks for requirements in the middle of the sprint as they move along developing the requirement. Then the sprint backlog is updated with the new task added. For instance, the team might identify that more integration testing is required for requirement 01 after all requirements are done and adds this task to the backlog. The idea is to make sure to capture everything that needs to be done for the sprint in the sprint backlog.
6. The team members generally volunteer for each of the tasks with the Scrum master facilitating in the case of indecision or confusion.
7. Team members can swap the tasks or help each other during the sprint. But the new allocations should be updated in the sprint backlog.
8. As you can see, the sprint backlog would be updated each day and at the end of each day, you will know exactly where you are. This gives you an insight whether you are headed for completing the requirements or not planned for the current sprint.
9. If there are special requirements like additional documentation or additional regression testing, those should be identified and estimated as different tasks.
10. During the iteration planning meeting if it is found that a story cannot be fitted inside an iteration then the team

should explore the option to split the story into smaller stories.

During the iteration planning meeting if it is found that a story cannot be fitted inside an iteration then the team should explore the option to split the story into smaller stories.

STORY MAPS

User story mapping is a technique to organize and prioritize user stories. The main aim of user story mapping is to understand the end-to-end user requirements and stitch them in a logical thread so that users get maximum return on the business value. The benefits of a user story map are as follows:

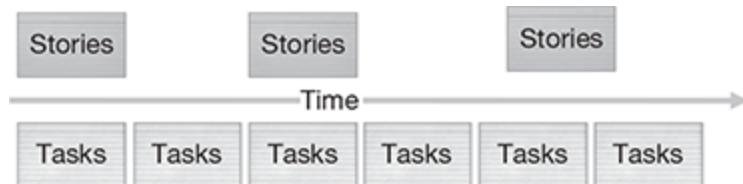
1. Provide visibility to the workflow or value chain
2. Bring out the relationships of larger stories to their child stories
3. Help the team understand the completeness of the backlog
4. Provide a basis for prioritization
5. Plan releases as complete and valuable slices of functionality for maximizing the business value

Below are the steps followed for creating a user story map:

1. Start with the major user activities [stories] to elaborate the end-to-end use of the system
 1. Arrange activities left to right in the order to show ‘What do users do with this system?’



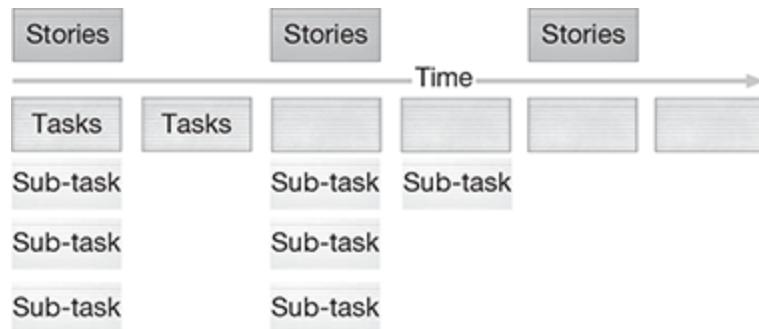
2. Detail the tasks involved against the user stories and arrange them side by side from left to right in the order for the user to perform the tasks.



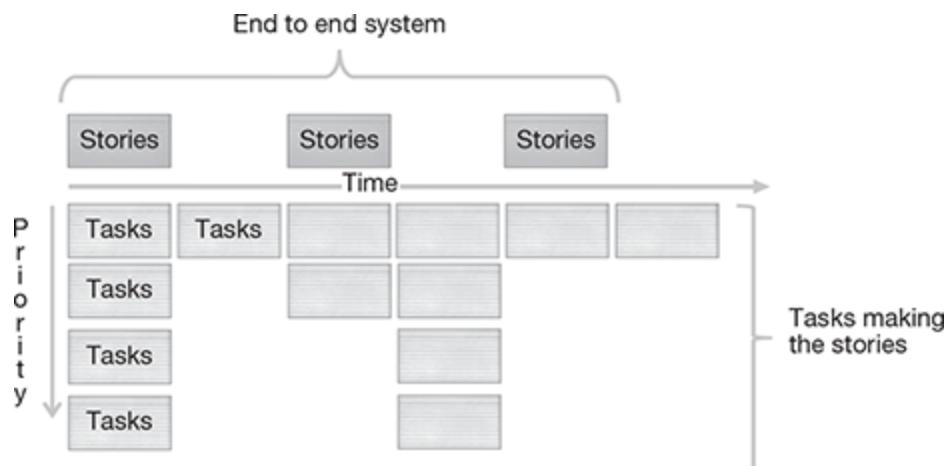
3. The user may be performing multiple parallel tasks to achieve a bigger task. Arrange these smaller tasks one below the other. For example, the user may be entering the customer details, verifying the ID proof and scanning the original documents to complete the customer registration process and all these tasks may go on simultaneously.



While discussing the workflow and the task level details with the users you may get information regarding granular level subtasks that the user is performing for a particular task. This information will be useful while developing the software related to those tasks. So record them in the story card, but do not use in the story map for the sake of simplicity.

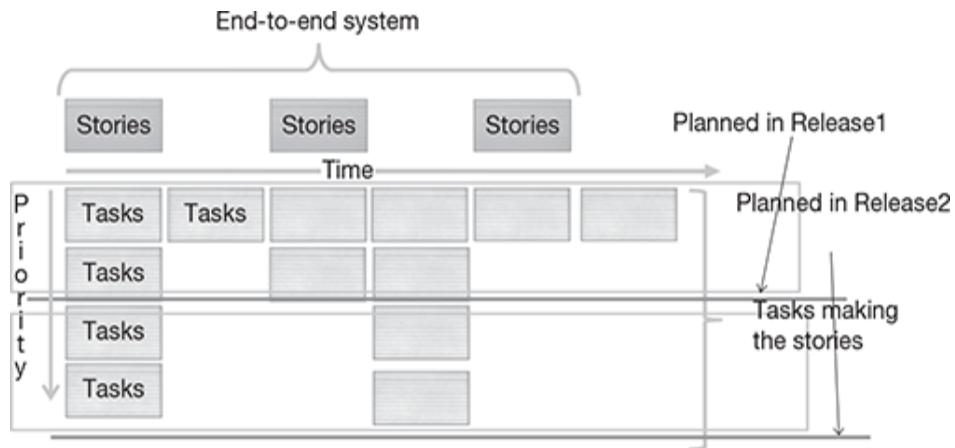


4. So, the user story map created is a typical workflow of the system and also the decomposition of the requirement at the task level.



Arrange the tasks in the order of necessity, keeping the highest priority ones at the top.

5. This map now makes it easy for you to choose the user stories for subsequent releases. The figure below shows how simple it is now to divide the features into releases and more so because each set of user stories chosen for the releases make an end-to-end system which will be useful for the customer.



This leads us to the next logical discussion of how the requirement should be captured in an agile project and validating the same with the customer.

AGILE MODELING

After collecting and validating the requirements, it is crucial that the team has the required clarity to start the design and at the same time, the customer and the end-users are in agreement of what is getting built. In traditional models like the waterfall technique the customer and the user community get to see the end-product only toward the end of the project which causes a lot

of friction at that stage due to expectation mismatch. But according to the Agile principle, showing the product to the end-users as early as possible is the key to success. Agile modeling techniques thus serve a dual purpose—clarify the doubt of the development team to start on the design with confidence and provide the end-users an opportunity to see the sample of the end-result at the early stage. There exist several modeling techniques as discussed below and based on the project scenario one or combination of more than one techniques can be chosen.

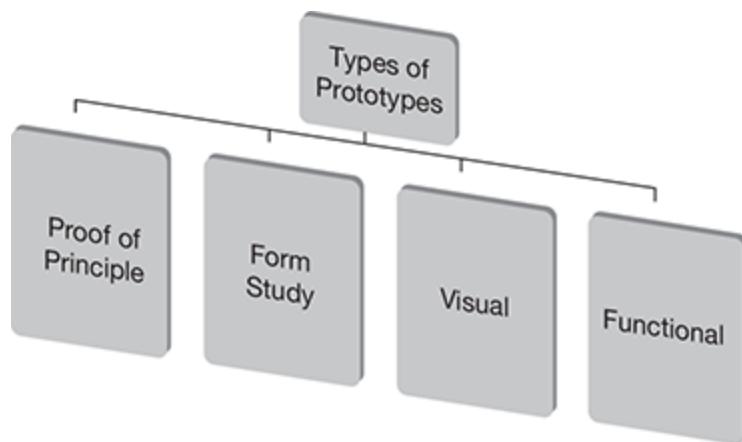


Figure 7.3 Types of prototype

Prototype

A model or prototype is an original type, form or instance of something serving as a typical example, basis or standard for other things of the same category. The word ‘prototype’ derives from the Greek word ‘prototypon’. Prototypes and models are the best ways of presenting ideas to users. They give users a glimpse of what they might get. More requirements are likely to emerge when users see what they will get and through your suggestion. This technique aims to get users to express requirements. The prototype is used to get feedback from the user. Business logic may not be coded in prototyping. Experimental systems are developed using prototyping. It is actually an initial version of the system. It is used in the user interface, design and testing.

Types of Prototype

There is no general agreement on what constitutes a ‘prototype’. Figure 7.3 represents different types of prototypes.

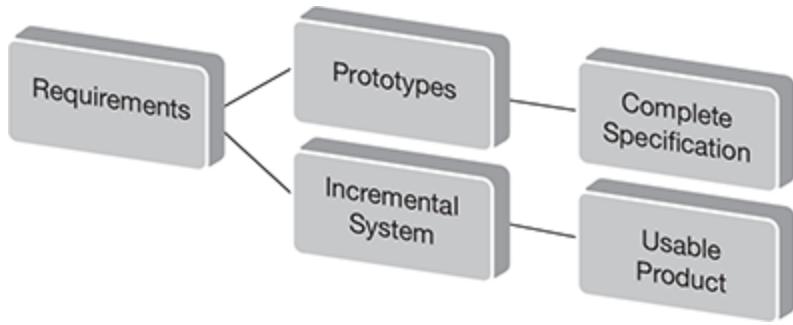
1. **Proof-of-principle prototype** (breadboard): Used to test some aspect of the intended design without attempting to

exactly simulate the visual appearance. Few working codes are part of it just for understanding the purpose.

2. **Form study prototype:** This type of prototype will allow designers to explore the basic size, look and feel of a product without simulating the functionalities.
3. **Visual prototype:** It simulates the appearance, colour, fonts and surface textures of the intended product but no representation of the final function(s) of the final product.
4. **Functional prototype:** It is also called a working prototype. It simulates the final design, aesthetics, materials and functionality of the intended work.

Advantages of Prototyping

1. Early visibility to users
2. Encourages active participation among users
3. Cost effective (development costs reduced)
4. Increases system development speed
5. Helps to reduce the potential risks of delivery
6. Improve system usability
7. Closer match to user needs
8. Design quality improves (try out choices) improve maintainability.



Sometimes the incremental system development is preferred over the prototype building as the former leads to a usable product. But prototypes are preferred when the complete specification needs to be validated within a short span of time. The basic requirement is the input for both prototypes and incremental system. But the prototype is used to develop the complete specification of the system and incremental system develops the usable product.

Wireframes

The visual forms of the prototypes are referred to as wireframes as well. Normally wireframes are used to substantiate the architecture design and the interfacing requirements along with the visual representation so that the team faces fewer

challenges while actually implementing the software.

Personas

Persona creation is a technique of researching and developing a model for the most important customer when you try to see things from the customer's point of view and try to understand their requirement better. Persona is the imaginary representation of the user role which helps in understanding the users of the product better. To this add the personal details to this imaginary character such as:

- Likes and Dislikes
- Profession
- Color of the car

Add as much detail as possible, for example instead of saying he is a professor, mention that he is a professor of Biology at The University of California. Also, you may add a portrait of the persona also if it helps in making the team understand the character.

Persona design helps you to:

1. Identify which group of customers you need to value most.
2. Enhancing your understanding of your customers profoundly and in detail.
3. Discover customer needs in details and unearthing requirements which your customers themselves have not recognized.

BUSINESS CASE DEVELOPMENT

The preparation of business case helps in gaining the confidence of the stakeholder and easy project approval. While developing the business case for Agile development, you might have to take two types of approaches. First, there may be doubt in the stakeholder's minds on how while executing a project in Agile will provide some benefit. And second how the project will be adopting the Agile methodology to maximize the ROI. Both these things need to be handled differently. For the first part, the below findings will help:

Some statistics demonstrating the business case for implementing Agile processes are as follows:

1. Agile projects are 16% more productive than traditionally-managed projects.
2. Agile projects have a 37% faster time to market at a statistically significant level of confidence.
3. Fifteen months after adopting Scrum, an Agile methodology, 86% of [Salesforce.com](#) employees reported much higher job satisfaction.
4. During their first year of being Agile, [Salesforce.com](#) reported that they delivered six times more value.
5. According to University of Calgary research, there is two-thirds less overtime when Agile processes are implemented.
6. After implementing Agile methods, the vast majority of respondents in a VersionOne study reported that defects had gone down by 10% or more.

However, experts warn that this may not be the case for everyone. Agile is a significant change from traditional project management approaches and thus organizations might face some resistance to the implementation of Agile methodologies. According to Rich Mironov, the chief marketing officer at Agile consulting firm Enthiosys:

‘I haven’t seen [anybody] goes through a transformation where everybody came out the other side happy. You’ll lose some folks because it’s not a style fit or they weren’t very good and

you may not fit with agile. Expect some fallout or some people who need to move to the part of the organization that's not going this way'.

The second part is to make sure that the stakeholders are shown the benefit of executing the current project in the Agile methodology rather than in the traditional model. A business case report includes the market research data and the reasons for initiating the project. Prepare the business case based on:

1. **Technical feasibility:** Is the software under development fit for Agile development model?
2. **Team readiness:** Is the client and the project team ready for the deeper involvement necessary for the successful Agile execution?
3. **Business value achieved:** Based on the type of project how often the business values can be realized by the customer?
4. **Quality:** Because of the methodology chosen, Is there any risk in compromising on the quality of the deliverable?

All these questions come to the customer's mind when they see the change in methodology from traditional to Agile. So answering those for the specific project by building up the strong

business case is the key to win the customer confidence.

Product chartering helps in bringing that context to the developer community.

CHARTERING

We want the community that is delivering the product and/or project to understand the why, how and who of the initiative—

Why are we building this software?
How will we judge if it is successful?

Who are the project stakeholders? During a chartering session, we bring the team together to create a common understanding of the product, its vision and its goals. This provides a richer context for all involved and leads to a higher level of engagement for the team members.

Establishing the Why: A chartering session starts with the product owner or business

sponsors explaining in a few statements what the product is and why it is valuable. This is often called an elevator pitch and it establishes the why.

Judging the Success Criteria: Next, the team starts a discussion on the success criteria. This can be driven by the product owner or sponsor with the feedback from the rest of the team. This reveals the success metrics and appropriate expectations from the product. The testers will get an idea from this input about what to explore in the system during the testing.

Roles: Finally, we discuss the roles of the overall product community (sponsors and senior management, subject matter experts and delivery team) and engage as necessary. Other items that may be useful to discuss at this point, if the product and team are ready, could include working agreements, iteration length and review process, standup times, etc. This identifies the community and helps identify any potential gaps in understanding or skill.

Chartering happens prior to starting the release planning. The chartering sessions often help the product owner understand any limitations, business constraints, or special considerations that need to be accounted for or monitored during the project.

The main purpose of the chartering is to bring your product community around a common understanding of the vision, goals and work to be done. Structure the chartering discussion in such a way that maximizes the free discussion during the session.

Summary

Agile analysis and design start with collecting the requirement and coming up with the product backlog. Agile puts emphasis on showing the working software to the customer as early as possible to the customer so that the

requirements are validated at the earliest. This also makes sure that the project team validates the architecture and system design before investing too much time on something which is not working in real production environment. In this chapter, we discussed product roadmap, backlog maintenance in Agile projects, story maps, the process of collecting requirements in Agile projects using the Agile modeling, prototypes, wireframes, personas and business case development and use of chartering.

Chapter 7 Questions and Answers

- Please set yourself a time clock of 1 hour to take this test
- Mark your answers using pencil in the answer sheet provided at the end of this question set
- The correct answers are provided at the end of this question set (after the answer sheet)
- Give one mark for each correct answer for evaluation purpose
- There is no negative marking for the wrong answers
- Practice this test multiple times for better results
- All the very best!

Question 1. Who is responsible for prioritizing the product backlog based on the business values?

1. Product owner
2. Project manager
3. Lead developer
4. Tester

Question 2. Which one of the following statements is correct regarding when a deliverable is accepted on an Agile project?

1. Only senior managers to sign off deliverables
2. At the end of every time-box/iteration the team should get acceptance of project deliverables from the appropriate stakeholders.
3. The team should worry about acceptance of project deliverables from the users only during a UAT phase at the end of the project.
4. All stakeholders are encouraged to sit with the developers during the sprints and provide acceptance then and there.

Question 3. Which one of the following statement is correct regarding a characteristic of the Agile project?

1. The products produced by an Agile project should be cheaper but of lesser quality than those produced by traditional projects.
2. The products will be of top quality in Agile projects but will be more expensive than the traditional models.
3. The product may not satisfy the customer need if produced in Agile way.
4. The customer's expectation on the product quality will be met, as customer representative is involved throughout the development process.

Question 4. If a new requirement emerges when an iteration cycle is running, it should be:

1. Immediately included in the work of the project.
2. Excluded from the project scope and left until a later project.
3. Assessed for business value and, if is important to the business, included in the Product backlog replacing less important requirements.
4. Maintain a list for consideration by the wider group of stakeholders after the project has been completed.

Question 5. In which phase of an Agile project does the customer perform scope verification?

1. At the end of the project
2. Throughout the iteration
3. At the end of the release
4. When the customer is free

Question 6. Which type of artifact presents the quick and easy visual representation of the project health by showing the remaining work in the project?

1. Burn-down chart
2. Burn-up chart
3. Product backlog
4. Velocity graph

Question 7. A persona in an Agile project can be thought of as:

1. The person providing the final acceptance
2. The most powerful person in the project
3. The user role to see requirements from user's point of view
4. The person developing the system

Question 8. Which one of these below is not a feature of the product roadmap?

1. Date/content/objectives of each release
2. Value adding to the customer in each release
3. When releases are needed
4. The estimate of the tasks

Question 9. Which of the following is not a valid reason for updating the product backlog?

1. Findings from the previous sprints
2. Change in market condition which added/deleted some features (stories)
3. Technical constraints or findings
4. Product backlog should never be changed

Question 10. An Agile project team has selected some items (user stories) from the product backlog which will be worked upon in the current iteration. They started to breakdown each user story to task level and estimate the same. What is the team preparing?

1. Sprint backlog
2. Release backlog

3. Burn-down chart
4. Product backlog refinement

Question 11. Which of the following statement BEST describes the use of the story maps?

1. To get end-to-end picture of the requirement
2. To create more task for each requirement
3. To build rapport with the user
4. To get early user acceptance

Question 12. Which style of decision-making is best suited during the Agile requirement gathering phase?

1. Command style
2. Consensus style
3. Consultative style
4. None of the above

Question 13. The estimation of user stories in a product backlog are maintained using

1. Story point
2. Days or hours
3. No estimation is possible in product backlog
4. Random selection of numbers

Question 14. Which technique do you use to rapidly churn out a good number of requirements?

1. Delphi technique
2. Facilitated workshop
3. Teleconferencing
4. Send questionnaire to users

Question 15. Which of the following is not the benefit for prototyping?

1. Early visibility to users
2. Encourages active participation among users
3. Gives breathing time to the developers
4. Increases system development speed

Question 16. A chartering session is conducted for an Agile project team. Which of the following is not the benefit the team will get from the chartering session?

1. Understand why they building this software
2. Get to know the project execution methodologies
3. Understand how they will judge if it is successful
4. Get to know the project stakeholder

Question 17. Which one of the following is not a characteristic of a SMART requirement?

1. Specific
2. Measurable
3. Maintainable
4. Achievable

Question 18. In which order the different backlogs are created in an Agile project?

1. Product backlog, release backlog, iteration (sprint) backlog
2. Iteration (sprint) backlog, product backlog, release backlog
3. Release backlog, product backlog, iteration (sprint) backlog
4. All are created simultaneously

Question 19. Which one of the following statements is true for Agile projects?

1. Each project stakeholders should attend every requirements workshop
2. Retrospectives are only conducted at the end of a project
3. Project manager should facilitate the project's workshops
4. The structure of a facilitated workshop will be managed by a facilitator

Question 20. A team is busy in development work inside an iteration. The project manager found that one day the team is engaged in a meeting to discuss the added items, deleted items, and changed items in the product backlog. What is this session called?

1. Product Backlog Grooming session
2. Estimation session
3. Sprint planning session
4. Retrospective session

Question 21. A project team is progressing with the design, but are not sure whether the architecture will support the features to be developed later. What should the use to confirm their concept?

1. Expert judgement
2. Prototypes
3. Hiring consultants
4. Contact the product owner

Question 22. The main purpose of business case development is

1. Get more money from the customer
2. Start the project without customer approval
3. Gain stakeholder confidence and project approval
4. Build the team better

Question 23. Who creates the product roadmap?

1. Scrum master
2. Product owner
3. Project manager
4. User community

Question 24. When the features are broken into tasks in a sprint backlog, what is the preferable length of task?

1. 4 to 16 hours
2. 1 to 8 hours
3. Exactly 8 hrs
4. Less than 4 hours

Question 25. While executing a sprint a team member found an additional task for a user story he is working on. What is next course of action?

1. Update the sprint backlog with the new task
2. Stop work on the story and move it to next iteration
3. Inform the project manager that there will be a delay
4. Ignore the task as the sprint backlog was frozen

Question 26. The status of the sprint backlog tasks should be updated

1. Weekly
2. At the end of the sprint
3. When the team has some free time
4. Daily

Question 27. Who assigns the sprint backlog items to the team?

1. Scrum master
2. Project manager
3. Product owner
4. Team members nominate themselves for particular tasks

Question 28. Which of the below is a benefit of creating story map?

1. Eases the estimation process
2. Creates more test cases
3. Plan releases as complete and valuable slices of functionality for maximizing the business value
4. Motivates the team to get the work done

Question 29. What is the most important benefit of a client participant focus group?

1. Team gets to know the client better
2. It ensures ownership from the client on the decision taken
3. Client makes sure that everyone is present in the meeting
4. Client will bear the cost of the meeting

Question 30. The main disadvantage of the command style decision making is

1. One or more of the group members may not agree with the decision and thus will have less buy-in in the decision
2. The decision is made faster
3. The decision is almost always wrong
4. Nobody likes the decision-maker

Question 31. Which technique does the facilitated workshop use to churn out requirements?

1. Group writing technique

2. Public speaking technique
3. Delphi technique
4. Brainstorming technique

Question 32. In which of the below situation you will use the consultative style of decision-making?

1. The project manager is very powerful
2. The group is very knowledgeable and reaches consensus easily
3. The group does not have the information, education, skills or experience to make a high quality decision
4. The group does not like to talk to each other

Question 33. Which of the following factors should you keep in mind while prioritizing the user stories?

1. Value, cost, risk
2. Value, quality, cost
3. Time, cost, risk
4. Time, cost, quality

Question 34. Which of the below is not a valid reason for adapting a project in a way that changes the end product?

1. Introduction to new competitor product
2. Discussion with the customer reveals additional requirements
3. The team decides to do the work differently

4. Customer altered the internal network that will interfere with the functionality of the end product

Question 35. During iteration planning meeting you came across a story which does not fit into the iteration. What is the best thing to do?

1. Drop the story from the product backlog
2. Split the story
3. Extend the iteration length
4. Report this to the project manager

Question 36. In Agile project _____ determines how much of the product backlog will be delivered in the current sprint.

1. Team
2. Project manager
3. Scrum master
4. Testers

Question 37. In which type of report you will find market research data and reasons to initiate a project?

1. Agile contracts
2. Prototypes
3. Persona
4. Business case development

Question 38. You have to document the staging requirements to scale a scrum project. Where will you document this?

1. Product backlog
2. Product roadmap
3. Retrospective report
4. Test cases

Question 39. Which of the below practice helps the team to understand what to build

1. Trust
2. Sitting together
3. Real customer involvement
4. Ubiquitous language

Question 40. Thumbing vote is a quick way to check for consensus. Holding thumb sideways indicates:

1. I agree
2. I disagree
3. I will go along with the group
4. I don't want to take part in vote

Question 41. At the end of an iteration, 11 user stories passed all the acceptance test criteria. What will be the status of these stories in the product backlog?

1. Done
2. In progress
3. Incomplete
4. Ready to be tested

Question 42. What below reveals where the project is going and why it is going there?

1. Vision
2. Release planning
3. Risk management
4. Slack

Question 43. Which helps the tester to have some idea about what to explore in the system?

1. Chartering
2. Release plan
3. Iteration plan
4. Black box testing

Question 44. Which of the below is not a type of prototype?

1. Proof-of-principle prototype
2. Form study prototype
3. Visual prototype
4. Test-driven prototype

Question 45. Following are the guidelines used to prioritize features except

1. Features most important to users
2. Features most important to the team
3. Features that provide value to the customer
4. Features are that high value and high risks

Question 46. Specify whether the below statement is true—‘Product knowledge is the knowledge about what will be developed and project knowledge is the knowledge about how it will be developed’

1. True
2. False

Question 47. The product owner has provided a plan to an Agile team which specifies the high level grouping of the features in planned releases. What is this plan called?

1. Product visioning
2. Product roadmap
3. Sprint plan
4. User story

Question 48. Which of the below statement is true regarding the product backlog?

1. All details of the user stories are found in the product backlog
2. The user stories in the product backlog is at high level and are elaborated as the project progresses.

3. Product backlog is created at the beginning of the project and then is thrown away
4. We can skip the stage of product backlog creation

Question 49. An Agile project team is preparing the list of items they will be working in the current sprint. What is this list called?

1. Sprint burn-down chart
2. Release backlog
3. Sprint backlog
4. Sprint report

Question 50. If there are additional requirements for doing extra regression testing in a user story how you will address that in your sprint backlog?

1. Get additional tester in the team
2. Remove the user story from the list
3. Keep the step of additional regression testing as a separate task and estimate for the same
4. Ask the product owner for more clarification

Answer Sheet for Chapter 7 Questions

Question Number	Answer	Question Number	Answer
-----------------	--------	-----------------	--------

Question 1	Question 26		
Question 2	Question 27		
Question 3	Question 28		
Question 4	Question 29		
Question 5	Question 30		
Question 6	Question 31		
Question 7	Question 32		
Question 8	Question 33		
Question 9	Question 34		
Question 10	Question 35		
Question 11	Question 36		
Question 12	Question 37		
Question 13	Question 38		
Question 14	Question 39		
Question 15	Question 40		
Question 16	Question 41		
Question 17	Question 42		
Question 18	Question 43		
Question 19	Question 44		
Question 20	Question 45		
Question 21	Question 46		

Question 22	Question 47
Question 23	Question 48
Question 24	Question 49
Question 25	Question 50

Answers for Chapter 7 Questions

Question Number	Answer	Question Number	Answer
-----------------	--------	-----------------	--------

Question 1	A	Question 26	D
Question 2	B	Question 27	D
Question 3	D	Question 28	C
Question 4	C	Question 29	C
Question 5	B	Question 30	A
Question 6	A	Question 31	D
Question 7	C	Question 32	C
Question 8	D	Question 33	A
Question 9	D	Question 34	C
Question 10	A	Question 35	B
Question 11	A	Question 36	A
Question 12	B	Question 37	D
Question 13	A	Question 38	A
Question 14	B	Question 39	C
Question 15	C	Question 40	C
Question 16	B	Question 41	A
Question 17	C	Question 42	A
Question 18	A	Question 43	A
Question 19	D	Question 44	D
Question 20	A	Question 45	B
Question 21	B	Question 46	A

Question 22	C	Question 47	B
Question 23	B	Question 48	B
Question 24	A	Question 49	C
Question 25	A	Question 50	C

Explanations for Chapter 7 Answers

1. **Answer A**

Product owner is the customer representative with most business knowledge and he/she should prioritize the product backlog based on business value.

2. **Answer B**

Agile emphasizes on getting the acceptance of the requirements that are delivered within an iteration at the end of the iteration.

3. **Answer D**

Because of continuous involvement of the customer during the requirement as well as the development process, there is very less chance in Agile projects that the deliverable is too far away from the customer's expectation.

4. **Answer C**

Product backlog can be updated based on the business value of the new requirements.

5. **Answer B**

The requirement verification is done by the customer either throughout the iteration or at the end of the iteration. The developed features are marked as done once the customer does the verification.

6. **Answer A**

A burn-down chart is a visual representation of the work remaining in an iteration and is easy to read.

7. Answer C

Persona technique is used to view requirements from customer's point of view.

8. Answer D

At the product roadmap level the estimate of the tasks are not available. All other are the features on product roadmap.

9. Answer D

Product backlog can be changed based on the conditions like point A, B and C.

10. Answer A

The team has selected the items to work in the current sprint and is elaborating the sprint backlog.

11. Answer A

A story map is created so that the project team as well as the user community is clear about the end to end use of the system.

12. Answer B

The consensus style ensures maximum participation from the users and thus is preferred for Agile requirement gathering.

13. Answer A

Story point is the relative estimation technique which is mostly used for product backlog estimation.

14. Answer B

Facilitated workshops are the best way to churn out large number of requirements rapidly as it helps in collective brainstorming.

15. Answer C

16. Answer B

17. Answer C

SMART stands for:

1. Specific—clearly states what is required
2. Measurable—to confirm when it has been met

3. Achievable—can be done, e.g., technically possible
4. Realistic—is reasonable, e.g., cost is not prohibitive
5. Timely

18. *Answer A*

Product backlog is created first from which release backlog selects the items for the next release and then from that sprint backlog selects the items for the next sprint.

19. *Answer D*

A facilitator manages the structure of the facilitated workshop.

20. *Answer A*

Throughout the execution of the Agile project, the product backlog gets refined. The session where new items and changes to the product backlog is discussed is called the Product backlog Grooming session.

21. *Answer B*

Prototyping is a quick way of validating the architecture of the team is not sure of the technical capabilities.

22. *Answer C*

Business case represents the viability of the project and thus helps in getting stakeholder support and confidence.

23. *Answer B*

Product owner represents the business community and thus creates the product roadmap.

24. *Answer A*

The preferred length of the tasks is 4-16 hours to make sure that those are not broken too small or too big so that those can be planned and tracked.

25. *Answer A*

The first thing will be to add the task in the Sprint backlog. After that if required the re-planning should happen based on the time available and time required.

26. *Answer D*

Sprint backlog progress status needs to be updated daily so that the burn-down chart can be prepared.

27. *Answer D*

In Agile the team is empowered and they plan and nominate themselves for the tasks.

28. *Answer C*

Story map helps in creating the release plan, which is best aligned with the business need and thus maximizing the business value.

29. *Answer C*

The idea of a focus group is to bring all the stakeholders together and brainstorm. So, the client helps in ensuring that all the parties are present in the group.

30. *Answer A*

Command style decision making doesn't take into account everyone's point of view and thus may leave few stakeholders unconvinced because they don't have a buy-in to the decision.

31. *Answer D*

The main purpose of the facilitated workshop is to brainstorm and churn requirements.

32. *Answer C*

The group members are consulted before making the decision but it is not necessary that everyone agrees with the decision because the group may not have the information, education, skills, or experience to make a high-quality decision.

33. *Answer A*

Product backlog prioritization should take into account three primary items - Value of the item, cost of developing the item, and risk associated with developing or not developing the item.

34. *Answer C*

Change in client need or market dynamics can change the product behavior but the team should not change it as their will.

35. *Answer B*

Very large user stories are difficult to plan and thus should be split.

36. Answer A

Planning is done by the team in Agile projects.

37. Answer D

Business case provides the information about the importance of the project and the return on investment.

38. Answer A

Product backlog contains all the requirements for the project, be it functional or non-functional.

39. Answer C

Agile recommends direct customer involvement throughout the project life cycle so that all are clear about the requirement.

40. Answer C

Holding one's thumb sideways means 'I will go with the decision of the group'.

41. Answer A

'Done'ness is defined as the passing of acceptance tests.

42. Answer A

Product vision provides the visibility of why the product is being built and who is going to use it. The details help all the stakeholders understand the necessity of the project.

43. Answer A

Chartering provides the idea of who is going to use the product and why which helps the testers preparing the test plan and cases accordingly.

44. Answer D

Proof-of-principle prototype, Form study prototype, Functional and Visual prototype are valid types of prototypes.

45. Answer B

Feature lists are prioritized according to business need and not according to the team's convenience.

46. Answer A

47. Answer B

Product roadmap is the prioritized and grouped set of requirements for the project.

48. Answer B

Requirements in Agile projects are evolved with the project and more details are added to the requirements as the project progresses.

49. Answer C

A subset of the product backlog which will be worked upon in the current Sprint is called the Sprint backlog.

50. Answer C

Every task that needs to be performed in a sprint should be part of the Sprint backlog and be estimated.

Key Terms

Product roadmap This is a plan where the product backlog items are categorized release wise so that the business users come to know what features are getting available at what point of time.

Product backlog Product backlog is the cumulative of all the stories which are yet to be implemented.

Release backlog A release backlog is a subset of the product backlog that is planned to be implemented and released in the coming release.

Sprint backlog The sprint backlog is the list of work the team will work on during the current sprint. At this stage, the features are broken down into tasks, which, as a best practice, should normally be between 4 and 16 hours of work.

Story maps User story mapping is a technique to organize and prioritize user stories. The main aim of user story mapping is to understand the end to end user requirements and stitch them in a logical thread so that users get maximum return on the business value.

Personas Persona creation is a technique of researching and developing a model for the most important customer when you try to see things from customer's point of view and try to understand their requirement better.

Product Quality and Process Improvement

KNOWLEDGE AND SKILLS COVERED

Tools and Techniques: High

- *Frequent Verification and Validation*
- *Test-driven Development*
- *Testing, including exploratory, and usability testing*
- *Definition of Done*
- *Value Stream Mapping Fishbone Diagram*

Level 1: High

- *Feedback Techniques for Product*
- *Incremental Delivery*
- *Project and Quality Standards for Agile Projects*

Level 2: Medium

- Self-assessment
- *Continuous Improvement Process*
- Process Analysis Techniques
- *Kaizen*
- Five WHY Analysis

Level 3: Low

- Control Limits

INTRODUCTION

Quality is defined as the degree to which a set of inherent characteristics fulfill requirements. The total cost of all efforts related to quality throughout the product life cycle is called cost ...

Interpersonal Skills

KNOWLEDGE AND SKILLS LEVEL

Tools and Techniques: High

- *Emotional Intelligence (EI)*
- *Collaboration*
- *Adaptive Leadership*
- *Negotiation*
- *Conflict Resolution*
- *Servant Leadership*

Level 1: High

- *Brainstorming Techniques*
- *Building Empowered Teams*
- *Coaching and Mentoring*
- *SHU-HA-RI, Dreyfus, Tuckman*
- *Knowledge Sharing*

- *Leadership Tools and Techniques*
- *Problem Solving Strategies, Tools and Techniques*
- *Team Motivation*

Level 2: Medium

- *Building High Performance Teams*
- *Facilitation Methods*

Level 3: Low

- *Systems Thinking*
- *Culture and Team Diversity*

Soft skills are an essential part of Agile projects, since Agile is more people-oriented. Soft skills are necessary to handle people and they include skills like conflict management, negotiation, leadership skills and interpersonal skills. The role of the project manager, the team, and the sponsors in an Agile project is very much different from a traditional project and thus it is important that all the stakeholders understand it for the success of an Agile engagement. A changed approach in leadership along with the proper coaching and mentoring techniques can build a high performance team in Agile.

EMOTIONAL INTELLIGENCE QUOTIENT (EQ)

Emotional intelligence is your ability to acquire and apply knowledge from your emotions and the emotions of others in order to be more successful and lead a more fulfilling life.

It indicates the ability to bring awareness to the emotions as they arise, notice them for what they are and decide how to best use them.

Coaching abilities can be improved by improving EQ. We need to control our emotions while working with the team.

Emotional intelligence is the ability of an individual to deal successfully with other people, to manage one's self, motivate other people, understand one's own feelings and appropriately respond to the everyday environment.

In working with emotional intelligence, author Daniel Goleman defines EI in the workplace as the ability of employees to recognize:

1. Their own feelings
2. The feelings of others
3. What motivates them?

4. How to manage their emotions, both in themselves and in relationships with others

	Definition	Hallmark
Self awareness	The ability to recognize and understand your moods, emotions and drives as well as their effects on others	Self-confidence Realistic self development Self-deprecating sense of humor
Self-regulation	The ability to control or redirect disruptive impulses and moods. The propensity to suspend judgment to think before acting	Trustworthiness and integrity Comfort with ambiguity Openness to change
Motivation	A passion for work for reasons that go beyond money or status The propensity to suspend judgment—to think before acting	Strong drive to achieve Optimism, even in the face of failure Organizational commitment
Empathy	The ability to understand the emotional makeup of other people Skill in treating people according to their emotional reactions	Expertise in building and retaining talent Cross-cultural sensitivity Service to clients and customers
Social skills	Proficiency in managing relationships and building networks An ability to find common ground and build rapport	Effectiveness in leading change Persuasiveness Expertise in building and leading teams

Figure 9.1 Components of emotional intelligence

According to Fisher's research, the most common negative emotions experienced in the workplace are as follows and this is applicable for Agile projects too:

1. Frustration/irritation
2. Worry/nervousness
3. Anger/aggravation
4. Dislike

5. Disappointment/unhappiness

There are five parts to EI. First is to know what you are feeling. The second is managing your feelings, especially distressing feelings. The third is self-motivation, the fourth is empathy and the fifth is managing relationships. Figure 9.1 summarizes the components of EI.

COLLABORATION

Animated interaction, two way conversation, real understanding and progress are all hallmarks of cooperation. The sum of the individual parts is called cooperation. Collaboration means the whole is greater than the sum of the individual parts. The collaboration includes innovation and astonishing results.

Team Collaboration and Team Commitment

This includes providing feedback, resolving issues and coordinating changes to enhance project performance and involves a combination of skills with special emphasis on communication, conflict management,

negotiation, leadership and recognition for high performance for better management (see Figure 9.2).

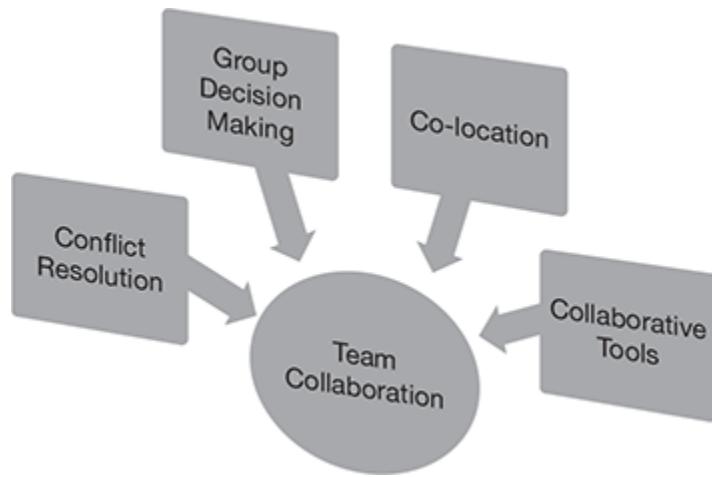


Figure 9.2 Team collaboration

Task 1: Facilitate close communication within the team and with appropriate external stakeholders through co-location or the use of collaboration tools in order to reduce miscommunication and rework.

Task 2: Reduce distractions in order to establish a predictable outcome and optimize the value delivered.

Task 3: Participate in aligning the project and team goals by sharing project vision in order to ensure that the team understands how their objectives fit into the overall goals of the project.

Task 4: Encourage the team to measure its velocity by tracking and measuring actual performance in previous iterations or releases in order for members to gain a better understanding of their capacity and create more accurate forecasts.

Co-location and Distributed Team

Is there really a difference in working in a co-located team vs working in a distributed team? Can we assume that co-located teams are always more efficient than the distributed team? We may tend to think this way, because working in co-located team is easy because of the ease of communication as all the team members are sitting together. But it is not really hard to have an equally effective team with distributed team, if the care is taken to establish communication channels effectively.

The pure Agile framework asks for the co-located team. It even goes to an extent of asking all the team members to sit in the same room. In fact, Agile manifesto states that ‘The most efficient and effective method of conveying information to and within a development team is face-to-face conversation’. Daily stand-up meetings and daily status tracking planning for true Agile teams are designed accordingly. In the case of distributed Agile team, communication is the biggest challenge which needs to be planned effectively for success.

ADAPTIVE LEADERSHIP

Organizations do not operate as machines; they operate as an adaptive system constantly changing with triggers like the market, workforce, economic situation, technology and several other factors. So, it matters a great deal how the leaders conceive their organizations as machines or living adaptive systems. This perception shapes the roles they and their people play. This in turn, has direct relationship on their effectiveness to tap human potential. Because of the complex nature of the projects that we carry

out, mechanically-based leadership and organizational practices are not adequate to the adaptive challenges being faced.

The old paradigm focuses only the mechanical aspects of how organizations operate; those activities that must be repeated in a standardized way to produce a standard output. In the mechanical sphere of operations, change and creativity are viewed as a threat to efficiency.

In an organization which is led like a machine, people are treated as parts of machines—mindless extensions of impersonal processes. When that happens, commitment level is greatly reduced and the creativity and latent potential of the team remains greatly untapped.

The adaptive view of organizations and leadership presents a philosophy which is in sharp contrasts along a number of dimensions.

From the above discussion you can understand how the working atmospheres differ. In the mechanically managed and structured

organization, people in one department know very little of the goals and contributions of the other departments because they do not realize how important it is for them to know that. Since work is highly specialized and interdepartmental communication is far from reality, things that need coordination slip through the cracks. Team motivation is likely to be poor.

Mechanical Approach	Adaptive Approach
Attention is focused on activities performed.	Attention is focused on the value-added outcomes.
Job descriptions are long, detailed and constraining defining clear boundaries.	Job descriptions are intentionally broad based to allow for flexibility and collaboration.
Role expectations are kept narrow and rigid.	Roles are fluid. Within limits, people are expected to play roles for one another.
Contacts are confined and communication is controlled by higher management.	Contacts are open and inter-group network formation is encouraged.
Policies are mostly oriented towards drive and control.	Policies encourage people to take a self-organizing approach to find solutions.
The organizational structure is bureaucratic and structured around many departments.	The structures are more fluid and of shorter duration to enhance flexibility and responsiveness.
Rank-based authority is practiced, and it is expected that influence will equate formal authority.	Value addition is given a preference over authority. A person who can nourish team's capability is preferred in the rank.
Efficiency and predictability are prime focus and reinforced through processes.	Achievement, innovation and change are primarily

	focused and rewarded keeping in mind the final goal.
Inter department communication and cooperation is reduced due to dependence on formalization and clearances.	Cooperation is a highly regarded value in the organization and is made easy through simple organization structure.
Information is kept closely guarded and is distributed on higher management's prerogative.	Information is easily available to facilitate work accomplishment and to provide opportunities for more people to add value to operations.
Traditional values such as unit loyalty and obedience are fostered which stifle initiative and hamper teamwork across departments.	Newer values such as collaboration, teamwork and responsiveness along with treating other units as internal 'customers' are encouraged.

Due to inundated reasons, the nature of the business is far from stable—mergers, acquisitions, new technology to absorb, changing demographics, de-regulation, global competition, competition from small, fluid, adaptive organizations all add to the chaos that any organization needs to handle. This calls for a more flexible and adaptive approach to the leadership.

Adaptive leaders demonstrate a set of values which are as follows:

1. Proactive attitude: Foresee opportunities and put the resources in place to capitalize them.
2. Employing a broad-based style of leadership which makes them personally more flexible and adaptive.
3. Consider and discuss diverse and divergent views when possible before making major decisions.
4. Ready to admit when they are wrong and alter or abandon a non-productive course of action.
5. Strategizes their organization's capacities to learn, transform structure, change the culture and adapt technology.
6. Are willing to experiment, take risks which may produce high value for the team.
7. Open to new ideas and stay abreast by being lifelong learners.
8. Love and encourage innovation across all the levels in the organization.

Interpersonal skills include problem management, conflict management, negotiation, motivational skills, and leadership skills.

Interpersonal Skills

Interpersonal skills (Figure 9.3), otherwise called soft skills, are very important for team development. Skills such as showing empathy to

stakeholders, knowing and using the technique of influencing others, always looking for creative ideas and preferring group facilitation are valuable skills required for a project manager when managing the team.

The project management team can greatly reduce problems and increase cooperation by understanding the sentiments of other members, anticipating their actions, acknowledging their concerns and following up on their issues.

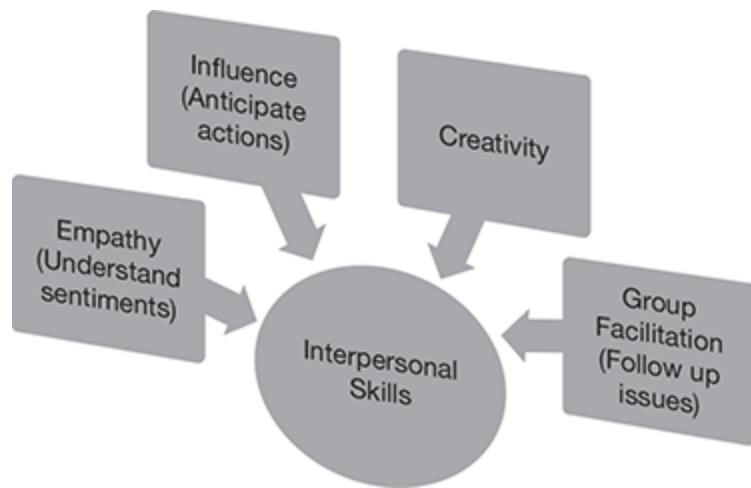


Figure 9.3 Interpersonal skills

NEGOTIATIONS

Negotiation mostly happens with functional managers or other project management teams to ensure receiving appropriate resources within required timeframes.

Win-Win Negotiations

1. Establish rapport and common goals
2. Probe for understanding of beliefs, goals, win-win options
3. Paraphrase for confirmation/affirmation
4. Analyze outcomes and risks
5. Summarize what was agreed on, and next steps (even if these are only “baby steps”)

The customer can get engaged in negotiation with all the internal stakeholders and come up with the prioritization ranking list. But most of the customers tend to prioritize 80% of the requirement as high priority requirements. This is a kind of biasing. We may need to employ few

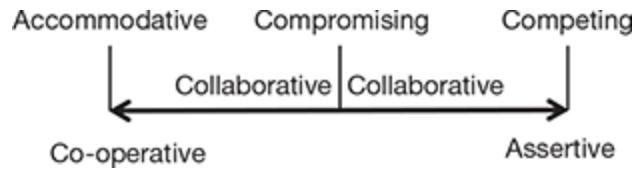
other technique to avoid this kind of biasing issues.

Typical Negotiating Challenges of Agile Projects

1. “Build or buy” decisions
2. feature prioritization
3. catch-up and Overtime needs
4. contract overruns
5. vendor Selection
6. Bonuses and pay scales for Agile Team
7. Risk Management strategies

Negotiation Mode

Following is the negotiation modes of which are used for conflict resolution also.



1. **Accommodative:** 100% co-operative, 0% assertive
2. **Competing:** 0% co-operative, 100% assertive
3. **Compromise:** Equal distance from co-operative and assertive
4. **Collaborative:** Combination of co-operative and assertive

Steps for Negotiation

Juliet Erickson, an international communication coach, gives the following nine steps to be used for negotiation (Figure 9.4).



Figure 9.4 Negotiation steps

Step 1—Determine Your Negotiation Objective

Determined what do you want to achieve through this negotiation? This is a key step and lots of people get into a negotiation without knowing the objective, thereby it fails and goes in favor of the party at the other side.

Step 2—Identify the Issues for You and the Other Person/Party

Understand as early as possible what the other party cares about and what they are or are not willing to trade.

Step 3—Think About Strengths and Weaknesses for Issues on Both Sides

Be practical and acknowledge your strengths and weaknesses. Would you trust someone taking a strong position on an issue that is known by everyone to be a weakness?

Step 4—Agree the ‘Attitude’ You will Take into the Negotiation

We all bring an ‘attitude’ to negotiating. Your attitude can come from your experience, your personal style or comfort zone. The other person has an attitude as well and you need to take this into consideration to avoid problems later. Ask yourself: ‘What do I want my relationship with this person to be like after the negotiation?’ The world is a small place and no doubt you will be negotiating with them again. Everything you do needs to be consistent with this attitude. If you

say you want to co-operate, do not behave like a bully.

Step 5—Decide on a Strategy for Each Issue

Not all issues are created equal. This means you will need a different strategy for each issue. For example, on issues where you are strong or that are critically important to you, be a bit tougher. On issues that you do not care about very much or do not impact you, be more flexible.

Step 6—Plan Your Bargaining Position for Each Issue

Here is a fun tool called a ‘bargaining position model’. Get a flip chart or some blank sheets of paper. Take a sheet for each issue and draw a horizontal line across it. At the left side write your ideal position. At the right-hand side of the line write the worst possible outcome you could accept. Create two more positions in between. Repeat this for each issue and then think about what the other side might do in reverse. Now comes the time to test whether this step is in

keeping with your overall attitude. I have seen many people say they can be flexible on a less important issue, then put a really tough opening position on the left-hand side of the bargaining model.

Step 7—What Tactics will be Used?

Tactics are the behaviors that bring your attitude to life. There is no point in saying you want a cooperative negotiation if your tactics are mean or rude. Hiding information, refusing to say what you think, stalling unnecessarily, raising new points at the last minute are all competitive tactics that annoy and frustrate people, and rarely add any value. Think of the tactics that may be used on you and how you will neutralize them.

Step 8—Plan Your Questions and Evidence

While planning, you will discover gaps in your knowledge of the other person's needs and priorities. I encourage you to talk with them directly to fill these gaps. Think about the

questions they will ask you and what you need to ask them.

Step 9—Decide on Your Plan B

It is always good to do some worst-case scenario planning. No matter how well you planned, things can still go wrong. You will have a lot more confidence if you go into the final stages of negotiation with a plan B. Tip: Care about the result, but not too much.

These simple but very effective planning steps will keep you focused on what is important and the actions that need to be in place to ensure you achieve your negotiation objective sooner with less stress.

CONFLICT RESOLUTION

While working with a team of diversified people it is natural to have conflicts within the team. Though we think of conflicts as the manifestation of the discontented workforce, but in reality, constructive conflicts can make the team bonding stronger and can create a lot of

innovative ideas inside the team. Before you approach to resolve any conflict among the team members you need to know the common reasons of the conflicts and how you can channelize the conflicts into a more constructive discussion among the team for attaining the bigger goal of the organization.

Conflict is inevitable. It should be addressed early and usually in private, using direct and collaborative approach. It is natural and forces for alternatives. Conflict should be considered as a team issue. Conflict resolution should focus on the present and not in the past and it should be issue based and not person based.

Conflict Resolution Techniques

1. **Problem-solving:** Confronting.
2. **Compromise:** Searching for a solution that brings some degree of satisfaction for all parties.
3. **Forcing:** Pushing one viewpoint at the expense of others.
Offers win-lose solution.
4. **Smoothing:** Emphasizing the areas of agreement rather than disagreement.
5. **Collaborating:** Incorporating multiple viewpoints and insights from differing perspectives leads to consensus and commitment

6. **Withdrawing:** Avoiding

Cause of Conflict

The reasons for conflict among team members may be due to the following:

1. **Facts (information):** This is often the biggest source of conflict when each person thinks that the information they have is the most authentic and manipulates their activity accordingly. A very clear communication on the organization policies and a frequent update on the organization's progress can often bring down this type of fact mismatch.
2. **Goals (roles):** This arises when the team members are not working towards the same goal resulting in the unclear roles definition. For example, between a developer and tester there may be a conflict regarding who should retest the bug identified during testing if the team's goal of preventing the defect leakage as much as possible is not clear to them.
3. **Methods (needs):** Different people approach their work differently, meaning the methods of achieving the same result may vary widely based on several factors. For example, one team member may like to work on a job until it is finished, without interruption, while another member will take scheduled breaks, no matter at what point they are in a task. So, disagreement surfaces between all people working on the job, about when to take breaks.
4. **Values (beliefs):** Values are deep rooted inside people resulting from the way individual upbringing was done.

But undermining other people's values often becomes the major reason for conflict.

Additional causes of conflict can be as follows:

1. Competition
2. Boundaries
3. Hidden agendas

Levels of Conflicts

Figure 9.5 illustrates the levels of conflicts in the negotiating process.

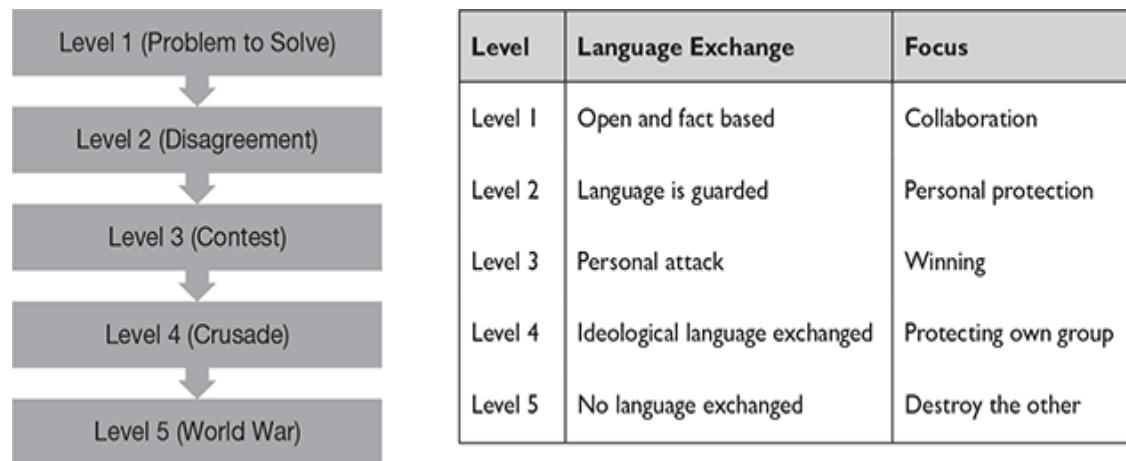


Figure 9.5 Levels of conflicts

Approach to Conflict

Your approach to any conflict is very important as the team members need to buy into your ideas when you are trying to pitch in inside a conflict.

1. Listen to understand the other person's position (or perspective, etc.).
2. Once understood, describe the other person's position (or perspective, etc.) as best you can see it.
3. Describe the problem as you understand it.
4. Describe what other member(s) do that you believe contributes to the problem.
5. Describe what you do, which could contribute to the problem.
6. Clearly state what you want or need from others to resolve the problem.
7. State the first steps you want to take to resolve the conflict.

This means before you take a dive to resolve any conflict make sure that all the members involved have clearly articulated statement of understanding of the problem. Make sure that the team has continual focus on the problem or issue rather than on the personalities or individuals. It is important that the team members understands that he/she is a part of the problem and must be part of the resolution. This will encourage

openness and willingness to pursue a win-win solution.

Qualities of a Constructive Conflict in Agile Project

As was mentioned earlier, conflict can actually help in building team effectiveness. Useful qualities of conflict include the following:

1. Members evaluate one another's ideas and discuss the shared view.
2. Members generate more ideas through constructive discussion.
3. Participation among members is high and widely shared.
4. Members show flexibility in their ideas and actions; they do not insist on one view or behavior but are open to consider all options.
5. Members strive for successful outcomes.
6. Members consciously work to find conclusions that others can accept.
7. Members successfully influence others and are also influenced by them in turn.

Consent Check, Consensus Check

They help to avoid misunderstandings and increase positive energy within the team.

Consent check hears all the voices that want to

be heard are heard. Consensus check ensures consensus is reached and the team moves forward.

QUICK QUIZ ON CONFLICT MANAGEMENT

Figure out what conflict resolution technique is being used in the below scenario and try to match it

I have decided to use the solution by the project leader and not going to listen from anybody.	Withdrawal
I don't have the time and you do whatever you want.	Problem-solving
Let's all sit together after 6pm and discuss how to solve this.	Forcing
You guys agree on all the points and only 2 points are differing. I am sure it's not a big issue.	Compromising
I request you to give-up the third point and take the point of Kumar as its valid it will be great.	Smoothing

Answer

I have decided to use the solution by the project leader and not going to listen from anybody.	Forcing
I don't have the time and you do whatever you want.	Withdrawal
Let's all sit together after 6pm and discuss to solve this.	Problem-solving
You guys agree on all the points and only 2 points are differing. I am sure it's not a big issue.	Smoothing
I request you to give-up the third point and take the point of Kumar as its valid it will be great.	Compromising

SERVANT LEADERSHIP

There are different leadership roles taken by Agile coach. They are as follows:

1. Bulldozer for impediments
2. Shepherd for Agile principles
3. Servant leader
4. Guardian for quality improvement

The approach of leadership that has been successful in Agile engagements is the servant

leadership.

Servant leadership is a philosophy and practice of leadership. The concept of servant leadership is ancient. Chanakya wrote, in the fourth century B.C., in his book *Arthashastra* ‘the king [leader] shall consider as good, not what pleases himself but what pleases his subjects [followers]’ and ‘the king [leader] is a paid servant and enjoys the resources of the state together with the people’.

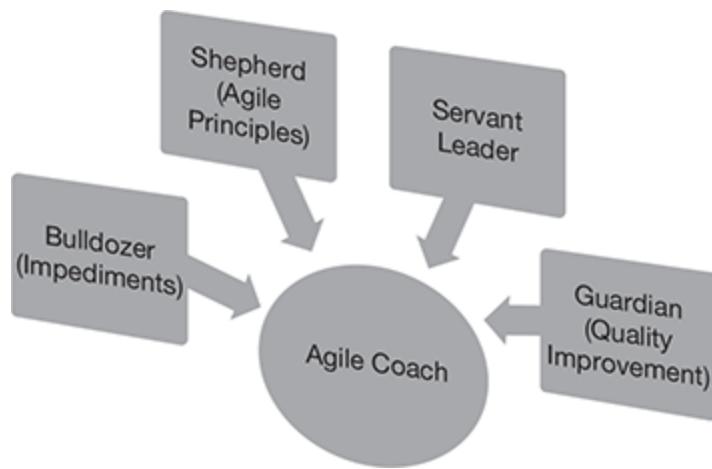


Figure 9.6 Agile coach role

The terms of ‘modern servant leadership’ and ‘servant leader’ were coined by Robert K. Greenleaf in 1970. He coined this phrase in his

essay ‘The Servant as Leader’ and supported by many leadership and management writers.

Servant leaders achieve results for their organizations by giving priority attention to the needs of their colleagues and those they serve. Servant leaders are often seen as humble stewards of their organization’s resources: human, financial and physical.

This is a management philosophy which implies a comprehensive view of the quality of people, work and community spirit. A servant leader concentrates on the needs of the team members and asks himself how he can help them to solve problems and promote personal development. His guiding principle is that only content and motivated people are able to reach their targets and to fulfill the set expectations, and thus he places his main focus on people.

In his essay *The Servant as Leader*, Greenleaf said:

‘It begins with the natural feeling that one wants to serve, to serve first. Then conscious choice brings one to aspire to lead. That person is sharply different from one who is a leader first, perhaps because of the need to assuage an unusual power drive or to acquire material possessions... The leader-first and the servant-first are two extreme types. Between them, there are shadings and blends that are part of the infinite variety of human nature’.

The most common difference in leadership styles is the distinction between autocratic and participative leadership styles. The authoritarian style of management gives stress on clearly defining tasks and monitoring their execution and results. The decision-making responsibility lies with the executive team. But in the practice of a participative leadership style, employees get involved in decision-making. More tasks are delegated. As a result of involving the employees in the decision-making, the employees’ influence and responsibility increases.

Servant leadership can be most likely mapped with the participative management style. The

highest priority of a servant leader is to encourage, support and enable the team members to unfold their full potential and abilities. This leads to an obligation to delegate responsibility and engage in participative decision-making resulting in the greatest possible performance and employee satisfaction.

Characteristics of Being a Servant Leader

Larry C. Spears has extracted a set of 10 characteristics that are central to the development of a servant leader:

1. **Listening:** A servant leader has the motivation to listen actively to team members and support them in decision identification. The servant leader particularly needs to pay attention to what remains unspoken in the management setting relying on his inner voice in order to find out what the body, mind and spirit are communicating.
2. **Empathy:** A servant leader attempts to understand and empathize with the team. Team members are not considered as only employees, but also as people who need respect and appreciation for their personal development. As a result, leadership is seen as a special type of human work, which ultimately generates a competitive advantage.
3. **Healing:** A servant leader tries to help people solve their problems and conflicts in relationships, because he wants to encourage and support the personal development of

each individual team members. This leads to the formation of a business culture, in which the working environment is dynamic, fun and the team is free of the fear of failure. A great strength of a servant leader is the ability for healing one's self and others by helping the team balance work and personal life.

4. **Awareness:** A servant leader needs to gain general awareness, especially self-awareness so that they can view situations from a more integrated, holistic position. This also provides them a better understanding of ethics and values which are the building blocks for their leadership quality.
5. **Persuasion:** A servant leader does not use their power and status for coercing compliance; they rather try to convince those they manage through effective discussion. This element distinguishes servant leadership most clearly from traditional, authoritarian models.
6. **Conceptualization:** A servant leader expands her horizon beyond day-to-day realities. That means she has the ability to see beyond the limits of the current operating business and also focuses on long-term operating goals rather than short-term benefits. As a result, they derive specific goals and implementation strategies for the benefit of the organization as well as the teams working towards the goals.
7. **Foresight:** Foresight is the ability to foresee the most likely outcome of a situation. This empowers the servant leader to learn about the past and to achieve a better understanding of the current reality to identify consequences about the future. This characteristic is closely related to conceptualization.
8. **Stewardship:** All the stakeholders in an organization have the task to hold their institution in trust for the greater good of society. In conclusion, servant leadership is seen as a dedication to help and serve others. Openness and

persuasion are more important than control of the servant leader.

9. **Commitment to the growth of people:** Therefore, A servant leader nurtures the personal, professional and spiritual growth of employees, because she believes that people have an intrinsic value beyond their contributions as workers. A servant leader encourages the ideas of everyone and involves workers in decision-making for empowering the workforce to grow in their value chain.
10. **Building community:** Servant leaders are committed to building a strong community feeling within their organization and develop a true community among businesses and institution.

Unlike leadership approaches with a top-down hierarchical style, the emphasis of servant leadership is on collaboration, trust, empathy and the ethical use of power. At the core, the individual is a servant first, making a conscious decision to lead in order to serve others better, not to increase their own power. The ultimate objective is to enhance the growth of individuals in the organization and to increase teamwork and personal involvement for achieving a common goal. Through surveys it has been found that the organization attains several benefits by promoting servant leadership:

1. Servant leadership through the exemplary treatment of employees leads to an excellent treatment of customers by employees of the company and a high loyalty of the customers.
2. Through enhanced engagement process there is high employee identification within the organization.
3. The corporate culture is significantly improved.
4. Leaders of a company define themselves by showing their significance to the people.
5. Servant leadership is an excellent way to improve the return on investment of staff, in all economic sectors. It has been found that the managers who empower and respect their staff get better performance in return.

In summary, servant leadership is all about developing people, building great teams, thinking strategically and identifying with the values of the organization. Employees are encouraged and motivated to participate in decision-making. They should be encouraged to solve problems themselves and to grow by achieving personal success. The leader's purpose is also to create a very positive and familiar working environment. The other characteristics of a servant leader are active listening, empathy, healing and the formation of a community. A servant leader strives for continuous improvement, understands the context of current actions and implications for the future,

strategizes goals into specific action plans and holds a future-oriented view of things. The courage to conceptualize and far-sighted visions are also the characteristics of a servant leadership approach. Through the personal identification with the goals of the company the employee motivation increases which leads to increase in performance and commitment.

BRAINSTORMING TECHNIQUES

1. In these groups, members meet face-to-face and rely on both verbal and nonverbal interactions to communicate with each other.
2. Brainstorming is meant mainly to overcome pressures for conformity in the interacting group that retard the development of creative alternatives.
3. It does this by utilizing an idea-generation process that specifically encourages any and all alternatives, while withholding any criticism of those alternatives.

In a typical brainstorming session, about six people sit around a table and the group leader states the problem in a clear and understandable manner so that it is understood by all the participants involved. Members then can suggest as many alternatives as they can in a given length of time. No criticism is allowed strictly,

and all the alternatives are recorded for later discussion and analysis. Brainstorming, however, is merely a process to generate ideas. A brainstorming session requires a facilitator, a brainstorming space, a white board, a flip chart, or software tool. The facilitator should guide the entire session and encourage active participation from all involved. We need to involve participants from various departments across the organization who have different backgrounds so that they can bring fresh ideas that can inspire the experts.

BUILDING EMPOWERED TEAM

Definition of Team

An Agile team is a cross-functional group of people 100% allocated to the project and there is no partial allocation. The agile team works best when all the team members are co-located but we have tools to manage the distributed Agile team. Agile team is also self-organizing team and they do their work efficiently (Theory Y) and only facilitation is required. One-to-one communication is very critical and important for

the growth of the team members and the team as a whole.

Types of Team

Some project teams deliver wonderful results and that also without facing much stress, but some teams actually struggle to deliver correct things even working in full stress level. Though from outside both of this type of workforce look like real teams but actually they can be categorized into a real team, pseudo team, potential team and workgroup based on the definition.

Real Team

This is what an ideal high performing team should look like:

1. This is a small group of people with complementary skills
2. All the members are equally committed to a common purpose, goals and working approach
3. All the members hold themselves mutually accountable for the results

Real Team, Stage Four, Characteristics

1. Customers are consistently receiving outstanding service and product
2. Team standards are met continuously and seamlessly
3. Members are comfortable with the consistency and shared leadership
4. Team outperforms all reasonable expectations by working as a cohesive whole

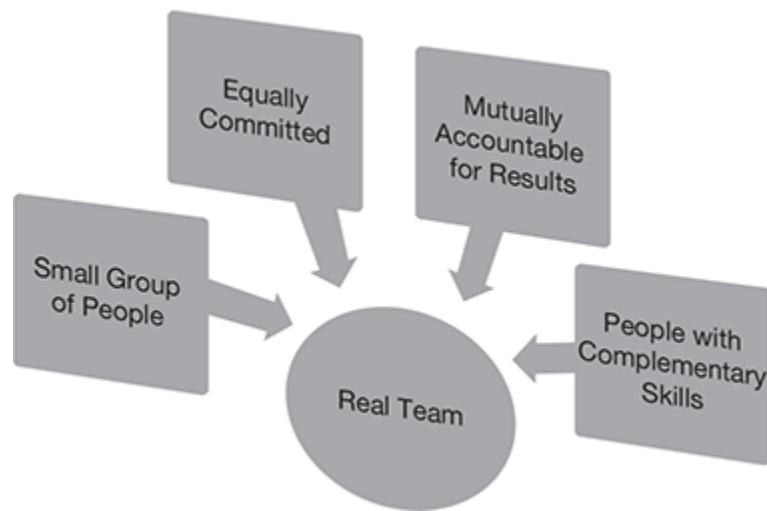


Figure 9.7 Real team

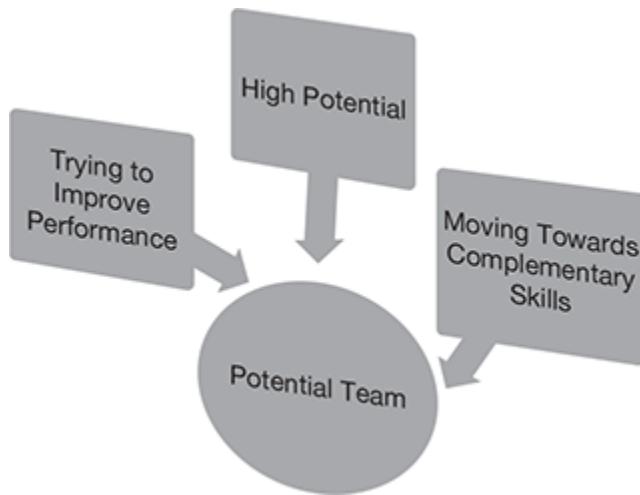


Figure 9.8 Potential team

Potential Team

1. There is a significant, incremental performance need for this group of people. It is trying to improve its performance impact by improving the collaboration.

2. If it gets more clarity about the purpose, goals or work products then it has a high potential to become a real team.

Potential Team, Stage Three, Characteristics

1. Ownership of performance standards is increasing
2. Customer focus has improved significantly
3. There exist mutual accountability and members starting to like and feel comfortable
4. More honesty and transparency among team members
5. Learning and improving from trial and error—recovery is rapid

Pseudo Team

This is the classic type of team which we generally think of as the real team, but when we look deep into the team structure and performance then it will turn out to be a pseudo team

1. For this group of people, there could be a significant, incremental performance needs or opportunity, but it has not focused on collective performance and not really trying to achieve it; rather they work individually without much collaboration
2. A common purpose or set of performance goals is missing

3. There may be individual brilliance within the group, but the sum of the whole is less than the potential of the individual parts due to lack of proper collaboration

Pseudo Team, Stage Two, Characteristics

1. A unmotivated team with a feeling that this definitely is not fun
2. Leadership and/or members are not in control of the situation
3. Everyone can feel that 'something is definitely wrong here'

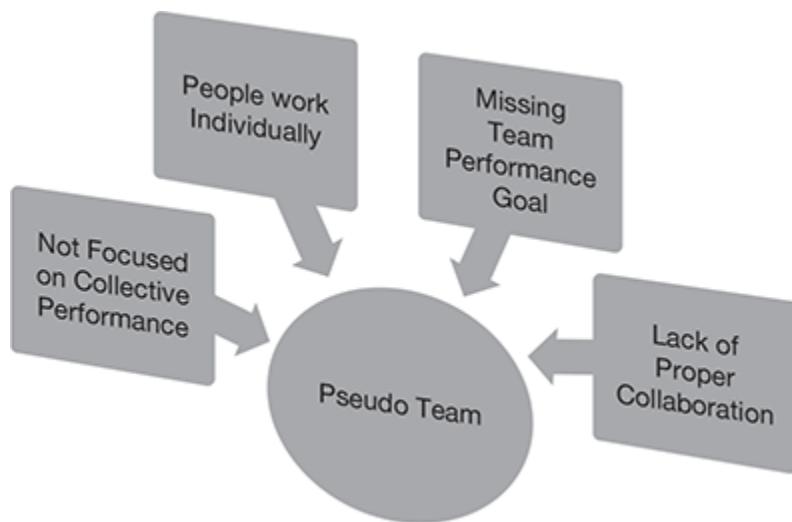


Figure 9.9 Pseudo team

1. 4. Team members are feeling uncertain and incapable
2. 5. Performance standards not being met and that leads to lots of finger-pointing, bringing down team's morale
3. 6. Customer focus is very low

Working Group

1. This is a group of people which is not feeling the need to become a team as there is no significant incremental performance needs or opportunity
2. The primary reason for interaction among the members of this group is to share information, best practices or perspectives

Working Group, Stage One, Characteristics

Figure 9.10 shows the characteristics of a working group.

1. This group has questions about what the rules really are
2. Standards are yet to be developed
3. High dependence is on the leader
4. Leader uses a directive approach to allocate and track tasks

We significantly rise up the value chain (Figure 9.11) when we move from working group to pseudo team to potential team and then to the real team.

If we identify a real team and find out what made them a real team, we will be able to

identify several categories each with certain characteristics:

Approach of Real Team

1. Members' complementary skills comes to use as needed
2. There is a shared leadership giving opportunity to people to grow
3. Members are clear on their roles and responsibilities
4. Various ideas are explored among the team before a decision arrives

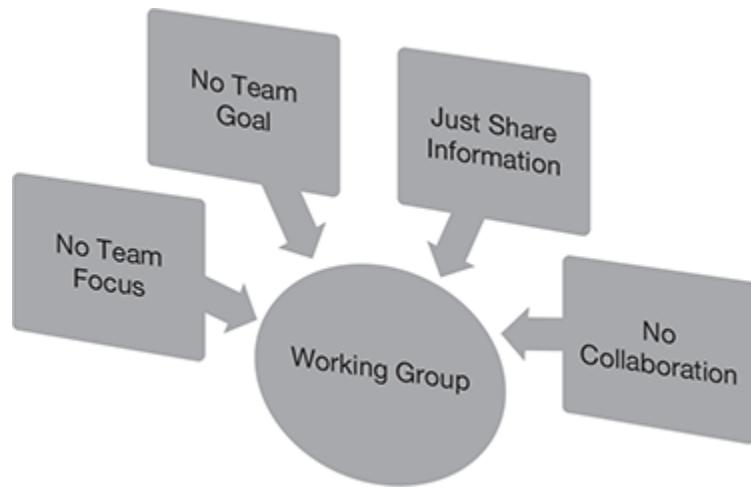


Figure 9.10 Value chain

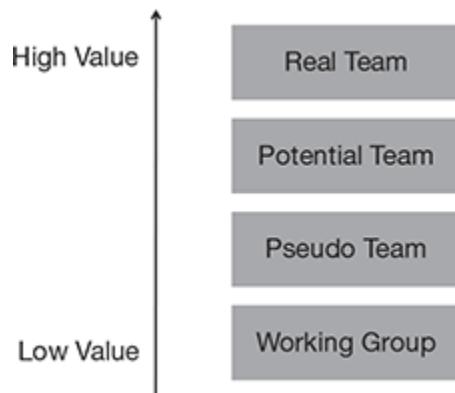


Figure 9.11 Value chain

Product Characteristics of Real Team

1. Throughput is high
2. Quality is excellent
3. Decision-making is effective and apt
4. Customer expectations have been exceeded consistently

Interpersonal Skills of a Real Team

1. Members feel comfortable about the team
2. Members are charged up and motivated
3. There are prides and confidence in the team
4. There are commitment and trust between team members

Customer Service of Real Team

1. Customer kept informed on task progress and there is overall transparency maintained
2. Customer service strategy is in place and is operating seamlessly
3. Relationship with customer is being developed and they are working as partners
4. Customer is overwhelmed with service and the product

Agile Team Stages (Shu-Ha-Ri, Dreyfus, Tuckman)

Figure 9.12 explains three different stages of the team (SHU-HA-RI) gaining knowledge and acquiring skills. First stage is to follow the rule to understand the entire gamut of the rule before breaking it. It also indicates a step-by-step approach. A team can be in one or all of these stages simultaneously.

SHU: At this first stage of learning the student follows the instruction of a single master and instead of trying to understand the underlying theory is concentrating mainly in following the rules.

HA: This is the second stage of learning when the student is learning from multiple masters and along with following the rules trying to understand the principles behind the rules.

RI: This is the matured level of learning as the student is now able to create his or her own approaches and is able to adapt to what he or she has learned according to the situations.

Agile coach stages in the similar lines are teaching-coaching-advising.

The SHU-HA-RI model is expanded in the Dreyfus model of skill acquisition. This model was proposed by the Dreyfus brothers in 1980 based on their teaching the pilots. This is based on a five-stage model with each stage increasing in skill.

Novice - Focus is on getting to know and following the rules.

Advanced beginner - Able to apply the rules according to the environment. All activities are treated separately with equal importance.

Competent - Able to perform multiple activities and handle multiple sources of information simultaneously.

Proficient - Able to get the holistic picture of the process and judge deviations from normal.

Expert - Able to use an analytical approach to apply the acquired knowledge to resolve new situations.

These principles are used in Agile coaching as group development tools.



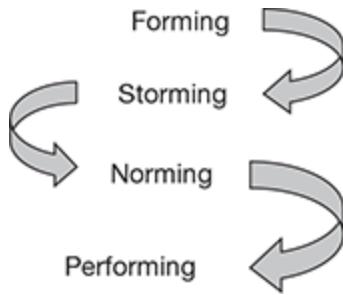
Figure 9.12 Agile team stages

War Room

The war room is a technique for team building. As part of this, the project team meets in this room daily at a specified time even if there are no issues to discuss. It actually helps to create a project identity. Agile recommends the team to work continuously in the war room itself and call that room as 'information space'.

Tools and Techniques for Team Building

According to Bruce Tuckman team-building happens through the stages of forming, storming, norming, performing and adjourning.



Forming

Team members are introduced and it is the polite stage in which the team starts to form. Everyone is trying to figure out the team concept. The team is usually positive for the initial meetings. No one will offend anyone at this point.

Storming

The team starts to move from 'as is' to 'to be'. The silent leaders may be clashing for control of the group and there will be a lot of storming and hence the name. People disagree and may even blame the team concept, saying it does not work and will not work. Management put efforts to overcome their differences, may even take separate one to one sessions with people, if required.

Norming

The team reaches consensus on the ‘to be’ process and they learned to work together, and has turned around from the storming phase. When issues crop up, the team will bounce back and forth between ‘storming’ and ‘norming’. When a new member is added they will move to the ‘forming’ stage.

Performing

The team is at high-performance level and very seldom they fall back into the ‘storming’ phase. At this level, the team automatically takes new work, and selling it to other teams as they are high performers.

Adjourning

The team shares the improved processes with others and many relationships formed within these teams continue long after the team disbands as they are high performance team.

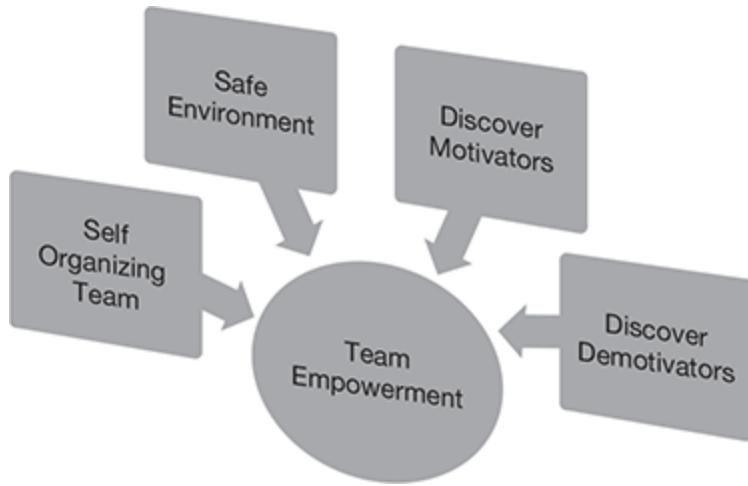


Figure 9.13 Team empowerment

Team empowerment (Figure 9.13) is a largely misunderstood concept in the corporate world. The real essence of empowering the team is to allow people at all levels to broaden their scope of decision-making to fully utilize their talents, skills and inherent creativity. Research suggested that empowering workers widen the boundary of individual and team contribution.

The agile framework provides special stress on the empowered team as only empowered teams can work towards self-direction which is the core of Agile values. Encourage emergent leadership within the team by establishing a safe and respectful environment in which new

approaches can be tried in order to make improvements and foster self-organization and empowerment. The project manager has a vital role in creating the empowered team. The concept of the role of project manager in Agile is quite different from the traditional models. So let us make our understanding clear about it.

It is the process of improving competencies, team interaction and the overall team environment to improve project performance, empowering the team to self-organizing around the work. Team work is a critical factor for project success and is achieved by open and effective communication, developing trust, encouraging collaborative problem-solving. Continuously discover the team and individual motivators and de-motivators in order to ensure that the team is continuously motivated throughout the project. Practice servant leadership by supporting and encouraging others in their endeavors so that they can perform at their highest level and continue to improve.

COACHING AND MENTORING

Every team needs some coaching and mentoring, no matter how long the team members are working together. Mentoring transfers the Agile knowledge and experience to the team and it may also come from outside the team. The context of Agile makes it a mentor, whereas the focus on the team makes it a coach. Figure 9.14 shows the difference between managing and coaching. Coaching happens simultaneously at two levels: individual level and the team level.

In the beginning of the sprint team level, coaching will be at high and individual coaching takes a back seat. In the middle individual coaching takes a front seat.

Agile coaching is 40% doing and 60% being. Coaching tones consist of loving, compassionate and uncompromising.

Managing	Coaching
Enhance individual	Enhance team performance
Directing the team	Letting the team find their own way
Fixing problems	Taking problems to the team
Talk of deadlines	Talk of value

Figure 9.14 Managing vs coaching

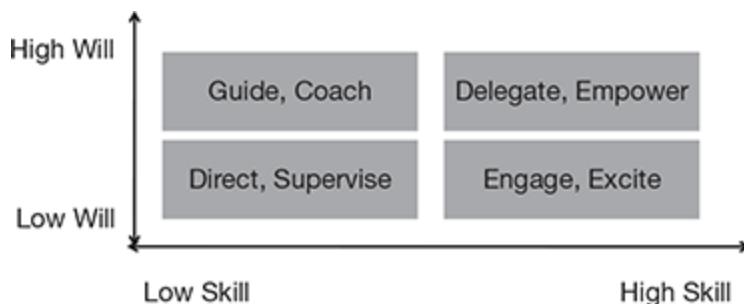


Figure 9.15

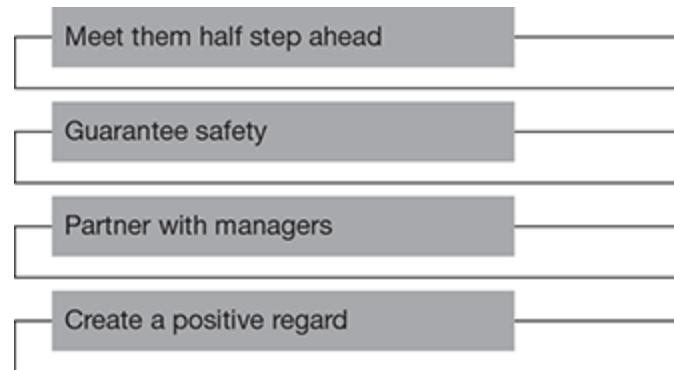


Figure 9.16 One-on-one coaching

Coaching and Mentoring Style

The coaching or mentoring style varies widely depending on the willingness of the team to learn new skills and the existing skill level of the team. Figure 9.15 shows the coaching style that is preferred based on the above two parameters.

Figure 9.16 describes the step to coach one to one. A coach needs to meet them half step ahead, meaning understanding the level of the person and guarantee safety to them, otherwise, they may not come and talk to them the next time. A coach also needs to create a positive regard so that things move out smoothly. The coach also needs to partner with managers to promise and take the correct decision.

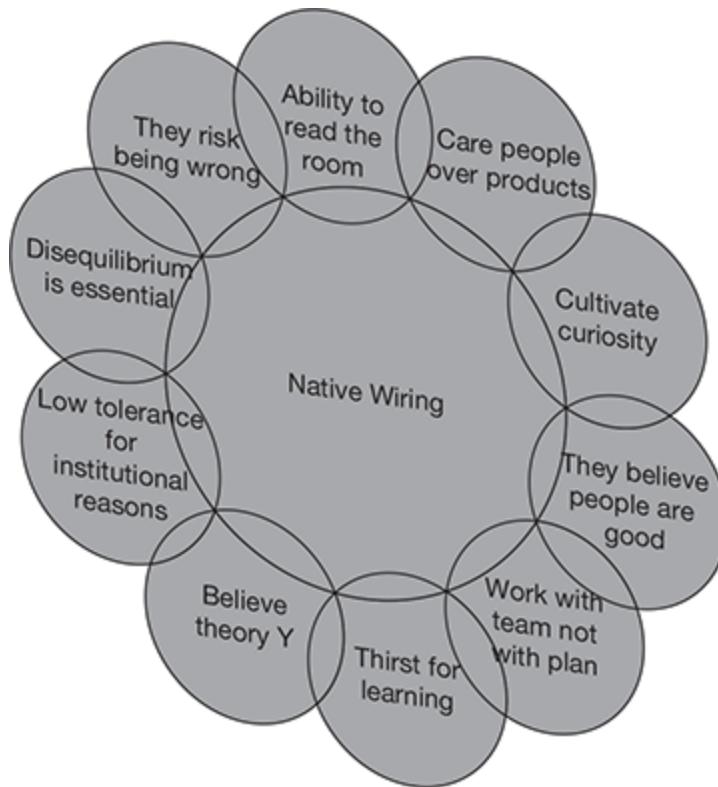


Figure 9.17 Native wiring

Common characteristics or qualities of Agile coach are termed as Native Wiring. Figure 9.17 depicts the various native wirings that can depict the success of an Agile coach.

The following table (Table 9.1) presents the different situations when coaching and mentoring is initiated for a team along with the approach that is best suited for such type of mentoring:

Table 9.1 Reason to coach and mentor

Reason to Coach and Mentor	Actions to Take
Skills Building: Take opportunities for new skills to be learned and practiced.	<ul style="list-style-type: none"> Break up large-scale tasks into smaller ones for planning coaching and mentoring to gradually introducing new skills. First coach and mentor your staff to identify performance targets they want to achieve and then select a training program aligned to that.
Progressing Projects: Oversee progress and monitor any problems in the projects.	<ul style="list-style-type: none"> Have a plan to link coaching and mentoring sessions with progress reports over the life of the project. Focus on the problems that could hinder the successful completion of the project.
Developing Careers: Guide team members to identify a clear career path.	<ul style="list-style-type: none"> Work on coaching and mentoring goals that could result in recognition of staff achievements. Long-term projects that are challenging and bring out potential are more congenial to career growth than small-scale jobs. So keep the focus on large complex engagements.

Reason to Coach and Mentor	Actions to Take
Solving Problems: Do not directly provide the solution. Guide team to identify problems and possible routes to a solution.	<ul style="list-style-type: none"> Encourage the team to define the problem and to come up with its own route to a solution. Provide guidance to the team's difficulties, while encouraging them to deal with problems robustly.
Brainstorming: Encourage the creative input of the team to keep projects on track.	<ul style="list-style-type: none"> Emphasize on the generation of creative options rather than getting bogged down with problems. Lead by example, take a lead by offering creative ideas of your own, and then invite the team to assess them.
Overcoming Conflicts: Resolve disagreements among team members.	<ul style="list-style-type: none"> Coaching and mentoring plays a great role in developing greater insights into others' perspectives and therefore avoid misunderstandings.
Re-motivating Staff: Restore enthusiasm and commitment of the team.	<ul style="list-style-type: none"> Take consideration of people's needs and aspirations and link these to performance targets. Be prepared to dig into the issues that really concern the employee and be open to views.

We will see later in this chapter how the project manager is perceived as a ‘guru’ for Mentor or Agile projects whose one of the primary responsibilities is to coach and mentor the team members.

Failure Modes and Alternatives

Agile projects fail due to various reasons. The most common of those are as follows:

1. Developers not collaborating on stories
2. Emphasizing iterations over stories
3. Accepting incomplete work at the end of the iteration
4. Lack of discipline
5. Not continually keeping design and code optimally clean
6. Not reflecting and adopting every iteration
7. Treating Agile as a checklist
8. Blaming the process
9. Not making available a clear, single-voice customer

Failure Modes

Mode	Remarks
The spy	Observe without notice
The seagull	Well intentioned observation but flies away again
The butterfly	Impart wisdom team to team
The admin	Middleman for logistics
The hub	Center for communication
The nag	Reminds the team about commitment
The opinionator	Express their opinions
The expert	Too much involvement

The above table describes various failure modes of an Agile coach and its nature. Following is a more detailed explanation for each mode.

1. **The spy mode:** In some of the projects, the Agile coach tends to play the role of a spy rather than coaching and helping the team. They try to act as the representative of some higher management who want them to do the watchdog role. This kind of Agile coach mode may not work well and the team may not gel with these kinds of people; thus understanding the pulse of the team over a period of time will become difficult. The team may not

trust this kind of people and will not share sensitive information as these people are perceived as a spy rather than a coach.

2. **The seagull:** This is the role of the traditional project manager. They will manage by exceptions and will only come into the picture whenever there are high escalations. They will not meet them daily and will not even know the exact status and feelings of the team. Like a seagull, they will fly back to their place (room) once the problem gets solved.
3. **The butterfly:** They impart wisdom to the team but will not help the team during difficult times. The butterfly role will fly inside the team only when the role needs it and not the other way around. In such cases, the wisdom may not be all that helpful in solving the existing problem within the team.
4. **The administrator:** They actually start doing the role of the operation manager and start doing all the operational jobs rather than doing the team coach job.
5. **The hub:** They actually start playing the role of the post office and act as the center for communication.
6. **The opinionator:** They will give only their opinion rather than trying to help the team solve their problems. Their opinion may be limited to their knowledge and may not be related to the problem at hand.
7. **The expert:** Too much of involvement is the problem here, which may not give space for the team to learn, and the team productivity and commitment may not improve over a period of time because of this. The coach may be an expert, but they should create an environment for the team to solve their problem themselves.

Success Modes

Mode	Remarks
------	---------

The ear	Hears everything from team
The magician	Revealing what team is not able
The child	Curious about everything
The heckler	Keep it fun
The wise fool	Ask dumb questions
The creeping vine	Make small moves
The dreamer	Think and voice the future
The megaphone	Make sure all voices are heard

The table above gives various success modes of Agile, which helps the team to perform their job.

1. **The ear:** Hears everything from the team before giving any opinion and solutions.
2. **The magician:** Try to help the team by doing something which is not usually possible with the team.
3. **The child:** The Agile coach needs to be curious about everything. The curiosity needs to be reflected in the voice, body language, and action.
4. **The heckler:** Even though they have a serious job of developing products in hand, the Agile coach maintains an element of fun so that things can be achieved quickly and effortlessly. That will help the team to stay focused.

5. **The dreamer:** The Agile coach always should think and dream about the future asking a lot of questions and plan and actions towards it. This also helps along the principle of continuous improvement.
6. **The creeping vine:** Always make small moves for continuous improvement rather than a big bang approach.
7. **The megaphone:** The Agile coach needs to ensure that all the voices are heard properly. The difference between the ear and megaphone modes is that in the former the coach hears everything from the team, whereas in the latter the coach hears from everybody to gain a wholesome perspective.

Alternative Success Modes

The below table compares the failure and success modes and points out the most suitable alternative success modes for each failure mode.

Failure Mode	Alternate Success Mode
The spy	The ear, the child
The seagull	The heckler, the creeping vine

Failure Mode	Alternate Success Mode
---------------------	-------------------------------

The butterfly	The megaphone
The admin	The dreamer
The hub	The dreamer
The nag	The wise fool
The opinionator	The megaphone
The expert	The creeping vine

TEAM MOTIVATION AND KNOWLEDGE SHARING

A true team shares knowledge and becomes motivated while expanding their knowledge. If you take away the sharing of knowledge, the team will no longer be a team, members will no longer be knowledgeable and constructive motivation will be gone.

True teams evolve into a knowledgeable workforce, which is highly effective for leaders who allow team members to be stronger than themselves. For an organization, the employees are an investment and thus additional training

increases their value. By using the investment concept, many opportunities become available for increasing the efficiency of the team. But if employees are considered as an investment, that is, 'training is not cost effective', then the opportunity for increasing efficiency becomes limited.

The philosophy of empowering the team with knowledge is a direct manifestation of the leadership approach followed in an Agile framework. In the traditional model, many leaders limit team's learning opportunity because they want undisputed control. They want employees to follow orders and not explore options for easy control. Control policies make it difficult for talented employees to be recognized because they want to show the independent thinking power in their work. Providing the team with responsibility makes early recognition possible because coworkers are first to recognize talent. This is why companies with true empowerment programs nourish the talent much better.

The teams which share knowledge will increase their knowledge while motivating others. The teams that only take orders learn nothing and over a period of time become demotivated. Leaders, who only give orders, inspire no one. Today's technology is a world of the vast pool of information. Inside the Agile framework, all the time team information radiators are encouraged. Team collaboration through knowledge sharing is the heart of Agile. The ability to acquire new skills, as new technology moves into the marketplace, is the secret to being a leader. To take advantage of emerging trends, team education in the workplace can supply the needed knowledge rather than a classroom training.

Team Empowerment

Task 1: Encourage team members to become generalizing specialists in order to reduce team size and bottlenecks, and to create a high-performing cross-functional team.

Task 2: Contribute to self-organizing the work by empowering others and encouraging

emerging leadership in order to produce effective solutions and manage complexity.

Task 3: Continuously discover team and personal motivators and demotivators in order to ensure that team morale is high and team members are motivated and productive throughout the project.

Team Motivation

There are lots of motivational factors for the team to work on an Agile project. An Agile coach (facilitator) needs to ensure that the following factors are taken care of.

1. Variety of work
2. Build trust
3. Be realistic
4. Create good environment
5. Team focused reward
6. Employee participation in planning
7. Immediate feedback
8. Team visibility
9. Immediate recognition
10. Career progression

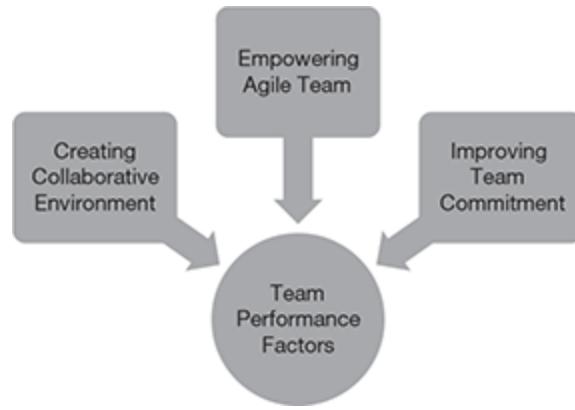


Figure 9.18 Team performance factors

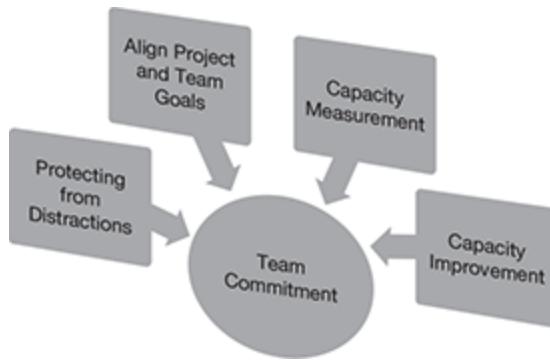


Figure 9.19 Team commitment

Figure 9.18 illustrates the team performance factors which will motivate the entire team.

Team commitment can be improved by aligning the team goal with the project goal and protecting the team from distractions. Team capacity can be measured and can be improved

thereby slowly improving the team commitment (Figure 9.19).

LEADERSHIP STYLES

1. **Directing:** Telling others what they need to do
2. **Facilitating:** Coordinating only with the input of others
3. **Coaching:** Instructing others clearly
4. **Supporting:** Providing assistance along the way to do work
5. **Autocratic:** Making decisions on own without input
6. **Consultative/Participative:** Inviting ideas from others before taking decision
7. **Consensus:** Problem-solving in a group with decision-making based only on group agreement

PROBLEM-SOLVING STEPS

Approach for Problem Resolution in Agile

Technical investigation to find an answer to a problem is called spike.

In Agile particular stress is given on empowering the team to resolve their issues themselves. The first point is to provide visibility to the team on the issues and risks arising in the

project and then providing them the necessary guidance to resolve those. In particular, the managers should understand two things:

1. An important goal of an Agile approach is to build teams that can solve their own problems effectively (but not all the organizational ones thrown at them). This helps to develop larger numbers of people who are, in fact, problem-solvers.
2. To get the most value out of a team's effort, those outside the team should be prepared to get involved in more problem-solving which means impediment clearing, than they may have previously expected. This will enable teams to focus on meeting iteration functionality expectations.

For this reason, the manager's role is more like a guide; he/she would not jump in to solve the problem even if he/she knows the solution. Having the team do it enables them to manage their own environment (to the extent they can). Whether you are working in Agile or otherwise, relying on single individuals to be 'problem-solvers' may mean that you set up a number of single-points-of-failure should such people leave, be transferred, etc. Figure 9.20 shows four simple steps to solve a problem in an Agile Project. First, detect the correct problem and

reflect on it. Take the problem to the team without trying to solve it yourself.

The practice of daily stand-up meetings in Agile plays a great role in enhancing the team's ability to surface problems and collectively reduce the risk of wasted effort by sharing progress information to the entire team members. Daily meetings provide everyone on the team an opportunity to see the status of all aspects of the project in real time. This allows the collective thought process of the team to fine-tune or redirect efforts on a daily basis to maximize throughput. This results in radical alteration of the software development process by allowing sharing of software resources. Development tasks which normally take days could often be accomplished in hours by using someone else's code as a starting point.

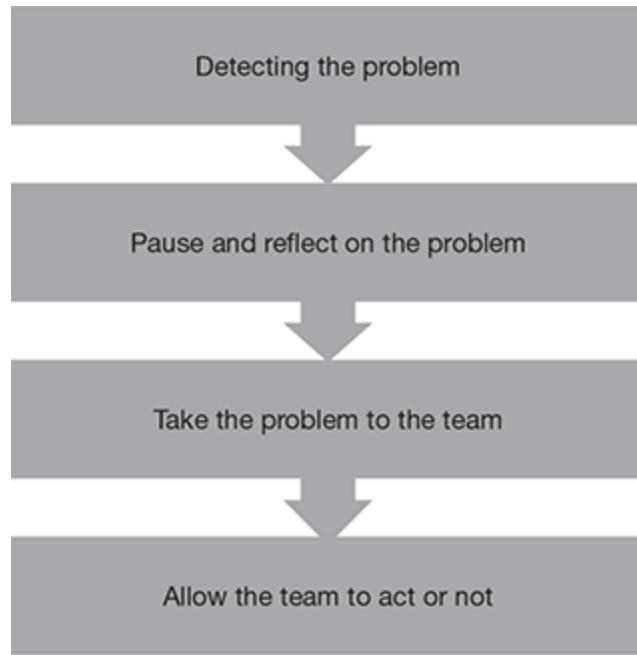


Figure 9.20 Problem solving rubric

Problem solving tips in Agile Projects:

1. Create an open and safe environment by encouraging conversation and experimentation in order to surface problems and impediments that are slowing the team down or preventing its ability to deliver value.
2. Identify threats and issues by educating and engaging the team at various points in the project in order to resolve them at the appropriate time and improve processes that caused issues.
3. Ensure issues are resolved by appropriate team members and/or reset expectations in light of issues that cannot be resolved in order to maximize the value delivered.
4. Maintain a visible, monitored, and prioritized list of threats and issues in order to elevate accountability,

encourage action, and track ownership and resolution status.

5. Communicate the status of threats and issues by maintaining the threat list and incorporating activities into the backlog of work in order to provide transparency.

Team Communication to Avoid Problems

From the beginning, Agile framework was designed to provide transparency to both team members and those outside the team. Refining the approaches for management reporting is an ongoing focus area for Agile community.

A typical Agile project maintains a Kanban board showing columns of user stories, development tasks relating to each story, and the state of each development task and tests associated with each story. As the task progresses from design to code to verification, the associated card is moved from one column to the next and finally to a done column. An updated burn-down chart along with a prioritized list of impediments is often posted as well. A manager can walk by a Kanban board and see the status of the team in a few seconds. If an impediments list is posted, a manager can add

relevant items to his or her worklist. The risk burn-down chart along with the impediments chart will provide the management idea of which areas to be focused on at the earliest. This eliminates the need for most status reporting, particularly if critical information is put online on a web page, a wiki or a reporting tool.

The executive management gets transparency into all operations by viewing important indicators quickly. Surprises should be avoided, as in software a surprise is rarely a pleasant one. We all know that projects do backfire; executives know this and so does everyone else. It is always a surprise the first time one hears bad news like delays or unforeseen risks. But the kind of surprise executives hate the most are the ones that have a significant impact and were known much earlier than when they were finally informed. This actually means that the decisions were made on faulty information and this was preventable. By use of the information radiators, this problem can be easily avoided as everyone in the project gets a clear view of the progress, risk, impediments, etc. through a visual tool.

Taking the problem to the team can be done in numerous ways but we need to bear the following in mind.

1. Address it directly
2. Reaffirm Agile
3. Reveal the system to itself
4. Use the retrospectives
5. Add a revealer

Self-organizing Team

Agile methodology is based on a different philosophy—the self-organizing team. The difference is apparent from the first step the team takes during planning. The team decides how much to commit to in each iteration. And for doing that the team decides on the estimation of the work items together. Research has shown that the team morale is significantly enhanced and the team feels more committed to the work when the team themselves decide how much to commit to and when this commitment is realistic and achievable. The team is empowered to take important decisions in the project and to stand by the decisions all along.

The next aspect of self-organization happens during the iteration cycles. Together the team decides who will work on which work item and make sure that all the work is completed. When the team is responsible for decision-making, they remain focused on the commitment because they own it; if the commitment is to be completed then they are the ones who must do it. On the other hand, if someone outside the team (the manager) is responsible for deciding who will be doing what then the team gets a subtle but clear indication that they are not the ones who are responsible; but it is the responsibility of the manager to figure out how to meet the commitment. In Agile, the manager's role is to support the team to achieve its target.

A classic example of the manager helping the team to become self-organizing rather than directing them what to be done is here: The manager Marry has entered the room where the daily stand-up meeting of the team was taking place. The team was deciding on the architecture of the new version of their product and they are planning to start coding on that day based on this architecture. All the components of the design

are identified and noted into small sticky notes on the wall. The team asked for the approval of Marry as she is a veteran in designing large complex modules. By looking at the components Marry could readily find out that a very vital component was missing which will cause a lot of rework from the second week unless it is incorporated into the design from the beginning. But she resisted the urge to tell the team directly what is missing, but she wanted them to find out themselves. She said 'Look guys, the design looks great; but a vital component is missing which will create a problem during integration with system X. I know you all are capable enough to find it. So I am going to grab a cup of coffee and come back in 10 minutes. I hope you will be able to figure out what is missing'. Now the team has a clue, they started thinking hard, examining and brainstorming each integration component. And suddenly they found out that a bridging component is not present to talk to a third party system. And what is more is that they changed one inefficient component while reviewing all the components for this exercise. When Marry came back they were proudly ready with their answers. So, the kind of leadership

Marry has shown here is more of a ‘guru’ than a ‘nanny’. She provided the team with the necessary guidance to achieve the goal, but she did not tell them the ready solution. This feels empowered and becomes more cohesive, committed and focused due to their greater involvement in the decision-making.

This is a paradigm shift for the organization culture and to make this successful there should be buy-in from the executives in the organization. A manager in Agile organization should have the following job responsibilities:

1. Help remove impediments that the team is not able to resolve by themselves
2. Provide guidance and input to the team on technical difficulties that come up
3. Hold regular one-to-one meetings with team-members to provide coaching and mentoring
4. Give input to features enhancement
5. Stay abreast of developments in tools, technologies and techniques that the team is working with
6. Plan training and continuous skills development for team members
7. Stay up-to-date on industry news and developments across similar organizations
8. Plan and manage budgets and financials of a project
9. Conduct performance evaluations and provide feedback to team members

10. Take up career development and career planning with team members
11. Induct new team members and remove the ones who are not able to perform

You can notice that the primary responsibilities that a manager in the traditional model has is to decide on what and how works need to be done, while committing to the work on behalf of the team is omitted in the Agile model. The manager is adding more value by involving the team for taking such decisions and in turn, is getting more commitment for the project goal.

Two types of leadership techniques that promote team empowerment are adaptive leadership and servant leadership. These concepts are quite different than the conventional leadership models. Due to the high focus of Agile methodology on the ‘people’ these styles of leadership are more popular among the Agile communities.

The concept of **High Performing Team** in Agile is based on the principles discussed above

like building an empowered team and a self-organizing team. In a survey by Quantitative Software Management Associates it was found that Agile teams are 37% faster to market and 16% more productive. This is possible because the high-performing Agile teams can deliver exceptional results over average time by efficient resolution of issues and shorten the cycle times. As discussed above, the most important factors that help to create high-performing teams are:

- Motivation and reward
- Skills
- Leadership
- Communication
- Work environment
- Clarity of purpose or goal
- Productive conflict resolution

Ground rules and their importance in team building

While working in a team, it is extremely important to set up ground rules, which guides the group's behavior. This will ensure that an environment is created where members can discuss difficult topics, get into constructive discussions, and free thinking is encouraged.

This helps in team building and team performing without unnecessary conflicts. Establish clear expectations regarding acceptable behavior for each and every member of the team. All team members share responsibilities once the ground rules are established clearly.

Each team's ground rule may be different and these can be modified based on what works for the team and what not. A few examples of ground rules can be:

1. During daily stand-up nobody gets engaged into cross-conversation.
2. After each code check-in the happy path testing script needs to be run by the developers.
3. During an ongoing Sprint, no change to the Sprint backlog will be allowed. The items will be added to the product backlog to be re-prioritized in later Sprints.

Most of the ground rules are set or modified during the retrospective meetings. After working together for one or more iterations, the team can take a call on the rules they want to change as those are working for the team. Some new rules may also get added if there is a need felt. So, it is important to keep the ground rules as a live

document and update it as and when there is a need.

Discussion Points

1. What are all the ground rules set for your project? Discuss about the advantages of the ground rules.

Constellation

It starts in an open space. Put an object in the middle of the floor (Figure 9.21). Ask the team members to stand around it at a distance. Read out a statement and if they agree with the statement, they need to move closer to the object as much they feel the statement is true. If they feel the statement is false then they need to move

away from the object. This helps to understand the pulse of the team.

Power

Power is the ability to influence the behavior of another. Understanding power and the use of power can influence the outcome of the project. The project manager needs to understand his power in the organization and also needs to understand the power of stakeholders in the project in order to utilize it for the success of the project.

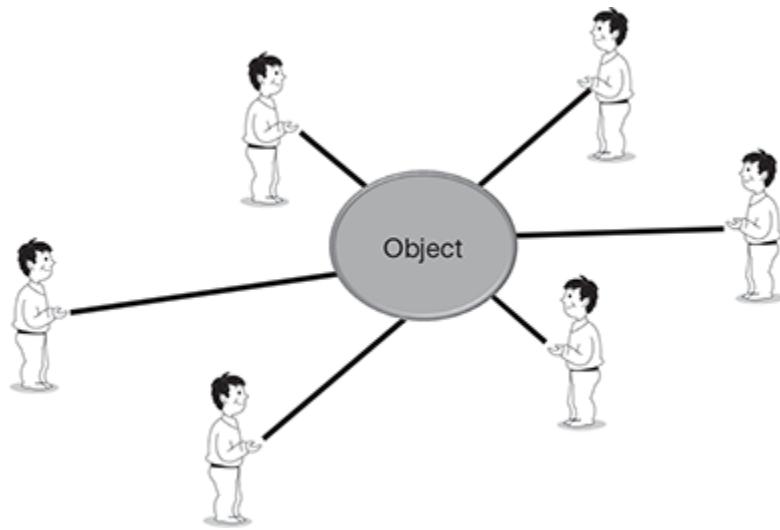


Figure 9.21 Constellation

Forms of Power

Based on ways to influence the people, there are different types of power.

1. **Reward power** can be gained from one's capacity to reward compliance and is used to support legitimate power. When someone is rewarded through recognition, a good job assignment, a pay rise, the employee may respond in kind by carrying through with orders. Rewards often comprise financial remuneration but can also be intangible as well.
2. **Coercive power** is the opposite of reward power and is the ability of the power holder to remove something from a person or to **punish** them for not conforming when a request is made and here also the employee may respond by carrying out the orders given due to fear of punishment.
3. **Legitimate power** comes from the authority of your rate and position in the chain of command and it is the legal power. Example: The project manager does some favor or unsavory act just because he is the project manager.
4. **Referent power** derives from your subordinates' identification or their association with you. You have this power by simply being 'the boss' and people identify with the ideals you stand for.
5. **Expert power** comes from your knowledge in a specific area through which you influence others.

CULTURE AND TEAM DIVERSITY

Cultural diversity and differences within a distributed team are inevitable. The business and workplace norms may vary significantly across locations causing uneasiness among the team. On the broader perspective, there is a sharp contrast in Western and non-Western values. The Westerners value the individualism, assertiveness, equality, informality, achievement while the non-Westerners value collectivism, indirectness, hierarchy, formality and modesty. So aligning the whole distributed team on the same cultural note should be the focus for creating a high performing team.

Cultural Alignment

Team work is the key enabler of any Agile project. It is very important that at the beginning of the project you plan to bridge the cultural difference between the various teams like, onsite development team, client team and the offshore development team so that the mutual trust is built between the teams.

One of the most critical steps for the success of a distributed Agile project is for the

distributed teams to come together in person at the beginning and spend quality time sharing key project information and building a relationship with each other. Following consideration are important to achieve this:

1. Significant investments by HR to impart cross cultural training
2. Visit by client team leads to development team location for initial setup and introduction
3. Key people will have significant onsite experience

Build the Team over Time

Real teams take the time to form, and it will work best if they stay together on multiple projects over long periods. One of the challenges of distributed teams is that this formation process gets extended by reduced face-to-face communication among the team members. As a result, new team members take longer to ramp up on the team's practices as well as the customer problem domain. It also takes longer for team members to get to know each other on the personal level that is required to build a healthy and open team environment for really effective work. That is the reason, reorganizing

distributed teams is even more disruptive than a co-located team.

Team formation Tasks

Cooperate with the other team members to devise ground rules and internal processes in order to foster team coherence and strengthen team members' commitment to shared outcomes.

Help create a team that has the interpersonal and technical skills needed to achieve all known project objectives in order to create business value with minimal delay.

There are additional items that you need to be aware of while working with a distributed Agile team:

1. Agile teams often do not produce a lot of formal artifacts, like specifications and design documents, which new team members could use to learn the relevant information from and get to speed quickly. So, special attention must be given to new team members who are working remotely when they join the project to understand their orientation need. Consider assigning them a buddy or mentor to help them get up to speed. Distributed teams may produce additional artifacts like wiki pages to document designs

and processes for ensuring knowledge flow continuity to the new team members and for helping new team members get up to speed.

2. A core team is your biggest asset—building teams from a set of core members who will be on a project for several releases will help. Adding other team members who are there for shorter periods will then be less disruptive as they can leverage the learning from the team’s core members. Over time, the composition of the core team may also slowly change. For each project avoid starting teams from scratch with no continuity guaranteed between projects.

Exercise 9.1 List the activities to develop a project team? (You just acquired the team)



Discussion Points

1. What type of power did you use in what type of situation? Discuss the implications of using other types of powers in the same scenario.

Exercise 9.2 Identify and match the types of powers as depicted in the following activities

Functional manager assigns an architect to work on your team	Reward power
You create a policy to award people who finishes their jobs earlier	Legitimate power
Lots of other project managers asks doubts to you about project management as you have very good knowledge and skills in it.	Punishment power
A project manager says that there will be 10% cut in our salary as we did not meet our profit target last year	Expert power

Discussion Points

1. List and discuss the characteristics of effective project teams. How will you develop the characteristics of an effective project team?

Discussion Points

1. Discuss the conflicts you managed in your recent project.

Summary

This chapter discussed the importance of Soft skills that are essential for Project managers. Since Agile projects are mostly people driven soft skills are necessary and we need skills like conflict management, negotiation, Leadership skills, and interpersonal skills to deal with people. Emotional Intelligence doesn't mean being soft—it means being intelligent about emotions—a different way of being smart.

Emotional intelligence is your ability to acquire and apply knowledge from your emotions and the emotions of others in order to be more successful and lead a more fulfilling life. Emotional Intelligence is the ability of an individual to deal successfully with other people, to manage one's self, motivate other people, understand one's own feelings and appropriately respond to the everyday environment. The sum of the

individual parts is called collaboration. Collaboration means the whole is greater than the sum of the individual parts. The collaboration includes innovation, astonishing results. This chapter also discussed the concepts of Adaptive Leadership concepts, Servant Leadership concepts in addition to emotional Intelligence, concepts.

Answers to Exercise Questions

Exercise 9.1: Possible Answers

- | |
|--|
| Hold team building activities for team members |
| Training for team members where needed |
| Establish ground rules for team member behavior. Set clear expectation for them. |
| Create and give recognitions and rewards. |
| Take all possible steps to correct mistakes, if any. |
| Place team members in the same location; co-location, if possible. Arrange for regular team meetings periodically. |

Exercise 9.2: Answers

Functional manager assigns an architect to work on your team	Legitimate power
You create a policy to award people who finish their jobs earlier	Reward power
Adding Resource during Escalation	Expert power
A project manager says that there will be 10% cut in our salary as we did not meet our Profit Target last year	Punishment power

Chapter 9 Questions and Answers

- Please set yourself a time clock of 1 hour to take this test
- Mark your answers using pencil in the answer sheet provided at the end of this question set
- The correct answers are provided at the end of this question set (after the answer sheet)
- Give one mark for each correct answer for evaluation purpose
- There is no negative marking for the wrong answers
- Practice this test multiple times for better results
- All the very best!

Question 1. Servant leadership can be mapped to which of the leadership style?

1. Autocratic
2. Persuasive
3. Participative
4. Mentoring

Question 2. Following are the examples of coaching tones except

1. Loving
2. Compassionate
3. Uncompromising
4. Compromising

Question 3. The ability to influence the behavior of other is called as:

1. Politics

2. Emotions
3. Power
4. Mentoring

Question 4. Which of the below statements is NOT correct about ground rules:

1. These rules are set for the project duration and should not be changed
2. These help in team building
3. If any rule is not working for the team then it can be changed or dropped
4. The team decides on which rules work for them and which not

Question 5. The work that a project team member is expected to perform in order to complete the project activities is called

1. Accountability
2. Responsibility
3. Role
4. Authority

Question 6. Which of the following is not a collaboration mode?

1. Accommodative
2. Competing
3. Compromise
4. Responsive

Question 7. Which of the following is not a real team's characteristic?

1. Customers are consistently receiving outstanding service and product
2. Team standards can vary based on the complexity of the project
3. Members are comfortable with the consistency and shared leadership
4. Team outperforms all reasonable expectations by working as a cohesive whole

Question 8. What is normally the reason behind a working group to interact with each other though they are not working as a team?

1. As the management mandates to work together
2. They want to become a team in future
3. The primary reason for interaction among the members of this group is to share information, best practices or perspectives
4. The interactions are mostly for personal needs

Question 9. Following are the stages of team development except

1. Forming
2. Norming
3. Adjourning
4. Closing

Question 10. You are the project manager of a complex project and the project is running for

the past 2 months and team members begin to work together and adjust work habits that support the team. They are in which phase of the team development?

1. Forming
2. Storming
3. Norming
4. Performing

Question 11. You are the project manager of a complex project and the project is running for the past 2 months and team members are interdependent and work through issues smoothly and effectively. They are in which phase of the team development?

1. Forming
2. Storming
3. Norming
4. Performing

Question 12. All of the following are the indicators of team effectiveness except

1. Reduced staff turnover
2. Increased team cohesiveness
3. Improvements in competencies
4. Increased staff turnover

Question 13. Conflict is inevitable in a project environment

1. True
2. False

Question 14. Conflict if managed properly can lead to increased creativity and better performance. Conflict should be addressed in all of the following manner except

1. Early
2. Usually in private
3. Using in direct approach
4. Using collaborative approach

Question 15. Which of the following are the characteristics of conflict except

1. Openness resolves conflict
2. Conflict is an individual issue
3. Conflict is a team issue
4. Conflict should focus on the present

Question 16. The success of project managers in managing their project teams often depends a great deal on

1. Their ability to resolve conflict
2. Their ability on negotiation

3. Their leadership ability
4. Their technical ability

Question 17. Factors that influence conflict resolution methods include all of the following except

1. Time pressure for resolving conflicts
2. Relative importance and intensity of the conflict
3. Position taken by players involved
4. Your style of conflict resolution

Question 18. Your project is about to be moved to UAT in next 2 days. Two key team members are fighting with each other about the environment issues in UAT production server. You are being called by them at night 12:00 clock, which of the following indicates your position?

1. Time pressure for resolving the conflict
2. Relative importance and intensity of the conflict
3. Position taken by players involved
4. Motivation to resolve conflict on a long-term or a short-term basis

Question 19. Your project is about to be moved to UAT in next 2 days. Two key members are fighting with each other about the environment issues in UAT production server. You are being

called by them at night 12:00 clock and you are trying to incorporate multiple viewpoints from other team members and insights from differing perspectives and trying to lead consensus and commitment. Which of the following techniques can you use?

1. Confronting
2. Collaborating
3. Smoothing
4. Forcing

Question 20. Which of the following conflict resolution technique will give best result for the project manager?

1. Collaborating
2. Compromising
3. Smoothing
4. Confronting

Question 21. Your project is about to be moved to UAT in next 2 days. Two key team members are fighting with each other about the environment issues in UAT production server. You are being called by them at night 12:00 clock and you are trying to emphasize the areas of

agreement rather than areas of difference. Which of the following techniques can you use?

1. Confronting
2. Collaborating
3. Smoothing
4. Forcing

Question 22. Ability to influence other people is called

1. Politics
2. Power
3. Conflict
4. Negotiation

Question 23. The ability of the power holder to remove something from a person or to punish them for not conforming when a request is made is called

1. Coercive power
2. Reward power
3. Legitimate power
4. Referent power

Question 24. Mahesh is the project leader of project ABC and he did his job excellently and received STAR of the year award from CEO of that company and he also got promoted and

appointed as project manager for another high priority project, this actually represents

1. Super effect
2. Halo effect
3. Recognition effect
4. Start effect

Question 25. Mahesh is the project leader of project ABC and he did his job excellently and received STAR of the year award from CEO of that company, and he also got promoted and appointed as project manager for another high priority project. Mahesh is taking decision on his own even if team members are not accepting it and this represents

1. Autocratic
2. Legitimate
3. Expert
4. Referent power

Question 26. Mahesh who is having 25 years of experience is having difference of opinion with Mohan who is 20 years of experience about testing process of your project. They are approaching you for a solution. You are asking

Mohan to accept what Mahesh says as he is more experienced. Which technique you follow here

1. Withdrawal
2. Problem-solving
3. Forcing
4. Escaping

Question 27. In SHU-HA-RI concept of gaining knowledge and skills SHU stage means:

1. At this stage the student tries to read the concepts and understand
2. At this stage the student can ask the master questions about the underlying theory of the rules
3. At this first stage of learning the student follows the instruction of a single master and instead of trying to understand the underlying theory is concentrating mainly in following the rules
4. At this stage the student will concentrate on the rules but can try to explore exceptions as well

Question 28. In the Dreyfus model of skill acquisition, which are the five stages?

1. Novice, learner, competent, proficient, expert
2. Novice, advanced beginner, competent, skillful, expert
3. Novice, advanced beginner, competent, proficient, expert
4. Practitioner, advanced beginner, competent, proficient, expert

Question 29. Where in Agile are the principles like SHU-HA-RI, Dreyfus, Tuckman, etc. used?

1. To ensure Agile principles are understood by all
2. To teach product backlog creation
3. These principles are used in Agile coaching as group development tools
4. To teach 12 principles of Agile

Question 30. Which of the following is not a common failure mode in Agile projects?

1. The spy
2. The icon
3. The hub
4. The butterfly

Question 31. You as a vendor were having a conflict with the client and both of you decided to go for a third party to resolve conflict. This is called

1. Arbitration
2. Negotiation
3. Problem-solving
4. Court settlement

Question 32. Which of the following conflict resolution technique provide temporary resolution?

1. Problem-solving
2. Smoothing
3. Compromising

4. Forcing

Question 33. The ability to bring awareness to the emotions as they arise, notice them for what they are, and decide how to best use them is called:

1. Emotional intelligent quotient
2. Awareness quotient
3. Collaboration quotient
4. Ability quotient

Question 34. Which of the following statement is true?

1. Cooperation is same as collaboration
2. Innovation is an example of cooperation
3. Sum of the individual part is collaboration
4. Astonishing results is an example of collaboration

Question 35. All of the following are hallmarks of cooperation except

1. nnovation
2. Two-way conversation
3. Real understanding
4. Progress in relationship

Question 36. Which of the following is the biggest challenge of distributed Agile team?

1. Conflict management
2. Communication
3. Configuration of code
4. Negotiation

Question 37. Which of the following is an expression of radiant thinking, a natural function of the human brain, and they allow creative ideas to bloom and flow.

1. Brainstorming
2. Mind mapping
3. Interviewing
4. Future visioning

Question 38. In which of the following facilitation technique team members imagine that all their problems are resolved, dreams have been achieved and goals have been fulfilled. Step back and describe how those happen?

1. Brainstorming
2. Mind mapping
3. Interviewing
4. Future visioning

Question 39. In which of the following facilitation technique, team members imagine the problem and then imagine there were no history, rules, regulations, culture or climate. If none of

these things existed because we were just starting up what might we do? How we might approach the solution?

1. Brainstorming
2. Mind mapping
3. Green field site
4. Future visioning

Question 40. Which of the following ensures that all voices that want to be heard are heard?

1. Consent check
2. Consensus heck
3. Team check
4. Overall check

Question 41. Which of the following ensures consensus reached and team moves forward?

1. Consent check
2. Consensus check
3. Team check
4. Overall check

Question 42. All the stakeholders agrees on a single course of action is called

1. Unanimity
2. Majority
3. Plurality

4. Dictatorship

Question 43. The fifth level of conflict is called

1. Crusade
2. Contest
3. World war
4. Disagreement

Question 44. Following are the examples of adaptive approach except

1. Attention is focused on the value-added outcomes.
2. Job descriptions are intentionally broad-based on allowing for flexibility and collaboration.
3. Roles are fluid. Within limits, people are expected to play roles for one another.
4. Policies are mostly oriented towards drive and control.

Question 45. Following are the causes of conflict in Agile project except

1. Facts
2. Goals
3. Values
4. Skill level of the team

Question 46. Following are the characteristics of 'real team' except

1. Customers are consistently receiving outstanding service and product
2. Team standards are met continuously and seamlessly
3. Members are comfortable with the consistency and shared leadership
4. Not focused on collaborative performance

Question 47. Following are the characteristics of ‘Servant Leader’ except

1. Empathy
2. Conceptualization
3. Command and control
4. Persuasion

Question 48. ‘Following the Rule’ is the stage of

1. SHU
2. HA
3. RI
4. All the stages

Question 49. Following are the Agile team motivational factors except

1. Variety of work
2. Individual salary hike
3. Build trust
4. Team focused reward

Question 50. It starts in an open space. Put an object in the middle of the floor. Ask the team

members to stand around it at a distance. Read out a statement and if they agree with the statement, they need to move closer to the object as much they feel the statement is true. If they feel the statement is false then they need to move away from the object.

1. Constellation
2. Consent check
3. Consensus check
4. Conflict management

Answer Sheet for Chapter 9 Questions

Question Number	Answer	Question Number	Answer
-----------------	--------	-----------------	--------

Question 1	Question 26		
Question 2	Question 27		
Question 3	Question 28		
Question 4	Question 29		
Question 5	Question 30		
Question 6	Question 31		
Question 7	Question 32		
Question 8	Question 33		
Question 9	Question 34		
Question 10	Question 35		
Question 11	Question 36		
Question 12	Question 37		
Question 13	Question 38		
Question 14	Question 39		
Question 15	Question 40		
Question 16	Question 41		
Question 17	Question 42		
Question 18	Question 43		
Question 19	Question 44		
Question 20	Question 45		
Question 21	Question 46		

Question 22	Question 47
Question 23	Question 48
Question 24	Question 49
Question 25	Question 50

Answers for Chapter 9 Questions

Question Number	Answer	Question Number	Answer
-----------------	--------	-----------------	--------

Question 1	C	Question 26	C
Question 2	D	Question 27	C
Question 3	C	Question 28	C
Question 4	A	Question 29	C
Question 5	B	Question 30	B
Question 6	D	Question 31	A
Question 7	B	Question 32	B
Question 8	C	Question 33	A
Question 9	D	Question 34	D
Question 10	C	Question 35	A
Question 11	D	Question 36	B
Question 12	D	Question 37	B
Question 13	A	Question 38	C
Question 14	C	Question 39	C
Question 15	B	Question 40	A
Question 16	A	Question 41	B
Question 17	D	Question 42	A
Question 18	A	Question 43	C
Question 19	B	Question 44	D
Question 20	D	Question 45	D
Question 21	C	Question 46	D

Question 22	B	Question 47	C
Question 23	A	Question 48	A
Question 24	B	Question 49	B
Question 25	A	Question 50	A

Explanations for Chapter 9 Answers

1. **Answer C**

Servant leadership can be mapped very well with participative leadership style.

2. **Answer D**

3. **Answer C**

4. **Answer A**

Ground rules are for the benefit of the team and if required can be changed or new rules added.

5. **Answer B**

6. **Answer D**

The negotiation modes that are used for conflict resolution are as follows:

1. Accommodative: 100% co-operative, 0% assertive
2. Competing: 0% co-operative, 100% assertive
3. Compromise: Equal distance from co-operative and assertive
4. Collaborative: Combination of co-operative and assertive

7. **Answer B**

Real Team, Stage Four, Characteristics

1. Customers are consistently receiving outstanding service and products
2. Team standards are met continuously and seamlessly
3. Members are comfortable with the consistency and shared leadership
4. Team outperforms all reasonable expectations by working as a cohesive whole

8. *Answer C*

The primary reason for interaction among the members of this group is to share information, best practices, or perspectives

9. *Answer D*

Adjourning is the last stage and is not closing.

10. *Answer C*

11. *Answer D*

12. *Answer D*

Increased staff turnover indicates some problem in the project. Reduced staff turnover is an indicator of team effectiveness.

13. *Answer A*

14. *Answer C*

15. *Answer B*

Conflict resolution should focus on issues and hence it is a team issue and not personalities or individual issue.

16. *Answer A*

17. *Answer D*

18. *Answer A*

19. *Answer B*

20. *Answer D*

Confrontation or problem-solving is always the best way to resolve conflict and will give permanent and effective result.

21. *Answer C*

Emphasizing areas of agreement is called smoothing.

22. Answer B

Power is the ability to influence others.

23. Answer A

24. Answer B

A star project leader need not be a good project manager as both requires different skill sets. Most of the companies are doing this mistake and is called halo effect.

25. Answer A

Taking decision on own without anybody's input is called autocratic.

26. Answer C

Here you are trying to force others to accept views and hence forcing technique being used.

27. Answer C

SHU: At this first stage of learning the student follows the instruction of a single master and instead of trying to understand the underlying theory is concentrating mainly on following the rules.

28. Answer C

Novice - focus is on getting to know and following the rules.

Advanced beginner - able to apply the rules according to the environment. All activities are treated separately with equal importance.

Competent - Able to perform multiple activities and handle multiple sources of information simultaneously.

Proficient - Able to get the holistic picture of the process and judge the deviations from normal.

Expert - Able to use an analytical approach to apply the acquired knowledge to resolve new situations.

29. Answer C

These principles are used in Agile coaching as group development tools.

30. Answer B

Common failure modes in Agile projects are - the spy, the seagull, the butterfly, the administrator, the hub, the opinionator, the expert.

31. *Answer A*

Going for third party is called arbitration.

32. *Answer B*

Smoothing provides temporary resolution and the same problem can get surface any time. It is the best answer out of others.

33. *Answer A*

34. *Answer D*

35. *Answer A*

36. *Answer B*

37. *Answer B*

38. *Answer C*

39. *Answer C*

40. *Answer A*

41. *Answer B*

42. *Answer A*

43. *Answer C*

44. *Answer D*

45. *Answer D*

46. *Answer D*

47. *Answer C*

48. *Answer A*

49. *Answer B*

50. *Answer A*

Key Terms

Concurrent engineering An approach to project staffing that, in its most general form, calls for

implementers to be involved in the design phase.
(Sometimes confused with fast tracking.)

1. Continuously practising the application of active listening frameworks helps to develop active listening skills within Agile groups.

Daily stand-ups Concise discussions of project status with the entire team.

1. Daily stand-up meetings, daily interactions with the product team and stakeholder co-ordination are ways of encouraging collaboration and coordination.
2. Standing helps the meeting to be short.
3. To report status to the customer all of the following forms are being used—formal documentation, informal documentation and daily stand-ups.
4. Conflict resolution allows teams to preserve trust amongst members.
5. Preferred level of conflict within team is—Level 1.
6. When one person complain about another, use of three step intervention approach of conflict is the best approach.
7. Sum of knowledge all of the people on the team is called Tacit knowledge.

Halo effect is the assumption that because the person is good technically, he will be good as a project manager.

Concurrent engineering An approach to project staffing that, in its most general form, calls for implementers to be involved in the design phase. (Sometimes confused with fast tracking.)

1. Continuously practising the application of active listening frameworks helps to develop active listening skills within Agile groups.

Daily stand-ups Concise discussions of project status with the entire team.

1. Daily stand-up meetings, daily interactions with the product team and stakeholder co-ordination are ways of encouraging collaboration and coordination.
2. Standing helps the meeting to be short.
3. To report status to the customer all of the following forms are being used—formal documentation, informal documentation and daily stand-ups.
4. Conflict resolution allows teams to preserve trust amongst members.
5. Preferred level of conflict within team is—Level 1.
6. When one person complain about another, use of three step intervention approach of conflict is the best approach.
7. Sum of knowledge all of the people on the team is called Tacit knowledge.

Halo effect is the assumption that because the person is good technically, he will be good as a

project manager.

Leadership styles

1. Directing: Telling others what they need to do
2. Facilitating: Coordinating only with the input of others
3. Coaching: Instructing others clearly
4. Supporting: Providing assistance along the way to do work
5. Autocratic: Making decisions on own without input
6. Consultative: Inviting ideas from others before taking decision
7. Consensus: Problem-solving in a group with decision-making based only on group agreement

Organizational breakdown structure A diagrammatic representation of the project organization arranged to relate work packages to organizational units.

Project management team They are the members of the project team who are directly involved in day-to-day project management activities. On some smaller projects usually it includes virtually all of the project team members.

Project manager The person responsible for managing a project who is also responsible for the success and failure of the project.

Projectized organization Any organizational structure in which the project manager has full authority to assign priorities and to direct the work of individuals assigned to the project.

Responsibility assignment matrix (RAM) It actually defines who does what and you should know that the staffing management plan defines when will people get added and removed from the project.

Expectancy theory People are ready to work more energetically when the expectation is clearly set for them and usually people put in more efforts because they accept to be rewarded for their efforts. Same theory works not only for reward but also for punishment. When they know the punishment they get if they are doing the job, they will tend to work to finish the job as per expectation to avoid punishment.

War room is a technique for team building. As part of this, the project team meets in this room daily at a specified time even if there are no issues to discuss. It actually helps to create a project identity.

10

Value-based Prioritization

KNOWLEDGE AND SKILLS LEVEL

Tools and Techniques: High

- *ROI, NPV, IRR*
- *Compliance*
- *Customer Valued Prioritization*
- *Relative Prioritization and Ranking*
- *MMF*

Level 1: High

- *Prioritization*
- *Value-based Decomposition*

Level 2: Medium

- *Value Based Analysis*

Level 3: Low

- *Compliance (Organization)*
- *Regulatory Compliance*

INTRODUCTION

Value is the primary driving factor for moving from the traditional way of executing projects to the Agile model. This chapter will discuss those values in detail along with the importance of those values to the stakeholders, in particular, the importance of those values in changing enterprise environmental factors.

DEFINITION OF VALUE

‘Values are not a management tool...nor are they bits ...

11

Agile Project Risk Management

KNOWLEDGE AND SKILLS LEVEL

Tools and Techniques: High

- *Risk-adjusted Backlog*
- *Risk Burn-down Graphs*
- *Risk-based Spike*
- *Architectural Spike*

INTRODUCTION

An important responsibility of the team in Agile project is to identify all potential risks. Risks are events or actions which affect the project positively or negatively. Risk management of

Agile project is similar to traditional project management. In Agile projects, the management of risk happens more vigorously through daily stand-up meetings, Scrum planning meeting, release planning meeting, etc. Agile project planning are business value driven and risk driven. The product backlog items are not only prioritized based on value and cost, but are also prioritized based on risk.

Agile framework has inbuilt risk handling mechanism, which enables quick risk identification, response and controlling capability. The iterative nature of Agile projects helps identifying risks earlier in the project execution and also the risk process repeats for each and every iteration, thereby managing it in a better way.

Agile supports frequent iterative delivery, constant inspection through various checkpoints and quick adaptation thereby handling risks properly. Continuous integration of Agile projects prevents a long risky integration phase.

Discussion Points

1. How many risks did you find out in your previous project?
2. Name the top three risks of your previous project.
3. How did you manage these three risks?

Definition of Risk: Risks are events or consequences that have the probability of occurring during a project and that are always measured by their impacts on the project.

We always perceive risk as something bad but they not only produce a loss for the project but can also produce benefits for the project, if handled appropriately. The main purpose of risk management is to increase the probability and impact of positive events, and decrease the probability and impact of negative events on the project.

Discussion Points

How do you think that risk produces benefits to the project?

Looking at risk, one should determine:

1. What is the probability of occurrence of that risk?
2. What is the amount at stake if the risk occurs?
3. The possible timing of occurrence of the risk.
4. How often will risk occur in the project?

Discussion Points

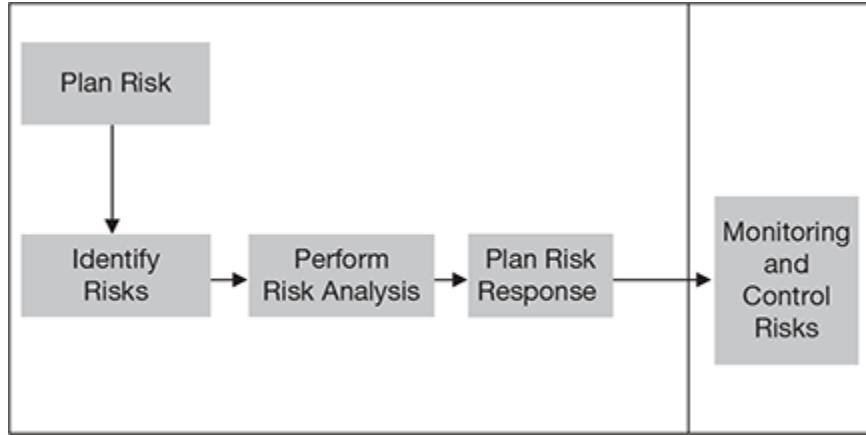
For the top three risks identified earlier in your project, discuss the probability of occurrence, the amount at stake and frequency of occurrence.

Different stakeholders of the project accept varying degrees of risk. This is called risk tolerance level. For example, a number of bugs in an iteration is acceptable up to a certain limit and this is called stakeholder tolerance level.

As per ISO 31000 ‘Risk management—Guidelines on principles and implementation of risk management’, following principles of risk management has been identified:

1. Risk management should be an integral part of organizational processes
2. Risk management should create value
3. Risk management should explicitly address uncertainty
4. Risk management should be part of decision-making process
5. Risk management should be systematic and structured
6. Risk management should be based on the best available information
 1. Risk management should take into account human factors
7. Risk management should be tailored

8. Risk management should be dynamic and iterative



9. Risk management should be responsive to change
10. Risk management should be transparent and inclusive
11. Risk management should be capable of continual improvement and enhancement

APPROACHING RISK IN AGILE PROJECTS

Careful and explicit risk management planning enhances the probability of success of other risk management processes. The primary objective of Agile risk planning is to enable the team to meet their long-term commitments. This contains the process of how are we going to identify the risks and at what level (product backlog level, sprint level and iterative cycle level), how we are going to analyze risks, what is a risk response plan, how are we going to monitor and control risks.

The primary objective of Agile risk planning is to enable the team to meet their long-term commitments. The purpose of the risk mitigation is to reduce the impact of risk.

Risk categories: It helps for the systematic identification of risks and contributes to the effectiveness and quality to identify the risk process.

Risk categories are lists of common areas or sources of risk, project management offices will have standard lists of risk categories that all projects can use to identify risks easily. Some of the risk categories are

1. **External categories:** Regulatory, markets, customers, suppliers
2. **Project management categories:** Initiation, planning, controlling, closing
3. **Organization level categories:** Resources, funding, prioritization

Mike Cottmeyer has categorized risk as follows:

1. **Business risk:** e.g., value, priority, satisfaction

2. **Technical risk:** e.g., code complexities/technical uncertainties
3. **Logistical risk:** e.g., scheduling, resourcing

IDENTIFY RISKS

This process determines which risks are likely to affect the project and documents the characteristics of each risk. Here the effect includes both the positive affect and negative effects.

The agile project can be canceled even if we encounter risks and issues during the planning phase. Identification of risk takes place continuously throughout the project.

Daily stand-up meetings identify continual risks and also helps to identify issues and risks of the Agile project. All project personnel should be encouraged to identify risks including customers.

We need to consider risk as an important feature of the product and need to identify and evaluate the project risks on a daily basis not in isolation but in collocation and collaboration.

Project team need to provide visibility about the project risks to everyone on the project and act on it to get rid of it or to minimize its impact quickly.

Look out for the transition indicators. A transition indicator is a signal when a risk is close to occurring.

An output of catastrophes brainstorming session is a risk census (risk register).

RISK ADJUSTED BACKLOG

It involves assigning a consequence (impact in days) and a likelihood score (probability of occurrence) and then calculate a risk score (called as risk exposure).

1. Risk-adjusted backlog focuses on where investment needs to be undertaken, based on risk. Plan for small releasable increments by organizing requirements into minimally marketable features/minimally viable products in order to allow for the early recognition and delivery of value.
2. The normal risk assessment database process will provide a decreasing list of priorities from the risk calculation: Potential Consequence \times Likelihood. Solicit customer and

user feedback by reviewing increments often in order to confirm and enhance business value.

3. It may be necessary to make decisions on which of these should be dealt with first within each of the risk bands.
4. Limit increment size and increase review frequency with appropriate stakeholders in order to identify and respond to risks early on and at minimal cost.

In Agile environment it is the responsibility of the whole team to identify risks on an iterative (sprint) basis. We can also create risk burn-down charts in Agile projects for each iterative cycle (sprint) to monitor the risks of the project. Risk analysis happens by identifying the risk exposure quickly.

Risks are to be marked for the items which are there in the product backlog. Assumptions or unknowns are the main sources of the project risk. Re-prioritization happens based on the risk outcome of the items. Usually, risks with high impact items are delivered early into the project.

This will document only the risks related to the features of the product backlog. We also need to identify the risks related to the execution part

of it and so we use different information gathering techniques.

An Agile team continuously re-prioritizes the product backlog (Figures 11.1 and 11.2) based on identified risks and their impact on user stories. The refined product backlog is called risk adjusted product backlog.

S. No	Risks	Probability of Occurrence	Impact in Days	Risk Exposure (Impact * Probability)
1	Risk 1	20%	15	3
2	Risk 2	50%	10	5
3	Risk 3	10%	20	2
4	Risk 4	25%	24	6

Total risk exposure (in days) = $3 + 5 + 2 + 6 = 16$ days

Figure 11.1 Risk register (Iteration 1)

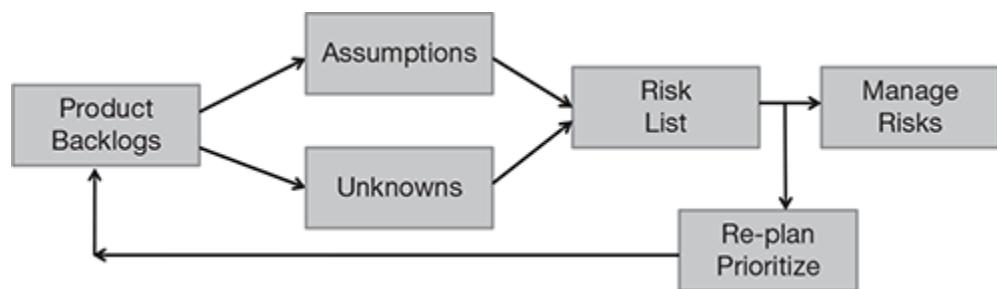


Figure 11.2 Risk adjusted product backlog

RISK BURN-DOWN GRAPHS

We can draw risk burn-down chart (graph) which contains iterative cycle number vs risk exposure days. Risks are monitored by the use of information radiators, daily stand-up meetings, and iterative cycle reviews and retrospectives. Y-axis of the risk burn-down chart contains risk exposure days. The X-axis of the risk burn-down chart contains the iterative number (Figure 11.3).

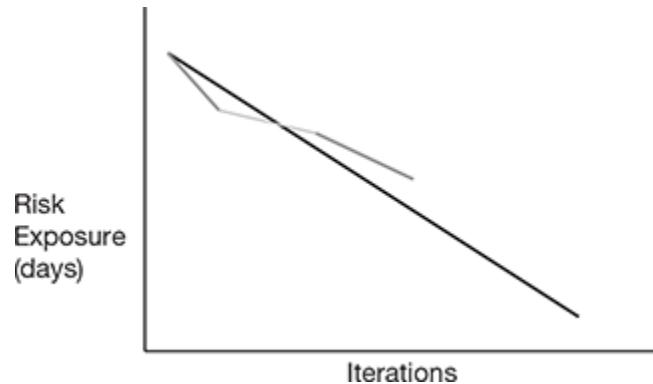


Figure 11.3 Risk burn-down chart

If the Agile project team observes that the actual risk trend line moved too much away from the ideal trend line of a risk burn-down graph, it indicates that the risk is not coming down at the appropriate rate.

If the actual risk trend line moves too much away from this ideal trend line that means the risk is not coming down at the appropriate rate and the team may want to budget some time in the next iteration to work directly on risk mitigation. Thus, the risk burn-down chart provides a convenient way of visualizing changes in risk. This should be displayed prominently in the team's work area so that the team is aware of the risk exposure of the project. This provides very useful visualization to the management as they can clearly identify whether the project risk mitigation is effective or not.

PMI-ACP WATCH

There are many ways to discover and identify risks, which includes:

- Brainstorming and information gathering techniques
- Delphi technique
- Checklist analysis

- Diagramming techniques
- Assumption analysis
- SWOT analysis
- Expert judgement

A risk trigger is a symptom of risk and it is an indirect manifestation of an actual risk event and also just because a risk trigger occurs, it does not mean that a risk will surely happen. An example of risk trigger is poor morale of the team; it is just an indication of the risk. Risk trigger is very helpful to identify the risk itself and it can be considered as the warning sign of risk.

A risk trigger is also known as a warning sign.

A signal when a risk is close to occurring is called as transition indicator.

Risk Register: Risk register in an agile project is called risk census. It is usually called risk tracker sheet and it contains the following information: list of identified risks, a list of potential responses to the risks identified, root causes of risk and updated risk categories.

PERFORM QUALITATIVE RISK ANALYSIS

Agile uses qualitative risk analysis and it uses probability impact analysis to prioritize the risks, and the risk register is informally documented using sprint risk tracker.

Agile follows only qualitative risk analysis.

Qualitative risk analysis is the process of assessing the impacts and likelihoods of identified risks and requires that the probability and consequences of the risks be evaluated using established qualitative analysis method.

For example, larger payback period results in higher risks impact in Agile project.

Qualitative risk analysis is a subjective analysis of a short list of identified risks. Subjective analysis classifies risks as high impact, medium impact and low impact, this in turn reduces the personal bias of classification.

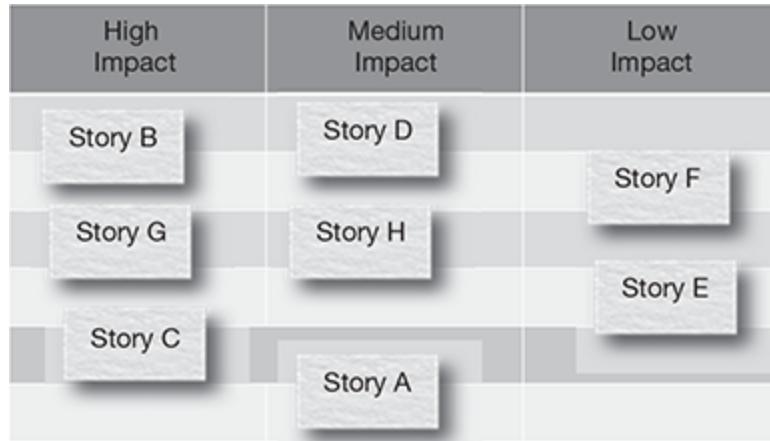


Figure 11.4 Sprint risk tracker

Organization can improve project's performance focusing immediately on high priority risks.

Performing qualitative risk analysis is usually quick and a cost effective means of establishing priorities. The team tries to attack all high impact risk items, the sprint master helps in this exercise.

Risk Probability and Impact Assessment

Risk probability—chance level of risks occurring in the project

Risk consequences—is the effect on the project if the risk occurs.

The definition of risk probability and impact is defined in the risk management plan. Risk probability assessment investigates the potential effect on project objectives such as schedule, cost and risk.

The levels of probability and impact are assessed and analysed usually in team meetings. Participants include risk-related subject matter experts along with the team members. The meeting involves measuring the probability that a risk will become a reality; impact analysis involves measuring the sensitivity of the project for risks identification.

Probability and Impact Matrix

Risk response action depends on the classification of the risk. Risks with high probability and high impact require further analysis, including quantification, quick and aggressive management. Lower risks may be

added to a watch list for monitoring purpose and later analysis.

Risk Data Quality Assessment

It evaluates the degree to which the data is useful (Figure 11.5) and accurate data is necessary for better risk analysis.

Risk data quality assessment involves the following steps:

1. Extent of understanding of a risk
2. Data available about the risk
3. Reliability of data
4. Gathering better data



Figure 11.5 Risk data quality assessment

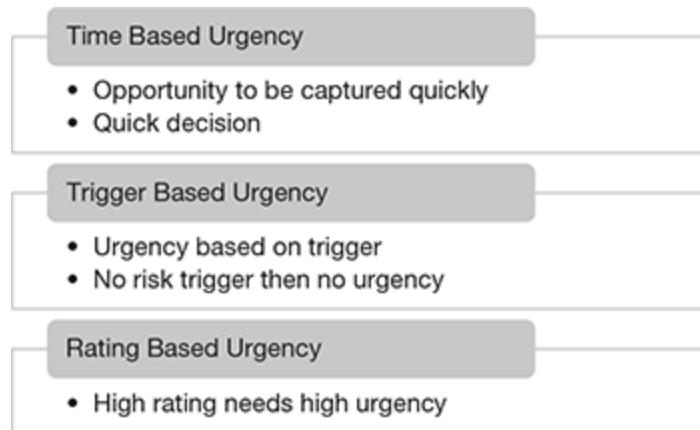


Figure 11.6 Risk urgency assessment

Risk Urgency Assessment

Indicators of risk (Figure 11.6) may include time to affect a risk response, symptoms and warning signs, and the risk rating.

- **Time based:** Positive opportunity may occur only for a short amount of time. Risk response can eliminate the risk only if it is dealt with quickly.
- **Symptoms and warning signs of risk:** These are also referred to as risk triggers and may only allow a small response window, sometimes hours before a risk is about to occur.
- **Risk rating:** High risk rating scores are typically those that have a high impact and high probability level and require our immediate attention and should be carefully monitored with an immediate risk response.

Risk Categorization

Risk categorization provides a structure that ensures a comprehensive process of systematically identifying risks to a consistent level of detail.

Risks in a project can be categorized based on the source of risks, area of the project affected or other useful category (sprint number/release number/iteration number) to determine areas of the project most exposed to uncertainty. During risk categorization, we also do the classification based on the following factors.

1. Relative rankings of risks
2. Causes of risks
3. Risks requiring response in near term

4. List of risks for further analysis
5. Watch list of low priority
6. Any trends in analysis results

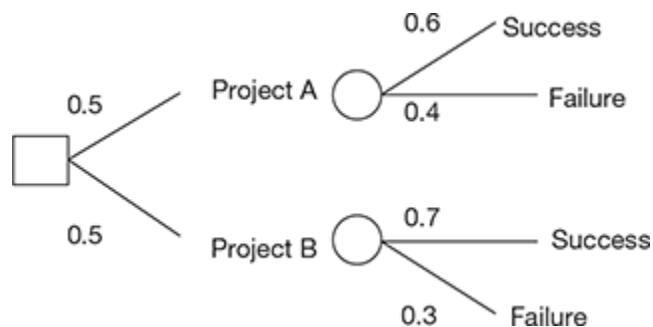
Decision Tree

It is also a tool used in risk analysis and has the chance node represented by a circle and decision node represented by a square.

PMI-ACP WATCH

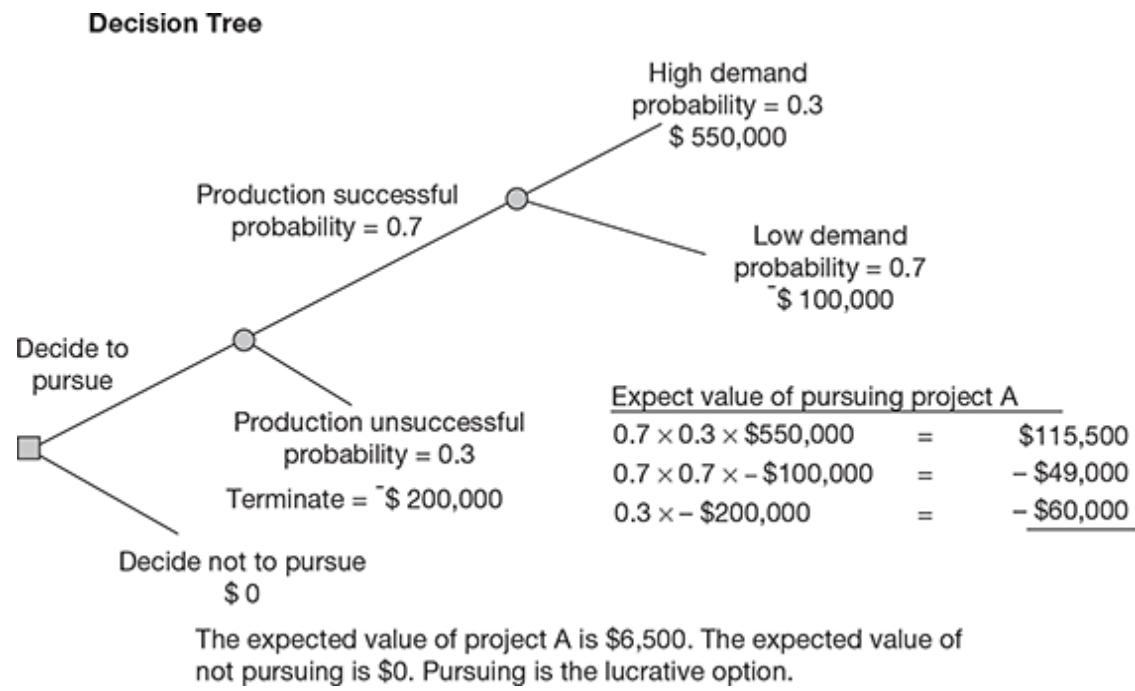
In decision tree analysis, the process starts from the right and works backward, meaning the future decisions are made first then they are rolled back to become part of the earlier decisions. The future decision can be a selection of project A, selection of project B, success of project A after selection, failure of project A after selection, success of project B after selection, failure of project B after selection.

Discrete distribution can be used to represent uncertain events such as the outcome of a test or possible scenario in a decision tree. Uniform distributions are used only if there is no obvious value that is more likely than anything between specified high and low bounds such as in the early concept stage of design.



What is the probability that Project B will be selected and will be successful?

Answer: 0.35



RISK-BASED SPIKE

In XP, the emphasis is given on obtaining real data for making a decision rather than relying on the speculation. For example, if there is any ambiguity in whether a particular database object will support the required functionality, the XP methodology will suggest that a quick program be written with that database object to check the capability and then proceed with the architecture design. The small, isolated experiment to gather more information is called spike. Normally, the term spike is used for technical investigation, but

this can very well be used for non-technical issues like management or team-related issues.

In many cases, the project risks lead us to utilize the spike solutions. The aim of these Risk-Based Spikes is to investigate the probable mitigation plans for the risks and also to provide input about the effectiveness of the mitigation plans. Spikes can also be called “Proof of Concepts”. It is also a time-boxed event. Spikes can be carried out between Sprints. Some of the spike solutions that are widely used are

1. Sensitivity analysis
2. Decision tree analysis
3. Expected monetary value analysis
4. Simulation

Sensitivity Analysis

Sensitivity analysis is used to find out risks which have the most potential impact on the project. It examines the extent to which the uncertainty of each project element affects the objective being examined when all other uncertain elements are held constantly. Tornado diagrams (Figure 11.7), also called tornado plots

or tornado charts, are useful for sensitivity analysis, which compares the relative importance of variables.

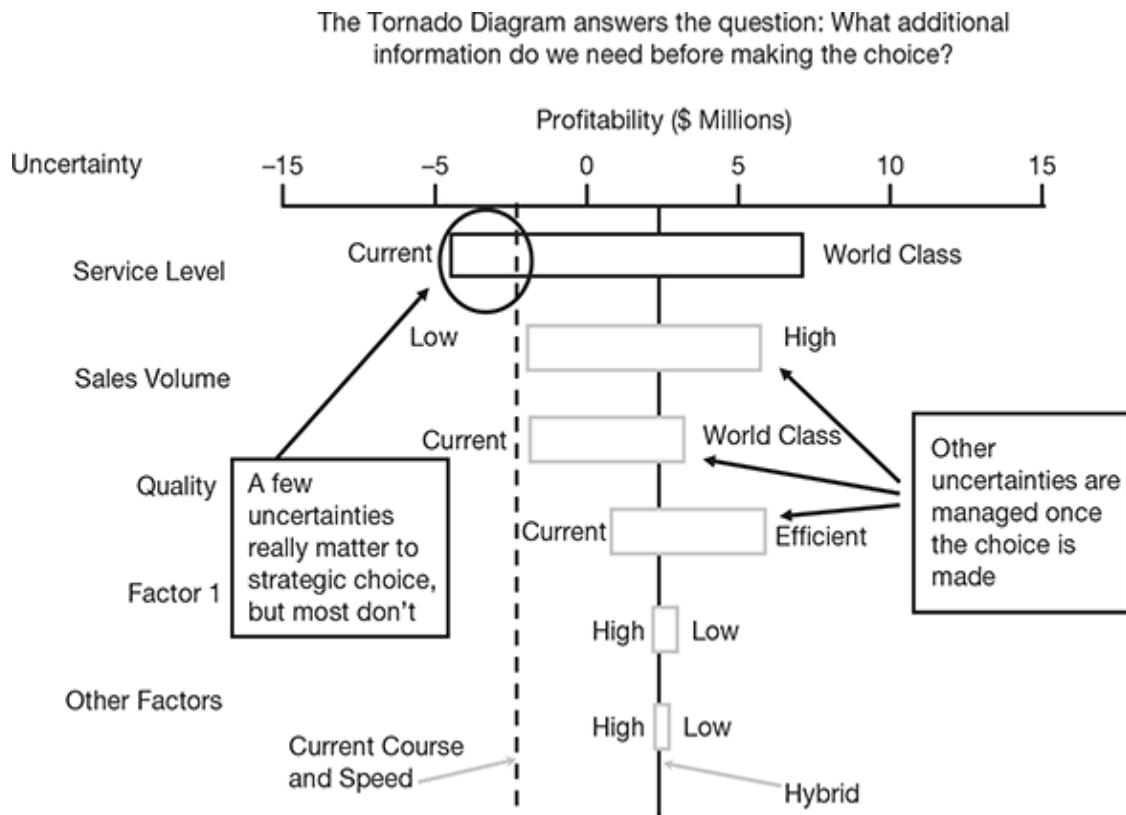


Figure 11.7 The Tornado diagram

Expected Monetary value Analysis

Expected monetary value (EMV) of a project is calculated by multiplying each possible outcome by its probability of occurrence and adding the results together.

Expected monetary value = Probability * cost.

The expected value is also a good guideline for the amount of money that might be spent to eliminate the risks in the project.

Modeling and Simulation

Simulation technique uses a model to analyze the behavior or performance of the system and the quantitative effect of risks on the system. The project is performed many times using simulation to get the statistical distribution of calculated results.

Monte Carlo analysis is a good example of simulation technique which uses the results to quantify the risk of various schedule and quality alternatives, different paths through the network diagram and different strategies of finishing the project successfully. It can be used to assess the range of possible cost options of the project using a computer-based system. The analysis results in a probability distribution.

ARCHITECTURAL SPIKE

Architectural spike is very important for continuous integration. Even though part of the product is delivered every Sprint, attention is to be given to the architecture of the product.

Spikes that are associated with an unknown area of the system such as technology or application domain are called architectural spikes. This kind of spike can also be created as a separate story.

An architectural spike may also lead to the following decisions of the project:

1. Make or buy decisions based on spike outcome
2. Creating debt stories (known open item stories)
3. To identify load impact of new story
4. To identify technologies that can be implemented

PLAN RISK RESPONSES

The process of developing options and actions to enhance opportunities and to reduce threats to project objectives refers to the plan risk response. The purpose of the risk mitigation in Agile project is to reduce the impact of the risk.

It is the process of determining actions with various possibilities to enhance opportunities and reduce threats to the project's objectives.

Risk response planning (Figure 11.8) should be based on the severity of the risk, cost effective in overcoming the challenge, timely to be successful, realistic within the project context, agreed upon by all parties involved, owned by risk owner (usually the team in Agile project) and overall it should be achievable in the above parameters.

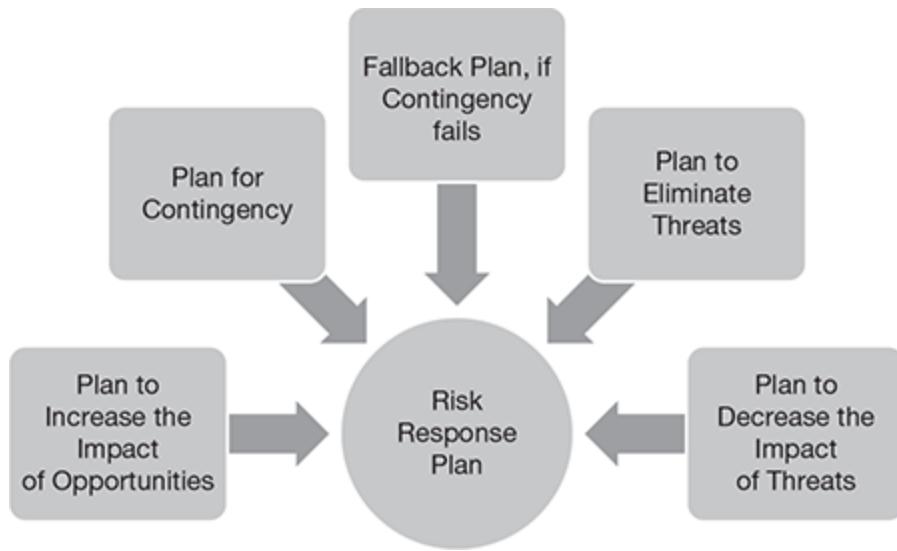


Figure 11.8 Objectives of risk response planning

Responses may include the following:

1. Action plan to eliminate the threats before they happen
2. Action plan to decrease the probability and/or impact of threats
3. Action plan to increase the probability and/or impact of opportunities
4. Plan for contingency
5. Fallback plans if contingency is not effective

Strategies for Negative Risks (Threats)

1. **Avoidance:** Eliminating a specific threat by eliminating the cause. Taking a different route by not traveling in one route is called avoidance. Agile projects because of its in-built system try to avoid a lot of risks. Shutting a project down on account of high risks is an example of avoidance.
2. **Transference:** Contracting, insurance warranties, guarantees, outsourcing the work are the examples of risk transfer, here the risk is getting transferred to another third party instead of planning the response by our self.
3. **Mitigation:** Trying to reduce the probability of occurrence, reducing the risk event value, or both and thereby reducing the cost of managing that risk. The act to reduce the impact is called mitigation.

Strategies for positive risks or opportunities

- *Exploit:* This strategy ensures that opportunity definitely happens. For example, assigning the most talented resource to your project to reduce the duration of the project.
- *Share:* Allocating part of the ownership of opportunity to a third party to ensure that the opportunity definitely happens and risk

is reduced. For example, going for a joint venture.

- *Enhance:* This strategy increases the positive impact of the opportunity. For example, adding more buffer resources to an activity to finish it early.

Discussion Points

How did you exploit the opportunities in your project? How did you enhance opportunity in your project?

Strategies for Both Threats and Opportunities

Acceptance: Accept (evade in Agile) the risk and develop a contingency plan should the risk event occur. Adopting a less complex process in the project is an instance of acceptance.

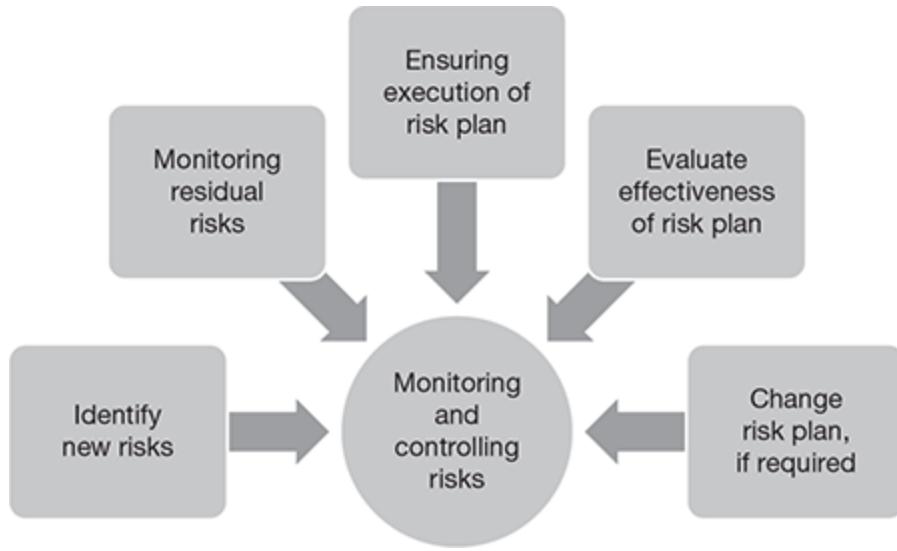


Figure 11.9 Objectives of monitoring and control risks

Contingency Reserve

The contingency reserve is created to tackle unknown risks and accepted known risks may be in the form of additional time, additional resources and additional billing from the customer.

The amount of the reserve is based on the expected impact of the risk.

Management Reserve

A separately planned quantity used to manage future situations (uncertainty) which are impossible to predict (unknown unknowns). Use of management reserve requires a change to the project's cost/schedule baseline, whereas contingency reserve need not get reflected in cost and schedule baseline.

Discussion Points

What is the difference between contingency plan and mitigation plan? Take a top priority risk of your project and discuss its contingency plan and mitigation plan.

MONITORING AND CONTROL RISKS

The process of evaluating the risk effectiveness throughout the project is called monitoring and control risks.

It is the process of keeping track of risks, monitoring residual risks and identifying new risks, ensuring the execution of risk plans and evaluating the effectiveness of the plan, changing the plan, if required.

Iteration retrospective, release retrospective are the places in Agile projects to identify and control risks. Checkpoints also help to monitor and control project risks in an Agile project.

It uses variance analysis and trend analysis technique on performance information generated during project execution to control risks related to various activities and also checks whether project assumptions are still valid.

Risks are being monitored in Agile projects by the use of high-visibility information radiators, daily stand-ups, and iteration reviews and iteration retrospectives.

CHECKPOINTS IN AGILE PROJECTS

The main advantage of Agile over other approaches is that the client is involved in the

decision-making the process not only at every stage of the iteration and daily stand-up meetings, product backlog grooming etc. but also at the checkpoints. The client accepts the incremental delivery at the end of the iterative cycle, which is called a checkpoint; the checking process is similar to verifying scope process. Because the iterative cycle length is very short (usually 2–4 weeks) and also controlled/driven by the customer closely, there is a little chance of rejection at the checkpoint. At the checkpoint the team and the business users spend time looking at what was done, what went wrong, what went right and planning the future functionality that will be built in the next iterative cycle.

Planned vs Actual Story Points

For the iterative cycle just completed, usually, the planned story points will be equal to the actual story points, the reason being we never change the iterative cycle (sprint) backlogs once it is finalized for the current iterative cycle (sprint). But sometimes rarely, due to unavoidable situation like schedule slippage, derived work, identified work etc., we may move

the story from the current iterative cycle to the next. Cycle plan (iteration plan, sprint plan) of the next iteration will be adjusted based on this.

Product Backlog Refinement

A product backlog is cumulative of all the stories which are yet to be implemented because the features which were incorporated in the previous iterative cycle (sprint) are no longer part of the current product backlog. The overall product backlog gets adjusted based on the execution of the recent iterative cycle (sprint). The following is a list of the questions to be answered during the client checkpoint phase:

1. What was planned (scope) in the cycle?
2. What was done (scope) in the cycle?
3. Is the team working as expected?
4. What went wrong in the concluded cycle?
5. What went well in the concluded cycle?

The answers to the above questions are key inputs to the planning of the next cycle and the functionalities to be added into the cycle. So we have the original work (scope) of the next iterative cycle and now we have the potential

revised work (scope) to consider as a result of the check point. In Agile the change management process is embedded in the client checkpoint. Changes are accommodated in Agile only in between iterative cycles (sprints). No changes to scope are incorporated within an ongoing iterative cycle (sprint).

We may update the product backlog as an outcome of the checkpoint. It is very normal that the items in a product backlog get re-prioritized based on the current market scenario, technical dependencies, the introduction of new features, and removing unwanted features from the backlog.

We may also take a decision to close the existing project as an outcome of the checkpoint or in some extreme cases to re-start the project from the scratch as things did not go well.

The cost of the termination in Agile projects are much lesser than the cost of termination in traditional projects as the decision is taken early in the project through checkpoints.

Termination may happen in the traditional project only after we spent all the money but at Agile we take the termination decision where there is still money/budget.

Team Work

Checkpoint also checks whether the team is working as per the expectation of all the stakeholders, conflicts, if any are being addressed here. Team commitment is the primary driving force of any Agile project and the checkpoints get this commitment at frequent intervals.

The sprint master needs to create a contusive environment for the team to work and the team members should not be afraid to raise the flag at the appropriate time and also raise concerns, if any. The checkpoint ensures that all the mistakes are being corrected before proceeding to the next iteration/iteration cycle.

Outputs of Checkpoint

1. Accepted deliverable from the current iteration/iterative cycle
2. Action items for the next iteration/iterative cycle
3. Updated product backlog
4. Reprioritized backlog
5. Determine iterative cycle length
6. Determine the functionalities to be implemented in the next iteration/iterative cycle

The deliverable (incremental product) for the current iterative cycle is being accepted in this phase and also action items for the next iteration/iterative cycle is also being determined.

The product backlog is being checked to determine whether it is still valid, updated or prioritized. If required, any new items can be added to it and can be re-prioritized before we take up the tasks for the next iterative cycle.

The client and the team work collaboratively in prioritizing the product backlog items and determine the items to be added into the next iteration cycle. The process that was used in the previous iterative cycle for prioritization of the items can also be used here. If the team (including the client) agrees then they can also

prioritize the items based on new techniques, this is being done considering the lessons learnt from the previous iterative cycles.

Exercise 11.1 Find out the type of risk management technique used in the below scenario.

1. You know that people will leave your project due to heavy work pressure and prepare a succession and rotation plan for the project.			
Avoided	Mitigated	Accepted	Transferred
2. You buy an insurance policy for the infrastructure used in the project related to a banking client as it is a big risk.			
Avoided	Mitigated	Accepted	Transferred
3. You did not want a particular person to be allocated into your project as you know the attitude of that person is not acceptable and he is technically incompetent.			
Avoided	Mitigated	Accepted	Transferred
4. You know that the billing value what you quoted for a particular client is very low and it yields very low profit margin as you are sure that you will get multiple projects from the same client in future and this project is a lead for the same.			
Avoided	Mitigated	Accepted	Transferred

Summary

In Agile projects, the management of risk happens more continuously through daily

standup meetings, Scrum planning meeting, release planning meeting, etc. Agile projects are business value-driven and risk driven. Agile projects have its own inbuilt risk handling mechanism, well aligned with quick risk identification, responding and controlling mechanism. The iterative nature of Agile projects identifies risks earlier in the project execution and also the risk process repeats for each and every iteration, thereby managing it in a better way. A secondary definition of Agile could be continuous risk management. Agile supports frequent iterative delivery, constant inspection through various checkpoints and quick adaptation thereby handling risks properly. Continuous integration of Agile projects prevents a long risky integration phase.

Answers to Exercise Questions

Exercise 11.1: Answers

1. You know that people will leave your project due to heavy work pressure and prepare a succession and rotation plan for the project.			
Avoided	Mitigated	Accepted	Transferred
2. You buy an insurance policy for the infrastructure used in the project related to a banking client as it is a big risk.			
Avoided	Mitigated	Accepted	Transferred
3. You did not want a particular person to be allocated into your project as you know the attitude of that person is not acceptable and he is technically incompetent.			
Avoided	Mitigated	Accepted	Transferred
4. You know that the billing value what you quoted for a particular client is very low and it yields very low profit margin as you are sure that you will get multiple projects from the same client in future and this project is a lead for the same.			
Avoided	Mitigated	Accepted	Transferred

Chapter 11 Questions and Answers

- Please set yourself a time clock of 1 hour to take this test
- Mark your answers using pencil in the answer sheet provided at the end of this question set
- The correct answers are provided at the end of this question set (after the answer sheet)
- Give one mark for each correct answer for evaluation purpose
- There is no negative marking for the wrong answers
- Practice this test multiple times for better results
- All the very best!

Question 1. Project risk management is defined as

1. The process of defining how to conduct risk management activities for a project.
2. The process of prioritizing risks for further analysis or action by assessing and combining their probability of occurrence and impact.
3. The process of developing options and actions to enhance opportunities and to reduce threats to protect objectives.
4. The process of conducting risk management planning, identification, analysis, response planning and monitoring and controlling a project.

Question 2. The process of numerically analysing the effect of identified risk on overall project objectives is called

1. Plan risk responses
2. Perform quantitative risk analysis
3. Perform qualitative risk analysis
4. Plan risk management

Question 3. The process of evaluating the risk process effectiveness throughout the project is called

1. Plan risk responses
2. Perform quantitative risk analysis
3. Perform qualitative risk analysis
4. Monitor and control risk

Question 4. The process of developing options and actions to enhance opportunities and to reduce threats to project objectives refers to the

1. Perform quantitative risk analysis
2. Perform qualitative risk analysis
3. Plan risk responses
4. Monitor and control risk

Question 5. Which one of the following statement is not true?

1. Project risk is related to future
2. Risk is an uncertain event or condition
3. Risk can occur without affecting any of the project objectives
4. The same risk can create one or more impacts

Question 6. The terms used to indicate varying degrees of risk that organizations are willing to accept is called

1. Risk tolerance
2. Risk limit
3. Risk level
4. Risk acceptance (risk evade)

Question 7. Which of the below helps to identify issues and risks of the Agile project on a continuous basis?

1. Distributed team
2. Daily stand-up meetings
3. Product backlog
4. Planning poker

Question 8. The primary objective of Agile risk planning is:

1. Perform quantitative risk analysis
2. Perform qualitative risk analysis
3. To enable the team to meet their short-term commitments
4. To enable the team to meet their long-term commitments

Question 9. The purpose of the risk mitigation in Agile project:

1. Avoid the risk
2. To reduce the impact of risk
3. Decrease the probability of occurrence
4. Decrease the frequency of occurrence

Question 10. Which of the following is true about risks in Agile projects?

1. Smaller payback period results in higher risks
2. Larger payback period results in smaller risks
3. No relationship between payback period and risks
4. Larger payback period results in higher risks

Question 11. Which of the following provides a structure that ensures a comprehensive process of systematically identifying risks to a consistent level of details?

1. Risk categories

2. Budgeting
3. Timing
4. List of risks

Question 12. All project personnel should be encouraged to identify risks including the customers.

1. True
2. False

Question 13. Which of the risk categorization is not according to Mike Cottmeyer's definition?

1. Business risk
2. Project management risk
3. Technical risk
4. Logistical risk

Question 14. Which of the following technique helps to reduce the bias in the data and keeps any one person from having undue influence on the outcome?

1. Brainstorming
2. Delphi technique
3. Check list analysis
4. Root cause analysis

Question 15. An Agile team is engaged in analyzing the effectiveness of mitigation plans for the risks with real data. This is called:

1. Risk burn-down chart
2. Risk impact analysis
3. Risk-based spike
4. Product backlog update

Question 16. An Agile team is busy in re-prioritizing the product backlog based on the identified risks and their impact on the user stories. What is this revised list called?

1. Risk adjusted product backlog
2. Customer-driven product backlog
3. Final product backlog
4. Sprint backlog

Question 17. At the checkpoint the team and the client spend time looking at:

1. All of these
2. what was done
3. what went wrong and what went right
4. planning the future functionality

Question 18. How does frequent checkpoints help in reducing risk exposure of an Agile project?

1. It gives project manager the chance to put pressure on the team
2. It reduces the probability of defects during the user acceptance test as client sees the in-progress product at regular interval
3. The team can find workarounds for problems early
4. Checkpoints are treated as waste of time and does not reduce risk

Question 19. Risk probability assessment investigates the potential effect on project objectives

1. True
2. False

Question 20. Which of the below approach followed in Agile projects prevents a long risky integration phase?

1. Daily stand-up meetings
2. Continuous integrations
3. Less documentation
4. Less testing

Question 21. Risk-adjusted product backlog is an important instrument in reducing the risks in Agile projects. This is because:

1. By re-prioritizing the backlog items the team can address the high-risk items earlier and thus reduce the risk in the project
2. Risk should not be a criteria for backlog prioritization
3. This helps the product owner to increase visibility in the project

4. During retrospective call this helps in making the team accountable for their work

Question 22. A spike is also called:

1. Risk mitigation
2. Data collection
3. Proof of concepts
4. Loss of time

\bQuestion 23. Which of the following techniques helps to determine the risks that have the greatest potential impact on the project?

1. Compact analysis
2. Sensitivity analysis
3. EMV analysis
4. Requirement analysis

Question 24. Which of the below is an example of sensitivity analysis?

1. Flow diagram
2. Process diagram
3. Tornado diagram
4. EMV diagram

Question 25. What does the X-axis of the risk burndown chart contain?

1. Iterative cycle

2. Risk exposure days
3. Planned story points
4. Count of risks

Question 26. EMV of a project is calculated by

1. Subtracting each possible outcome by its probability of occurrence and add it together
2. Multiplying each possible outcome by its probability of occurrence and find the maximum value
3. Multiplying each possible outcome by its probability of occurrence and adding the results together
4. Dividing each possible outcome by its probability of occurrence and adding the results together

Question 27. The agile project can be canceled even if we encounter risks and issues during the planning phase

1. True
2. False

Question 28. Shutting a project down on account of high risks is which types of risk response plan?

1. Transfer
2. Avoid
3. Accept (Evade)
4. Mitigate

Question 29. Outsourcing is a part of which risk response plan?

1. Transfer
2. Avoid
3. Accept (Evade)
4. Mitigate

Question 30. As a project manager you have decided to transfer part of risk to a third party. All of the below are examples of risk transferring except

1. Insurance
2. Performance bonds
3. Guarantees
4. Prototype

Question 31. An Agile project team observed that the actual risk trend line moved too much away from the ideal trend line of a risk burn-down graph. This means

1. That the risk is coming down very well
2. That the risk is not coming down at the appropriate rate
3. That the project is not managed properly
4. That the team is working very hard

Question 32. Taking early action to reduce the probability and/or impact of a risk occurring

during project execution is an instance of which type of risk response plan?

1. Transfer
2. Avoid
3. Accept (Evade)
4. Mitigate

Question 33. Establishing a better process at a cost of implementation is an instance of which type of risk response plan?

1. Transfer
2. Avoid
3. Accept (Evade)
4. Mitigate

Question 34. Which of the below is an output of checkpoint in an Agile project?

1. Accepted deliverable from the current iteration/iterative cycle
2. Action items for the next iteration/ iterative cycle
3. All of these
4. Updated and re-prioritized product backlog

Question 35. Maintaining a contingency reserve to handle risks is an instance of which type of risk response planning?

1. Transfer

2. Avoid
3. Accept (Evade)
4. Mitigate

Question 36. If risk is not coming down at appropriate rate then what can be the possible action from the team?

1. The team should stop the work and escalate the situation to project manager
2. The team should ignore this as a one-off case
3. The team may want to budget time in the next iteration to work directly on risk mitigation
4. Ask the customer what to do

Question 37. All of the following are examples of positive risks except.

1. Exploit
2. Share
3. Accept
4. Mitigate

Question 38. Which of the below give an opportunity to surface any probable risk as frequently as daily?

1. Daily onsite-offshore call
2. Daily burn-down chart update
3. Daily stand-up meeting
4. Status call

Question 39. Suppose the government invests considerably in PPP (private-public partnership), which type of strategy is the government is implementing?

1. Share
2. Exploit
3. Accept
4. Mitigate

Question 40. Monitor and control risks can involve all of the following except

1. Choosing alternative strategies
2. Executing a fallback plan
3. Taking correction action
4. Analysing the risk

Question 41. Which of the following is not a part of ISO 31000 ‘Risk Management—Guidelines on Principles and Implementation of Risk Management’?

1. Risk management should be an integral part of organizational processes
2. Risk management should create value
3. Risk management should explicitly address uncertainty
4. Risk management is the responsibility of the project manager

Question 42. Comparing the amount of the contingency reserve remaining to the amount of risk remaining at any time in the project is called.

1. Risk analysis
2. Reserve analysis
3. Contingency analysis
4. Project analysis

Question 43. Management reserves are used to handle which of the following type of risk?

1. Pure risks
2. Business risks
3. Financial risks
4. Unknown unknowns

Question 44. When should risk identification be performed? (Select the best among options provided)

1. During initiation phase
2. During planning phase
3. DSuring execution phase
4. Risk identification should be performed throughout the project on a regular basis

Question 45. If a project has a 60% chance of a \$200,000 profit and a 40% chance of a \$200,000

loss, the expected monetary value of the project is

1. \$40,000 profit
2. \$80,000 loss
3. \$30,000 profit
4. \$40,000 loss

Question 46. Performing Risk Probability and Impact analysis is the responsibility of

1. Project team
2. Project manager
3. Client
4. Product owner

Question 47. What is a transition indicator in Agile risk management?

1. The stage when the project phase is changed
2. A transparent process
3. A signal when a risk is close to occurring
4. Risk log

Question 48. A risk trigger is also called as

1. Scope point
2. Kill point
3. Warning sign
4. Phase end point

Question 49. From this table it can be easily found out which risks should be handled first as those have high impact and chance of occurrence. This table is called:

1. Risk register
2. Risk Probability and Impact matrix
3. Risk-adjusted backlog
4. Risk-based spike

Question 50. How often should identification of risk take place?

1. Only when the project manager is free
2. One per phase is enough as the project is big
3. Throughout the project
4. One in the planning phase

Answer Sheet for Chapter 11 Questions

Question Number	Answer	Question Number	Answer
-----------------	--------	-----------------	--------

Question 1	Question 26		
Question 2	Question 27		
Question 3	Question 28		
Question 4	Question 29		
Question 5	Question 30		
Question 6	Question 31		
Question 7	Question 32		
Question 8	Question 33		
Question 9	Question 34		
Question 10	Question 35		
Question 11	Question 36		
Question 12	Question 37		
Question 13	Question 38		
Question 14	Question 39		
Question 15	Question 40		
Question 16	Question 41		
Question 17	Question 42		
Question 18	Question 43		
Question 19	Question 44		
Question 20	Question 45		
Question 21	Question 46		

Question 22	Question 47
Question 23	Question 48
Question 24	Question 49
Question 25	Question 50

Answers for Chapter 11 Questions

Question Number	Answer	Question Number	Answer
-----------------	--------	-----------------	--------

Question 1	D	Question 26	C
Question 2	B	Question 27	A
Question 3	D	Question 28	B
Question 4	C	Question 29	A
Question 5	C	Question 30	D
Question 6	A	Question 31	B
Question 7	B	Question 32	D
Question 8	D	Question 33	B
Question 9	B	Question 34	C
Question 10	D	Question 35	C
Question 11	A	Question 36	C
Question 12	A	Question 37	D
Question 13	B	Question 38	C
Question 14	B	Question 39	A
Question 15	C	Question 40	D
Question 16	A	Question 41	D
Question 17	A	Question 42	B
Question 18	B	Question 43	D
Question 19	B	Question 44	D
Question 20	B	Question 45	A
Question 21	A	Question 46	A

Question 22	C	Question 47	C
Question 23	B	Question 48	C
Question 24	C	Question 49	B
Question 25	A	Question 50	C

Explanations for Chapter 11 Answers

1. **Answer D**

Option A is the process of plan risk management, option B is perform qualitative risk analysis, option C is plan risk responses and only option D is the definition of risk management.

2. **Answer B**

Assigning numeric value means quantifying the figure and hence option B is right.

3. **Answer D**

Risk process effectiveness will happen in monitor and control risks and also throughout the life cycle of the project.

4. **Answer C**

5. **Answer C**

‘Not true’ means false. So rephrase the question as which one of the below is false before answering the question.

When risk occurs it will affect any of the project objectives, since it affect the project objectives, it is known as risk. The same risks also can create more than one impact on the project.

6. **Answer A**

Degree of risk acceptance is called as risk tolerance.

7. **Answer B**

8. **Answer D**

9. Answer B

10. Answer D

11. Answer A

Systematically identifying a risk is possible in consistent detail with the help of risk categories.

12. Answer A

Do not exclude customer in the risk identification process. He is an important stakeholder in the identification of risks.

13. Answer B

Mike Cottmeyer has categorized risk as follows:

1. Business risk: e.g. value, priority, satisfaction
2. Technical risk: e.g. code complexities/technical uncertainties
3. Logistical risk: e.g. scheduling, resourcing

14. Answer B

Delphi technique is an expert technique where participants are anonymous. Anonymity is the plus point of Delphi technique which helps to reduce the bias on the data and also helps to avoid one person's domination and undue influence on the outcome.

15. Answer C

The aim of these risk-based spikes is to investigate the probable mitigation plans for the risks and also to provide input about the effectiveness of the mitigation plans.

16. Answer A

17. Answer A

The following is a list of the questions to be answered during the client checkpoint phase:

1. What was planned (scope) in the cycle?
2. What was done (scope) in the cycle?

3. Is the team working as expected?
4. What went wrong in the concluded cycle?
5. What went well in the concluded cycle?

18. *Answer B*

The client is able to check the product at regular intervals and thus the chances of surprise rejection reduce.

19. *Answer B*

Risk impact assessment investigates the potential effect on a project objective such as schedule, cost and quality.

20. *Answer B*

21. *Answer A*

Risk-adjusted backlog - The highest risk items are being delivered first in order to deliver value to the customer and the backlog is adjusted based on the risks.

22. *Answer C*

Spike is also called Proof of Concept.

23. *Answer B*

Sensitivity analysis helps to determine the risk that has the most potential impact on the project.

24. *Answer C*

25. *Answer A*

26. *Answer C*

$P^* C$ is the formula for EMV. C stands for possible outcome and P is the probability of occurrence of that event.

27. *Answer A*

28. *Answer B*

Shutting the project is an example of risk avoidance.

29. *Answer A*

Outsourcing and insurance are good examples of risk transfer to others.

30. *Answer D*

Outsourcing, insurance, performance bonds, guarantees are examples of risk transfer to others.

31. *Answer B*

32. *Answer D*

Action to reduce the impact is called mitigation.

33. *Answer B*

34. *Answer C*

Outputs of Checkpoint

1. Accepted deliverables from the current iteration/iterative cycle
2. Action items for the next iteration/iterative cycle
3. Updated product backlog
4. Reprioritized backlog
5. Determine iterative cycle length
6. Determine the functionalities to be implemented in the next iteration/iterative cycle

35. *Answer C*

When we keep contingency reserve means we are accepting the risk and are ready to handle it with contingency.

36. *Answer C*

37. *Answer D*

Accept is the only strategy useful for both negative and positive strategy. Mitigation is the strategy for negative risks.

38. *Answer C*

The important benefit of daily stand-up calls is that the team can surface risks as early as possible.

39. *Answer A*

It is sharing responsibility and sharing risks strategy.

40. *Answer D*

41. *Answer D*

As per ISO 31000 ‘Risk Management—Guidelines on Principles and Implementation of Risk Management’, the

following principles of risk management have been identified:

1. Risk management should be an integral part of the organizational processes
2. Risk management should create value
3. Risk management should explicitly address uncertainty
4. Risk management should be part of the decision-making process
5. Risk management should be systematic and structured
6. Risk management should be based on the best available information-risk management should take into account human factors
7. Risk management should be tailored
8. Risk management should be dynamic and iterative
9. Risk management should be responsive to change
10. Risk management should be transparent and inclusive
11. Risk management should be capable of continual improvement and enhancement

42. *Answer B*

Reserve analysis compares the amount of the contingency reserve remaining to the amount of risk remaining at any time in the project.

43. *Answer D*

44. *Answer D*

45. *Answer A*

46. *Answer A*

Project team is responsible for identifying and mitigation planning for the risks.

47. **Answer C**

48. **Answer C**

Probability and Impact matrix is created in a tabular structure, which helps in risk categorization and prioritization.

49. **Answer B**

50. **Answer C**

Key Terms

Amount at stake The extent of adverse consequences in terms of dollar (also referred to as risk impact).

Business risk The inherent chances for both profit and loss associated with business activities.

Contingency planning Alternative strategies to be used if specified risks occur.

Contingency reserve A separately planned quantity for ‘unknown unknowns’ and are intended to reduce the impact of missing cost or schedule objectives, included in the project’s cost and schedule baselines.

Deflection Transferring all or part of a risk to another party, usually through contract.

Expected monetary value Multiplication of an event's probability of occurrence and the gain or loss that will result.

Impact analysis Examining the nature of individual risks on the project and potential arrangements of interdependent risks.

Insurable risk Risk covered by an insurance policy and is also referred to as a pure risk.

Management reserve A separately planned quantity to reduce the risk of missing cost or schedule objectives and requires a change to the project's cost baseline.

Monte carlo analysis A schedule risk assessment technique to calculate a distribution of likely results using simulation.

Opportunities Positive outcomes of risk.

Project risk management Processes concerned with identifying, analysing and responding to project-related risk.

Risk identification Determining risks which are likely to affect the project.

Risk management plan Part of the overall project plan which documents the procedures that will be used to manage risk which covers who is responsible for managing various risk areas; how contingency plans will be implemented, and how reserves will be allocated.

Risk quantification Process of evaluating the probability of risk event occurrence and effect.

Risk adjusted backlog The highest risk items are being delivered first in order to deliver value to the customer and backlog is adjusted based on the risks.

Risk burn-down graph It draws risk exposure in each iteration and when it spikes out it is a big risk.

Risk response development Steps for opportunities and threats.

Threats Negative outcomes of risk.

Total certainty All information is known without confusion.

Total uncertainty No information is available and there is total confusion.

Workaround A response to a negative risk event and is not planned in advance of the occurrence of the risk event.

Transference Shift the consequence of the threat to a third party.

Mitigation Action plan to reduce the probability/impact of a risk to an acceptable threshold.

Enhance This strategy modifies the size of an opportunity by increasing probability and/or positive impacts, and by identifying and

maximizing key drivers of these positive impact risks.

12

PMI Code of Ethics and Professional Conduct

KNOWLEDGE AND SKILLS LEVEL

Level 1: High

- *Structure of the Code*
- *Values that Support this Code*
- *PMI Code of Conduct*
- *Business Ethics*
- *Developing Cultural Competencies*

INTRODUCTION

As practitioners of project management, we are committed to doing what is right and honorable.

We set high standards for ourselves and we aspire to meet these standards in all aspects of our lives—at work, at home and in service to our profession.

This code of ethics and professional conduct describes the expectations that we have of ourselves and our fellow practitioners in the global project management community. It articulates the ideals to which we aspire as well as the behaviors that are mandatory in our professional ...

Model Question Paper–1

- This Practice test is based on the actual syllabus for Agile certification examination
- Please set yourself a time of 3 hours
- Mark your answer on the template sheet provided at the end
- Strictly stop the test when the clock hits 3 hours
- The correct answers are provided at the end of this question set
- Give one mark for each correct answer
- There is no negative marking for wrong answers
- Practice this test multiple times for better results
- All the very Best!

Questions

Question 1. In traditional methods, the project success is measured in terms of meeting time, budget and specifications. So, as a project manager following the traditional methods, a project meeting these criteria may be considered a success, but may still be perceived as failure because:

1. The users of the system are not comfortable with the product
2. The end product does not satisfy customer needs

3. The business process does not get much value add from the project
4. Project management success and business success are two different things and should not be mixed

Question 2. While executing agile projects there are lot of constraints which limits the options of applying agile practices effectively. Which of the following constraints won't stop you in applying agile practices effectively?

1. Key team members working on multiple projects simultaneously
2. Distributed teams and junior team members
3. Commitments of Management and customer to Agile
4. Customer is not available to the project team

Question 3. While writing a program finding today's date, finding next working day, finding next day, finding day of the week, finding month of the date and all closely related concepts are written together. This design concept is called as:

1. Cohesive
2. Decoupling
3. DRY
4. Coding as design

Question 4. The characteristics of a good pilot project.

1. A project you can complete within 2 weeks

2. A project you can complete in 2 to 8 weeks
3. A project you can complete in 10 weeks
4. A project you can complete in 3 months

Question 5. During the retrospective meeting of a Scrum team, few members highlighted the issue they faced with the Product Owner they have. Which of the following is not a Valid problem with the Product owner?

1. He was not ensuring that the team has a common vision for the project
2. He was not available during the planning poker session to clarify questions
3. He was not able to make decisions that provide a better ROI from the project
4. He was not providing enough funds for the project

Question 6. Commitment to delivery and prioritization in Agile projects are usually part of:

1. Daily stand-up meetings
2. Frequent demos
3. Iterative planning
4. Retrospective meeting

Question 7. This concept gives the whole team more opportunities to notice what is working well and what is not

1. Root cause analysis

2. Informative work space
3. Pair programming
4. Retrospective

Question 8. Thumbing vote is a quick way to check for consensus. Holding thumb downside indicates:

1. I agree
2. I disagree
3. I disagree and want to explain why
4. I don't want to take part in vote

Question 9. In Agile projects the requirement are captured in the form of User Stories. Which of the following is correct about user stories?

1. The contractual agreement between customers and developers is written in form of User Stories
2. The functional specifications or features which should be part of the product for providing value to the customer are captured as User Stories
3. User stories document what the users want to see in the system in detail
4. They are promises to converse rather than detailed specifications, have value to users, are independent, are testable and are appropriately sized

Question 10. There are less important stories that team can set aside by deleting half written codes, if they need more time and it is called as

1. Silver stories

2. Gold stories
3. Zero stories
4. Set aside stories

Question 11. Last year 2 Organizations of the same nature started implementing Agile. But now, after a year, the first organization has reaped lot of benefit from Agile whereas the second one failed miserably. All of the following can be possible explanation of this situation, EXCEPT:

1. Customers and management may not have the buy-in. Agile is initiated at the developer level.
2. The organization understood the concept of Agile and rolled out to all stakeholders in planned manner
3. The management wanted to drive Agile, but the teams are not ready for the mind set change
4. There was an expectation mismatch as Agile was initiated to fix the complex issues quickly

Question 12. Which agile concepts enables programmers to improve code quality without changing its behavior

1. Customer Test
2. TDD
3. Refactoring
4. Spike Solutions

Question 13. How many operations should complete in a given period of time in a story is

called as:

1. Customer Test
2. Responsiveness
3. Latency
4. Throughput

Question 14. All of the following are the activities done in iteration 0, except

1. Establishing development environments
2. Finalizing contracts with third parties
3. Assembling the project team
4. Coding

Question 15. An essential user case is closer to

1. User stories
2. Future card
3. Functional specification
4. Functional point

Question 16. If you don't have a diverse team, your Agile process can lead to

1. Team think
2. Agile think
3. Diverse think
4. Group think

Question 17. There are three legs that hold up every implementation of empirical process control, they are:

1. Visibility, inspection and adaptation
2. Visibility, value and quality
3. Value, quality and risk
4. Visibility, quality and risk

Question 18. While you are calculating the ideal days, following are the assumptions except:

1. The story being estimated is the only thing you will work on
2. Everything you need will be on hand when you start
3. There will be no interruptions
4. Bug fixing is also part of this

Question 19. Pejorative meaning of excess time which is arbitrarily added to an estimate is called:

1. Padding
2. Buffer
3. Extra time
4. Over time

Question 20. Tasks such as establishing development environments, finalizing contracts with third parties and assembling the project are usually part of

1. Iteration 0
2. Iteration 1
3. Scrum planning
4. Release planning

Question 21. Agile coaching interventions have strongest effect in

1. In the beginning of the project only
2. At the end of the project only
3. In the middle of the project
4. Both in the beginning and end of the project

Question 22. Following are the principles of foundation for successfully moving to Agile, except:

1. Embrace change
2. Use human-centric method
3. Achieve technical excellence
4. Freeze the requirement

Question 23. A secondary definition of Agile could be

1. Continuous risk management
2. Continuous cost management
3. Continuous schedule management
4. Continuous quality management

Question 24. QWAN in Agile project stands for

1. Quality with a name
2. Quality without a name
3. Query with a non-functional item
4. Quality with a non-functional item

Question 25. Following are the weakness of XP, except

1. Scalability
2. Dependency on testing
3. Need for team maturity
4. Quality via frequent testing

Question 26. Agile practice which calls for all stakeholders to be involved with the planning process is called as:

1. Collaborative planning
2. Progressive planning
3. Team planning
4. Agile planning

Question 27. Which curve illustrates how change initiatives cause organizational go through period of resistance

1. The vergina satir change curve
2. Delphi Change Curve
3. Burn-down chart curve
4. Cumulative chart curve

Question 28. All of the following ensures that you address the customer's needs in an Agile project, except

1. Clearly define the customer(s)
2. Develop a relationship with the customer
3. Be an advocate for the customer at all times

4. Strictly follow the plan laid in the beginning

Question 29. In Agile methods, who should create the user stories?

1. The customer because the business language used in the user stories help in prioritizing those for inclusion into iterations
2. The developers as the technical details and the codes and test cases needs to be included in it
3. The entire team, which includes the Product Owner and the Development team, because they understand the project needs best
4. The product manager who sponsors the project because the acceptance test criteria comes from this person

Question 30. All of the following are examples of medium priority projects, except:

1. Adding the ability to book hotels on an existing travel site
2. Delivering a maintenance release onto an existing platform
3. Adding customizable stock quotes to a portal page
4. A project to ensure regulatory compliance

Question 31. An average feasibility analysis begins with

1. 1 or 2 hours meeting
2. 4 hours meeting
3. 1 day meeting
4. 1 or 2 days meeting

Question 32. A reason for team members cynicism relate to the following, except

1. Misunderstanding of Agile principles
2. A belief that the current process works fine
3. Attitude problem
4. Lack of detail in the Agile process

Question 33. All of the following are benefits of feature card, except:

1. Customer focus
2. Identifying risks early
3. Common understanding
4. Identifying quality early

Question 34. The first objective of kick-off meeting in Agile project is

1. To bring stakeholders and sponsors up to speed
2. To bring support groups up to speed
3. To understand the scope of the project
4. To understand the timeline of the project

Question 35. The second objective of kick-off meeting in Agile project is

1. To bring stakeholders and sponsors upto speed
2. To bring support groups up to speed
3. To understand the scope of the project
4. To understand the timeline of the project

Question 36. By measuring _____ you can find out whether people are intimidated by their managers.

1. Manager buy-in
2. Developer buy-in
3. Power distance
4. Management style

Question 37. When we create feature cards we record acceptance test too.

1. True
2. False

Question 38. Choose the right one

1. Effective project leaders focus on people, product and process —in that order
2. Effective project leaders focus on product, people and process —in that order
3. Effective project leaders focus on product, process and people —in that order
4. There is no order and all are of same focus

Question 39. All of the following are the tools related with agile project support, except:

1. Time tracking
2. Status tracking
3. Wiki tools
4. Defect tracking

Question 40. Everybody pays attention to the process and practices of development and this is called as

1. Velocity

2. Mindfulness
3. TOC
4. Iterations

Question 41. A Scrum master who takes teams beyond getting Agile practices up and running into their deliberate and joyful pursuit of high performance is a(an):

1. Agile coach
2. Product owner
3. Project manager
4. Project sponsor

Question 42. A higher than expected velocity means the total number of iteration will

1. Increase
2. Reduce
3. Constant
4. Increase and reduce

Question 43. Who dives into the Scrum depths and brings tools and techniques from many disciplines to help the team:

1. Agile coach
2. Product owner
3. Project manager
4. Scrum master

Question 44. Someone who can help the management at all levels of the organization to

understand the benefits of working Agile

1. Agile coach
2. Product owner
3. Project manager
4. Project team

Question 45. What is the effect of having large visible burn-down chart on a wall?

1. It distracts the onlookers and is a risk
2. It clearly communicates progress to the team and other stakeholders
3. It is dangerous for the team as the management gets full visibility into the project
4. It is useless, as it does not provide any information to the team

Question 46. All of the following are the forms taken by the Agile coach, except:

1. Facilitator
2. Mentor
3. Problem solver
4. Delivery manager

Question 47. Choose the common characteristics prevailing across coach

1. They risk being wrong
2. They don't believe disequilibrium is essential
3. They don't cultivate curiosity
4. They care about products more than people

Question 48. Which style comes when the team has fully internalized the practices, values and principles of Agile and radiates good health:

1. Teaching
2. Coaching
3. Advising
4. Directing

Question 49. If the team have the basic practices working well and producing new insights that let them improve each sprint, they are in which stage?

1. Shu
2. Ha
3. Ri
4. All three stages

Question 50. Horizontal software is developed for

1. Many organizations
2. Many industries
3. Across wide range of industries
4. Across wide range of organizations

Question 51. The most important thing you may learn from the coding standard is

1. How to disagree constructively
2. How to agree constructively
3. Quick coding

4. Fast tracking

Question 52. Coaching product owners falls into three broad categories, except

1. Running their own business
2. Being a product owner for the team
3. Getting good at the product owner's role
4. Being a Scrum master for the team

Question 53. In an Agile project, the coach at the start focuses the team on task orientation rather than on relationship orientation.

1. True
2. False

Question 54. Following are the activities which helps in team building in Agile projects, except

1. Journey lines
2. Market of skills
3. Constellation
4. Individual appraisals

Question 55. Language is open and fact based is at what level of conflict:

1. Level 1
2. Level 5
3. Level 3
4. Level 4

Question 56. The contract which allows for built in checkpoints to provide the customer visibility and go/no-go decision along the way is referred as

1. Visibility contract
2. Staged contract
3. Decision base contract
4. Checkpoint contract

Question 57. The term FIT in acceptance testing stands for

1. Famous integration testing
2. Framework for integration testing
3. Framework for iteration testing
4. Framework for intelligent testing

Question 58. The length of the iteration is determined by

1. Control needs of the customer
2. Limitation of the team
3. Project manager
4. Team lead

Question 59. Which of the following is not an assumption for Agile EVM

1. Measure progress at the release level, not the iteration or product level
2. Measure progress at the iteration level if actual iteration velocity and actual iteration cost is known

3. Functionality is done at the end of each iteration
4. Functionality is does at the end of each release

Question 60. Quality is not

1. Planned
2. Designed
3. Built in
4. Inspected in

Question 61. The iteration review meeting is usually a low ceremony with minimal preparation.

1. True
2. False

Question 62. Enumeration of complexity in software development is limited to the three most significant dimensions, they are

1. Requirements, designing and testing
2. People, process and procedure
3. Quality, risks and procurement
4. Requirements, technology and people

Question 63. Agile project deliver reliable results by

1. Engaging customers in frequent interactions and shared ownership.
2. Recognizing that individuals are the ultimate source of value and creating an environment where they can make a difference.

3. Making continuous flow of value our focus.
4. Group accountability for results and shared responsibility for team effectiveness.

Question 64. As per the report from the standish group, what percentage of the features in a product are never used?

1. 10%
2. 45%
3. 25%
4. 60%

Question 65. An Agile manager never does the following:

1. Helps the development team to track the true status
2. Acts as a buffer for outside interruptions and limits team distractions
3. Helps the team to break their work into small chunks that can be delivered quickly
4. Documents the requirements and tests a feature

Question 66. Agile projects have the ability to always be on time and within budget if _____ is flexible

1. Team
2. Scope
3. Risk
4. Product

Question 67. Which of the following ceremony in Agile project by default is open to every stakeholders of the project?

1. Demo
2. Retrospective
3. Planning
4. Review

Question 68. Stories are _____ centric, describing the results in terms of _____.

1. Programmer, business results
2. Project manager, business results
3. Customer, business results
4. Developer, \$ value

Question 69. Risk evade is similar to

1. Risk mitigation
2. Risk acceptance
3. Risk avoid
4. Risk contain

Question 70. Who is responsible for figuring out how to turn product backlog into an increment of functionality within an iteration?

1. Project manager
2. The team
3. The Scrum master
4. Product owner

Question 71. There is a metric imposed by concentrating knowledge in just a few people's heads; it is called as

1. Bus number
2. Knowledge number
3. Knowledge hub
4. Truck number

Question 72. In Agile projects, who can change the sprint backlog?

1. Project manager
2. The team
3. The Scrum master
4. Product owner

Question 73. Which of the following is not a valid reason for updating the product backlog?

1. Findings from the previous sprints
2. Change in market condition which added/deleted some features (stories)
3. Technical constraints or findings
4. Features that development team want to add for helping the users

Question 74. In Agile projects the plan is progressively elaborated - the strategic planning focuses on the long term goals and the tactical planning focuses on the short term tasks for achieving the goal gradually. Which one of the

following planning type combinations implement this strategic and tactical planning approach in a project?

1. Iteration planning, agile planning
2. Release planning, iteration planning
3. Detailed planning, test first planning
4. Release planning, portfolio planning

Question 75. Fractional assignment is common in what kind of structure?

1. Projectized structure
2. Functional structure
3. Matrix managed structure
4. Flat structure

Question 76. XP emphasizes on

1. Face-to-face collaboration
2. E-mail collaboration
3. Document collaboration
4. Wiki collaboration

Question 77. Customer tests help ensures

1. Programmers' intent matches customers' expectations
2. Testers' intent matches programmers' expectations
3. Testers' intent matches customers' expectations
4. Customers' intent matches programmers' expectations

Question 78. An organization which expects employees to work extensive overtime week

after week, is called as

1. Live march organization
2. Death march organization
3. Live week organization
4. Death week organization

Question 79. Who is the worst possible choice to act as a proxy user?

1. Sales persons
2. Domain experts
3. Development managers
4. Customers

Question 80. In XP project, onsite customers' most important activity is

1. Release planning
2. Sprint planning
3. Customer test
4. Customer integration

Question 81. Following are the characteristics of better user stories, except:

1. Writing closed stories
2. Writing open stories
3. Written in active voice
4. Slice the cake principle

Question 82. Which of the following is essential for the team to thrive?

1. Trust
2. Sitting together
3. Ubiquitous language
4. Coding standards

Question 83. Which of the following keeps the team efforts aligned with stakeholder goals?

1. Trust
2. Sitting together
3. Iteration demos
4. Coding standards

Question 84. In Agile projects, the vision might be vague at first, perhaps stated in market terms rather than system terms.

1. True
2. False

Question 85. In XP project who is the backup when the product manager travels

1. Scrum master
2. Domain expert
3. Team lead
4. Onsite customer

Question 86. Which of the following ensures that the completed work is ready to release?

1. Done done
2. Version control

- 3. Continuous integration
- 4. Collective code ownership

Question 87. If a Gantt report includes only _____, and not _____, it is merely a graphical representation of the product backlog.

- 1. Requirements, tasks
- 2. Tasks, requirements
- 3. Tasks, duration
- 4. Duration, tasks

Question 88. In XP project, we need to keep the build under _____ minutes

- 1. 5 minutes
- 2. 1 minutes
- 3. 10 minutes
- 4. 20 minutes

Question 89. What is second adopter system?

- 1. First team is often successful; second wave XP projects struggle
- 2. First team often struggles; second wave XP project passes
- 3. Every second projects succeeds
- 4. Every second projects struggles to execute XP

Question 90. A properly conducted _____ can be a very rapid way to write a great number of stories

- 1. Questionnaire
- 2. Interview

3. Observation
4. Story-writing workshop

Question 91. Little pockets of code which only few people know is called as

1. Knowledge silos
2. Pocket silos
3. People silos
4. Code silos

Question 92. Inferring high level design from a brief review of the code is:

1. Green belt design
2. White belt design
3. Black belt design
4. Red belt design

Question 93. Which of the following will reduce the impact of the risk?

1. Mitigation activities
2. Contingency activities
3. Transition indicators
4. Risk exposure

Question 94. Collectively, the developers have a sequence in which they would like to implement the stories, as well the customer. When there is a disagreement to the sequence, who wins?

1. The developers

2. Third party
3. Customers
4. Nobody wins

Question 95. When a story is too large it is sometimes referred to as a(n)

1. Epic
2. Features
3. Big story
4. User story

Question 96. A Scrum team is half-way through their third 2-week iteration. Due to changes in the marketing strategy, the clients have started requesting many changes throughout the day. Because of this the team is losing focus of their current deliverable and the productivity in the current sprint is going down. How you will manage the situation so that the team can be shielded from all these disturbances but you still can produce valuable output for the customer.

1. Push all the requests that come in for discussion until the next iteration planning
2. Cancel the current iteration half-way and work on a new iteration incorporating the requested changes
3. Accept those changes that are high priority and work it in along with your current iteration, volatile requirement is normal in Agile
4. Change the iteration length from 2 weeks to a 2-day iteration for incorporating changes

Question 97. Past studies indicates that less than _____ percentage of time is spent on coding

- 1. 90%
- 2. 80%
- 3. 50%
- 4. 20%

Question 98. Which of the following is not true about user stories?

- 1. User stories emphasize written rather than verbal communication.
- 2. User stories are comprehensible by both you and the developers.
- 3. User stories are the right size for planning.
- 4. User stories work for iterative development.

Question 99. The accumulation of problems caused by poor quality practices in Agile is called as

- 1. Problem debt
- 2. Technical debt
- 3. Quality debt
- 4. Accumulation debt

Question 100. In XP project, brief experiment to learn about an area of the application is called as

- 1. Spike
- 2. Technical debt

3. Technical stories
4. Experiment

Question 101. Agile business objective which is to support business growth and profitability

1. Continuous innovation
2. Product adaptability
3. People and process adaptability
4. Reliable results

Question 102. The danger in using a _____ as a user proxy is that it does not lead to a comprehensive view of the product to be built.

1. Sales persons
2. Domain experts
3. Development managers
4. Customers

Question 103. During a 4-week iteration, an Agile team delivered 6 stories estimated at 5 story point each and 4 stories estimated at 3 story point each. What is the velocity of this team?

1. 42
2. 10.5
3. 30
4. 12

Question 104. Scope creep does not exist in Agile projects because

1. There is no change in scope
2. Scope is expected to change
3. Tracking of the scope is not happening
4. Scope is not planned in Agile projects

Question 105. Agile development focuses on all of the following except

1. Speed
2. Mobility
3. Quality
4. Project plan

Question 106. An Agile project manager is planning the seating arrangement for the team and he is making sure that all the team members are visible to each other. Why is he planning this way?

1. The project manager wants authority
2. He has space crunch
3. The team will be able to chat easily
4. This helps in easy communication flow

Question 107. There is no upfront design phase, Agile processes are characterized by frequent short bursts of design.

1. True
2. False

Question 108. User stories originated as part of

1. Scrum
2. DSDM
3. Extreme programming
4. Kanban

Question 109. Agile project unleash creativity and innovation by

1. Engaging customers in frequent interactions and shared ownership.
2. Recognizing that individuals are the ultimate source of value and creating an environment where they can make a difference.
3. Making continuous flow of value our focus.
4. Group accountability for results and shared responsibility for team effectiveness.

Question 110. You are starting an Agile project and the team has come to you for a clarification regarding the design process to be followed. As an Agile practitioner which of the following statement regarding Agile design process you will recommend for the team:

1. Iterative, emergent design and frequent feedback
2. Robust design at the beginning and little change afterwards
3. Direct coding as designing will waste much time
4. Ask the customer for design and start from coding

Question 111. Idea that the right set of simple rules, applied within a group of highly interactive individuals, generates complex

behaviours such as innovation and creativity is called:

1. Simple intelligence
2. Idea intelligence
3. Swarm intelligence
4. Interactive intelligence

Question 112. Daily stand-ups, daily interactions with the team and stakeholder coordination are ways of:

1. Getting work done by team
2. Enhancing collaboration and coordination
3. Project tracking
4. Team building

Question 113. A persona is an imaginary representation of a user role. We attach as much details with the persona as possible for the purpose:

1. Understand the users of the product better
2. Make the users feel that their needs are captured
3. Write the acceptance test cases better
4. Help the project team to build rapport with the customer

Question 114. Shared space has two requirements

1. Value and visualization
2. Visualization and commonality
3. Quality and value

4. Value and commonality

Question 115. An Agile project team wants to display the current impediments to all the stakeholders in an easy to understand way. What is the best place to put this information?

1. On an information radiator
2. In an e-mail
3. Tell the project manager
4. Call a team meeting

Question 116. Acronym DEEP is referred in preparing for spring planning stage stands for:

1. Detailed appropriately, estimated, emergent and prioritized
2. Detailed, estimated, emergent and planned
3. Disagreement, expectation, evolution and prioritized
4. Developed, elaborated, explorated and productized

Question 117. Following are the examples of management reports except:

1. Status e-mail
2. Productivity report
3. Throughput report
4. Time usage

Question 118. All of the following are Mayer's five things for Agile projects, except:

1. Empiricism
2. Rhythm

3. Collaboration
4. Daily planning

Question 119. Adaptive project management (APM) is

1. Execution-biased model
2. Planning-biased model
3. Control-biased model
4. Monitoring-biased model

Question 120. Which of the following is not true about collaboration?

1. Decision-making is the heart and soul of collaboration.
2. Collaboration means working together to jointly build a feature, create a design or write a product's documentation.
3. How a team makes those decisions determines whether the team is a truly collaborative one.
4. In collaboration, some teams are driven to quick decisions by a senior technical individual, while others are driven by those with the loudest voice.

Answer Sheet for Model Question Paper 1

Question Number	Answer	Question Number	Answer	Question Number	Answer
Question 1		Question 41		Question 81	
Question 2		Question 42		Question 82	
Question 3		Question 43		Question 83	
Question 4		Question 44		Question 84	
Question 5		Question 45		Question 85	
Question 6		Question 46		Question 86	
Question 7		Question 47		Question 87	
Question 8		Question 48		Question 88	

Question 9		Question 49		Question 89	
Question 10		Question 50		Question 90	
Question 11		Question 51		Question 91	
Question 12		Question 52		Question 92	
Question 13		Question 53		Question 93	
Question 14		Question 54		Question 94	
Question 15		Question 55		Question 95	
Question 16		Question 56		Question 96	
Question 17		Question 57		Question 97	
Question 18		Question 58		Question 98	
Question 19		Question 59		Question 99	
Question 20		Question 60		Question 100	
Question 21		Question 61		Question 101	
Question 22		Question 62		Question 102	
Question 23		Question 63		Question 103	
Question 24		Question 64		Question 104	
Question 25		Question 65		Question 105	
Question 26		Question 66		Question 106	
Question 27		Question 67		Question 107	
Question 28		Question 68		Question 108	
Question 29		Question 69		Question 109	
Question 30		Question 70		Question 110	
Question 31		Question 71		Question 111	
Question 32		Question 72		Question 112	
Question 33		Question 73		Question 113	
Question 34		Question 74		Question 114	
Question 35		Question 75		Question 115	
Question 36		Question 76		Question 116	
Question 37		Question 77		Question 117	
Question 38		Question 78		Question 118	
Question 39		Question 79		Question 119	
Question 40		Question 80		Question 120	

Answers for Model Question Paper 1

Question Number	Answer	Question Number	Answer	Question Number	Answer
Question 1	B	Question 41	A	Question 81	B
Question 2	R	Question 42	R	Question 82	A

QUESTION 12	U	QUESTION 72	U	QUESTION 62	U
Question 3	A	Question 43	D	Question 83	C
Question 4	B	Question 44	A	Question 84	A
Question 5	D	Question 45	B	Question 85	B
Question 6	C	Question 46	D	Question 86	A
Question 7	B	Question 47	A	Question 87	A
Question 8	C	Question 48	B	Question 88	C
Question 9	B	Question 49	B	Question 89	A
Question 10	A	Question 50	C	Question 90	D
Question 11	B	Question 51	A	Question 91	A
Question 12	C	Question 52	D	Question 92	C
Question 13	D	Question 53	A	Question 93	A
Question 14	D	Question 54	D	Question 94	C
Question 15	B	Question 55	A	Question 95	A
Question 16	D	Question 56	B	Question 96	A
Question 17	A	Question 57	B	Question 97	D
Question 18	D	Question 58	A	Question 98	A
Question 19	A	Question 59	D	Question 99	B
Question 20	A	Question 60	D	Question 100	A
Question 21	D	Question 61	A	Question 101	D
Question 22	D	Question 62	D	Question 102	A
Question 23	A	Question 63	A	Question 103	A
Question 24	B	Question 64	B	Question 104	B
Question 25	D	Question 65	D	Question 105	D
Question 26	A	Question 66	B	Question 106	D
Question 27	A	Question 67	A	Question 107	A
Question 28	D	Question 68	C	Question 108	C
Question 29	C	Question 69	B	Question 109	B
Question 30	D	Question 70	B	Question 110	A
Question 31	A	Question 71	D	Question 111	C
Question 32	C	Question 72	B	Question 112	B
Question 33	D	Question 73	D	Question 113	A
Question 34	A	Question 74	B	Question 114	B
Question 35	B	Question 75	C	Question 115	A
Question 36	C	Question 76	A	Question 116	A
Question 37	A	Question 77	A	Question 117	A
Question 38	A	Question 78	B	Question 118	D
Question 39	D	Question 79	C	Question 119	A
Question 40	B	Question 80	A	Question 120	D

Model Question Paper–2

- This Practice test is based on the actual syllabus for Agile certification examination
- Please set yourself a time of 3 hours
- Mark your answer on the template sheet provided at the end
- Strictly stop the test when the clock hits 3 hours
- The correct answers are provided at the end of this question set
- Give one mark for each correct answer
- There is no negative marking for wrong answers
- Practice this test multiple times for better results
- All the very Best!

Questions

Question 1. DRY principle was introduced by Andy Hunt and Dave Thomas which is widely followed in Agile development process to avoid code duplication. DRY is the acronym for

1. Don't repeat yourself
2. Don't review yourself
3. Do review yourself
4. Do retrospect yourself ...

Model Question Paper–3

- This Practice test is based on the actual syllabus for Agile certification examination
- Please set yourself a time of 3 hours
- Mark your answer on the template sheet provided at the end
- Strictly stop the test when the clock hits 3 hours
- The correct answers are provided at the end of this question set
- Give one mark for each correct answer
- There is no negative marking for wrong answers
- Practice this test multiple times for better results
- All the very Best!

Questions

Question 1. This agile concept helps the team to understand what to build.

1. Trust
2. Sitting together
3. Real customer involvement
4. Ubiquitous language

Question 2. All of the following are the examples of progress reports except:

1. Burn-up chart

2. Status e-mail
3. Weekly demo
4. Time usage

Question 3. This helps to keep the team's efforts aligned with stakeholder's goals

1. Trust
2. Coding standards
3. Iteration demos
4. Reporting

Question 4. A brief statement designed to impart the intent of the project within 2 minutes:

1. Intent statement
2. Elevator statement
3. Scope statement
4. Agile statement

Question 5. This allows you to release the software without a separate testing phase

1. Continuous integration
2. Done done
3. Collective code ownership
4. No bugs

Question 6. This reveals where the project is going and why it is going there.

1. Vision
2. Release planning
3. Risk management

4. Slack

Question 7. Product breakdown structure is identified in

1. Envision phase
2. Speculate phase
3. Explore phase
4. Adapt phase

Question 8. Following are the ways to improve velocity except:

1. Improve customer involvement
2. Offload programmer duties
3. Provide needed resources
4. Increase iteration length

Question 9. A normal sized team will typically work

1. 10 to 15 stories every iteration
2. 1 to 4 stories every iteration
3. There is no such standard
4. 4 to 10 stories every iteration

Question 10. All of the following are examples of special stories, except:

1. Spike stories
2. Documentation stories
3. Non-functional stories
4. Functional stories

Question 11. Which of the following enables testers to identify gaps in the team's thought process?

1. Customer test
2. TDD
3. Refactoring
4. Exploratory testing

Question 12. Following are the conflict response mode in Agile projects:

1. Competing, collaborating, compromising, accommodating and avoiding
2. Competing, coordinating, collaborating, avoiding and additive
3. Attentive, avoiding, accommodating, additive, advising
4. Competing, compromising, committing, complaining and avoiding

Question 13. The manifesto for Agile software development authored by the founding members of:

1. Agile alliance
2. Scrum alliance
3. Agile program leadership network
4. Agile project leadership network

Question 14. One particularly effective way of finding surprises, gaps and holes is:

1. Customer test
2. Exploratory testing

3. Unit testing
4. Blackbox testing

Question 15. The last phase of XP project is called:

1. Exploration phase
2. Planning phase
3. Iterations to Release phase
4. Productionizing phase

Question 16. In Agile projects the release plan is time boxed which means that the release date is defined in advance. What is NOT defined in this type of release plan?

1. Scope
2. Schedule
3. Cost
4. Quality

Question 17. Which of the following helps to understand your experiences?

1. Charters
2. Observation
3. Experience
4. Mindfulness

Question 18. All of the following are the strengths of XP, except:

1. Need for team maturity

2. Customer focused
3. High visibility
4. Quality via frequent testing

Question 19. All of the following are the characteristics of a good pilot project, except:

1. A project that hits all phases
2. Medium priority project
3. No external customer
4. External customer

Question 20. All of the following are the guidelines used to prioritize features except:

1. Features most important to the user
2. Features most important to the team
3. Features that provide value to the customer
4. Features that are high value and high risks

Question 21. The meaning of: You own up to your impact whether harm was intended and whether you think the other person should feel hurt or not:

1. Emotional intelligence
2. Emotional wake
3. Emotional quotient
4. Empiricism

Question 22. Which of the following will give the best estimating results?

1. Dis-aggregation
2. Analogy
3. Expert opinion
4. Combination of the above

Question 23. The extent to which people try to satisfy their own concerns is called

1. Assertiveness
2. Cooperativeness
3. Satisfaction
4. Celebration

Question 24. All of the following examples projects of business critical, except:

1. A project to ensure a revenue stream
2. A project with expiring funds
3. A project that supports meeting regulatory compliance
4. A project tied to a fixed bid

Question 25. Reconfirming commitment and removing impediments in Agile projects are usually part of:

1. Daily stand-up meetings
2. Frequent demos
3. Iterative planning
4. Retrospective meetings

Question 26. Software development is definitely:

1. Defined process
2. Empirical process

- 3. Predictable process
- 4. Non-empirical process

Question 27. This allows the team to reliably deliver results every iteration

- 1. Vision
- 2. Release planning
- 3. Risk management
- 4. Slack

Question 28. You have joined an Agile project where you found that few portions of the codes are familiar to only one to two members of the team and thus there is no collective ownership of the code that exists. Which XP practice you will employ to help other team members get familiar with those codes and bring in collective ownership feeling?

- 1. Coding Standards
- 2. Continuous Integration
- 3. Sustainable Pace
- 4. Pair Programming

Question 29. The more successful XP is, the more these worries grow

- 1. Organizational culture
- 2. Organizational structure
- 3. Organizational process
- 4. Organizational antibodies

Question 30. In SDLC, if we don't know what to do next, we just look at the life cycle documentation to determine the next step. This approach is most beneficial and suitable when we have

1. Experienced people
2. Inexperienced people
3. For mixed experienced people
4. For project managers

Question 31. The characteristic that make Agile easier to adopt

1. Lack of Agile knowledge
2. Fixed bid contract work
3. Distributed development
4. Customer availability

Question 32. Agile practice which calls for all stakeholders to be involved with the planning process is called as:

1. Collaborative planning
2. Progressive planning
3. Team planning
4. Agile planning

Question 33. Following are the career stage of an employee in an Agile project, except:

1. New employees

2. Individual contributors
3. Agile coach
4. Project manager

Question 34. A project team is creating detailed project progress charts weekly and displaying the same in the project location. Also, they are keeping those charts in their shared servers so that the onshore team and the customer can access those reports any time. This is an example of:

1. Information Technology
2. Team-space
3. Information Radiator
4. Secure documentation

Question 35. Which of the following is not true about feature cards and user stories?

1. Feature cards share the same goals as user stories
2. User stories and feature cards collect conversations with customers
3. Feature card has additional fields for dependencies
4. User stories has additional fields for dependencies

Question 36. Many Scrum team takes _____ days to complete in between iteration work

1. 1 or 2 days
2. 1 or 2 hours
3. 1 or 2 weeks

4. 1 or 2 months

Question 37. Which factor of technology is relatively new, but there are pockets of expertise to turn to—although the cost and availability of these experts might be an issue?

1. Well-known
2. Familiar
3. Leading edge
4. Bleeding edge

Question 38. It provides a way to analyse and improve the entire iterative cycle process

1. Root cause analysis
2. Informative work space
3. Pair programming
4. Retrospective

Question 39. How much delay is acceptable between starting an operation and receiving feedback about that operation?

1. Customer test
2. Responsiveness
3. Latency
4. Throughput

Question 40. Release map is usually for the duration

1. 12 months

- 2. 2 years
- 3. 3 months
- 4. 2 weeks

Question 41. The declaration of interdependence was developed with _____ in mind.

- 1. Project managers
- 2. Delivery managers
- 3. Software developers
- 4. Project leaders

Question 42. _____ act as a bridge between the wants and needs of the business and what the system does for the business users.

- 1. Use cases
- 2. Story cases
- 3. Capability cases
- 4. Requirement cases

Question 43. When dealing with non-customer facing stories. The first controversy is:

- 1. When to make priority decisions?
- 2. How to make priority decisions?
- 3. Who makes these priority decisions?
- 4. Why are we making priority decisions?

Question 44. Common characteristics of a Agile coach is called:

- 1. Native wiring

2. Distant wiring
3. Wireframe
4. Relative wiring

Question 45. Who creates customer tests in XP projects?

1. Customers
2. Customers and testers
3. Programmers
4. Testers

Question 46. All projects have constraints. The number one constraint is frequently

1. Scope
2. Quality
3. Cost
4. Time

Question 47. All of the following are limitations of feature cards, except:

1. Project complexity
2. Compliance and traceability
3. Identifying risks early
4. Customer is not available

Question 48. BDUF stands for

1. Broad design upfront
2. Big design upfront
3. Bid design upfront
4. Bring design upfront

Question 49. Succeed through a rapid progression of failures and drive by hindsight not foresight is called

1. Empiricism
2. Self-organization
3. Prioritization
4. Collaboration

Question 50. Due to what reason it is a good practice to add a unique story numbers to each user stories?

1. To create some work for the team
2. Product owner has instructed to do so
3. For easy estimation
4. It ensures traceability so that nothing gets missed out during implementation

Question 51. Which of the following allows programmers to be confident that their code does what they think it should?

1. Customer test
2. TDD
3. Refactoring
4. Spike solutions

Question 52. Which type of features can be ignored?

1. High risk high value
2. Low risk high value

3. High risk low value
4. Low risk low value

Question 53. Product owners need short business term goals to act as stepping stone and this is called as:

1. Microdefinitions
2. Macrodefinitions
3. Small definition
4. Short-cut definition

Question 54. If you have a large user population, then a great way to get information about how to prioritize the stories is

1. Questionnaires
2. Interview
3. Observation
4. Workshop

Question 55. Useful framework for coaching other coach to deepen their skills consists of three parts, namely,

1. Exploratory, apprentice, coach
2. Planning, execution, controlling
3. Card, conversation, condition
4. Exploratory, spike, mentoring

Question 56. After analyzing the product backlog and the project time line, if it is found

that there is not enough time to develop all the requested features, then what would be the **BEST** action?

1. Focus on delivering only the highest priority User Stories
2. Put more members in the team to deliver all features
3. Reduce the scope of some tasks and negotiate with the client
4. Increase team velocity by introducing new best practices

Question 57. High performance team live in a world of constant constructive agreement.

1. True
2. False

Question 58. While creating the release plan for a project the team is exploring the basic minimum features of the product so that people can start using it rather than waiting for something else. What is this set of features called?

1. Minimum marketable feature
2. Product backlog
3. Sprint backlog
4. Basic feature list

Question 59. You manage a customer support organization and your current focus is to effectively manage your support activities. The support requests arrive frequently (hourly), but

you want to make sure that you don't end up having a large and unmanageable queue of pending service requests at any point of time. In this situation, what approach is the best to choose?

1. Scrum
2. Six Sigma
3. Kanban
4. Lean

Question 60. Agile coach plays _____ role in the retrospective, especially for a new team

1. Mentor
2. Facilitator
3. Problem solver
4. Negotiator

Question 61. Following are the roles of the manager in an Agile project, except:

1. Value maximizer
2. Team champion
3. Boundary keeper
4. Business value driver

Question 62. While analyzing a project for Lean implementation the team found some regression works which is run for every module in parallel by different testing teams, but can be combined

to be run for all the modules together with one single team. Which Lean principle is this team employing during this analysis?

1. Deliver Fast
2. Eliminate Waste
3. Build Quality In
4. Create Knowledge

Question 63. A process of splitting the repository into distinct alternate histories is known as:

1. Reverting
2. Rollbacking
3. Branching
4. Merging

Question 64. A common timeframe for a release is

1. Monthly basis
2. Quarterly basis
3. Yearly basis
4. Daily basis

Question 65. Which one of the following is basically about reducing uncertainty through the systematic, proactive and early gathering of information?

1. Planning

2. Monitoring
3. Scanning
4. Sensing

Question 66. Which of the following is not an output of release planning phase?

1. Release plan
2. Risks
3. Release backlog
4. Product vision

Question 67. _____ is empowered to make whatever decisions necessary to complete its work.

1. The team
2. The project manager
3. The team lead
4. The project lead

Question 68. In an Agile project who can be thought of as an change control board

1. The product owner
2. The team
3. The sponsor
4. The project manager

Question 69. Who owns the product roadmap?

1. The project manager
2. The team
3. The customer

4. The Scrum master

Question 70. Iteration retrospective follows the iteration review.

1. True
2. False

Question 71. Agile project managers conducting individual performance appraisal is considered healthy and adds value.

1. True
2. False

Question 72. The best form of Agile software project status is

1. Weekly status report
2. Demo of working product
3. Cost management report
4. Daily status report

Question 73. Who is responsible for representing the interests of everyone with a stake in the project and its resulting system?

1. Product owner
2. The team
3. The Scrum master
4. The project manager

Question 74. In Agile project, others who might be interested in the project, but they aren't on the hook are called as:

1. Pig
2. Chicken
3. Hen
4. Goat

Question 75. The duration of sprint retrospective meeting is:

1. Three hours
2. Eight hours
3. Four hours
4. One hour

Question 76. Whose job is to protect the team from impediments during the sprint?

1. Project manager
2. The team lead
3. The Scrum master
4. Product owner

Question 77. Scrum defines _____ reports for the product owner and Scrum master to create at the end of each sprint.

1. One
2. Two
3. Three
4. Four

Question 78. When more than one Scrum team works simultaneously on a project, it is referred to as:

1. Multiple project
2. Multiple Scrum
3. Scaled project
4. Non-scaled project

Question 79. XP ensures _____ number of people review every piece of code.

1. One
2. Two
3. Three
4. Four

Question 80. XP produces working software

1. Every day
2. Every week
3. Every month
4. Every hour

Question 81. Which phase resulted in a well-articulated business or product vision?

1. The envision phase
2. The speculate phase
3. The explore phase
4. The adaptive phase

Question 82. In Agile projects the task level breakup of the user stories is done by the team. What is one of the main benefits of this being done as a team rather than individually?

1. It keeps the team busy
2. This sometimes brings out the perspectives that a single developer or tester would have missed during the actual software development.
3. This ensures that the product owner knows each team member
4. This should have been done by individuals

Question 83. XP does not require experts; it does require a habit of mindfulness.

1. True
2. False

Question 84. A mistake proof process is achievable and desirable.

1. True
2. False

Question 85. Which provides a template for seamlessly joining the teams work together?

1. Trust
2. Sitting together
3. Ubiquitous language
4. Coding standards

Question 86. A buffer iteration is same as hardening iteration.

- 1. True
- 2. False

Question 87. Reliance on overtime indicates a systemic problem.

- 1. True
- 2. False

Question 88. One powerful way to design your application to speak the language of the domain is to create a:

- 1. Process
- 2. Document
- 3. Domain model
- 4. Domain expertise

Question 89. Who can dominate in stand-up meetings?

- 1. The customer
- 2. The team
- 3. The product owner
- 4. It is a meeting of equals

Question 90. Creating a coding standard is an exercise in building

- 1. Trust

2. Confidence
3. Consensus
4. Assurance

Question 91. Team inspects and adapts in which process?

1. Iteration retrospective
2. Iteration planning
3. Iteration demo
4. Iteration review

Question 92. Which of the following is true about Agile coach?

1. Coach will move towards coordinating individual contributions
2. Coach will move towards driving
3. Coach will move towards letting the team find their own way
4. Coach will move towards fixing problems

Question 93. Which of the following is a way to find unexpected bugs?

1. Unit testing
2. System testing
3. Exploratory testing
4. Integration testing

Question 94. A process of combining multiple changes and resolving any conflicts in version control is known as:

1. Reverting

2. Rollbacking
3. Branching
4. Merging

Question 95. An Agile project team started to estimate new stories by comparing the size with a group of 3 stories that have already been estimated. What this type of estimation commonly known as?

1. Three-point estimation
2. Disaggregation
3. Relative sizing
4. Guess Estimation

Question 96. Creating story for ‘get data’, ‘validate data’, ‘write to database’ is an example of

1. Vertical strips
2. Horizontal strips
3. Technical strips
4. Domain strips

Question 97. Which of the following says that in part, that every system has a single constraint that determines the overall throughput of the system?

1. Velocity
2. Mindfulness
3. TOC

4. Iterations

Question 98. Which of the following reflects how much time or money you need to set aside to contain the risk?

1. Mitigation activities
2. Contingency activities
3. Transition indicators
4. Risk exposure

Question 99. On a story-driven project the customers and users remain involved throughout the duration of the project.

1. True
2. False

Question 100. An epic that comprises multiple shorter stories is called as

1. Complex Story
2. Compound Story
3. Big story
4. Confused story

Question 101. Which of the following is the better word to describe requirement collection in terms of user stories?

1. Collecting
2. Elicitation
3. Trawling

4. Identification

Question 102. During a story-writing workshop the focus should be on _____ rather than _____.

1. Quality, quantity
2. Quality, risk
3. Risk, quality
4. Quantity, quality

Question 103. An Agile project team is creating a release plan. They first defined the release dates but kept the decision of the specific work people will be doing pending for later. What type of release plan are they creating?

1. Time-boxed release plan
2. Scope-boxed release plan
3. Uncertain release plan
4. Adaptive release plan

Question 104. You are encouraging XP methodology in your organization for reducing the risk of dependency on the individuals so that if anyone goes for long leave, falls sick or resigns then there is not a productivity loss of the team for training the new person with all the knowledge that the other person had. Which of

XP practice addresses these problems best of all, reducing them almost entirely?

1. Sustainable Pace
2. Coding Standards
3. Collective Code Ownership
4. Continuous Integration

Question 105. User scenarios have the following characteristics, except:

1. Settings
2. Actors
3. Goals
4. Relations

Question 106. Agile business objective, which is to deliver on future customer requirements?

1. Continuous innovation
2. Product adaptability
3. People and process adaptability
4. Reliable results

Question 107. In which phase the team hypothesizes about the specifications of the product and builds a release plan?

1. The envision phase
2. The speculate phase
3. The explore phase
4. The adaptive phase

Question 108. The top goal of a project team is:

1. Deliver value to customers
2. Deliver products as per the schedule
3. Deliver products as per the cost estimated
4. Delivery high quality products

Question 109. While analyzing a burn-up chart, you found that the total story point line has gone up on the 5th day of the iteration. Which might be the reason for this?

1. The graph was plotted wrongly
2. New user story added to the sprint increasing the total story point of the sprint
3. Some user stories have been removed from the sprint
4. Team is working harder to achieve more tasks

Question 110. Which of the following is not one of the three particularly important issues involved in delivering customer value?

1. Focusing on innovation
2. Concentrating on execution
3. Lean thinking
4. Efficiency and optimization

Question 111. Which kind of sequence is the most popular for estimating user stories in an Agile project?

1. Non-linear
2. Fibonacci sequence

- 3. Linear
- 4. Story points

Question 112. When project leaders focus on delivery, they add _____ to projects. When they focus on planning and control, they tend to add _____

- 1. Quality, overhead
- 2. Overhead, quality
- 3. Value, overhead
- 4. Overhead, value

Question 113. The exploration factor's technology dimension has _____ categories.

- 1. One
- 2. Two
- 3. Three
- 4. Four

Question 114. Choose the four categories of the exploration factor's technology dimension

- 1. Bleeding edge, leading edge, familiar, well-known
- 2. Familiar, not familiar, known, unknown
- 3. Known, unknown, risk, no risk
- 4. Adaptive, flexible, manageable, risky

Question 115. ICS in Agile project stands for:

- 1. Internal customer stories

2. Internal customized stories
3. Inter-team commitment story
4. Inter-team customer story

Question 116. Kanban uses the following principle:

1. Push system
2. Interactive system
3. Pull system
4. Invert system

Question 117. The high WIP problem is addressed by:

1. Scrum
2. Kanban
3. FDD
4. TDD

Question 118. Signalling system into the scheduling process is introduced by:

1. Scrum
2. Kanban
3. FDD
4. TDD

Question 119. Increased customer involvement leads to all of the following benefits, except:

1. Quicker feedback
2. Accurate delivery
3. Rapid decisions

4. Higher ROI

Question 120. In Agile the measure of success:

1. Client getting more business value
2. Delivering project on time
3. Delivering project within budget
4. Delivering project as per customer expectation

Answer Sheet for Model Question Paper 3

Question Number	Answer	Question Number	Answer	Question Number	Answer
Question 1		Question 41		Question 81	
Question 2		Question 42		Question 82	
Question 3		Question 43		Question 83	
Question 4		Question 44		Question 84	
Question 5		Question 45		Question 85	
Question 6		Question 46		Question 86	
Question 7		Question 47		Question 87	
Question 8		Question 48		Question 88	
Question 9		Question 49		Question 89	
Question 10		Question 50		Question 90	
Question 11		Question 51		Question 91	
Question 12		Question 52		Question 92	
Question 13		Question 53		Question 93	
Question 14		Question 54		Question 94	
Question 15		Question 55		Question 95	
Question 16		Question 56		Question 96	
Question 17		Question 57		Question 97	
Question 18		Question 58		Question 98	
Question 19		Question 59		Question 99	
Question 20		Question 60		Question 100	
Question 21		Question 61		Question 101	
Question 22		Question 62		Question 102	
Question 23		Question 63		Question 103	
Question 24		Question 64		Question 104	
Question 25		Question 65		Question 105	

Question 26	Question 66	Question 106	
Question 27	Question 67	Question 107	
Question 28	Question 68	Question 108	
Question 29	Question 69	Question 109	
Question 30	Question 70	Question 110	
Question 31	Question 71	Question 111	
Question 32	Question 72	Question 112	
Question 33	Question 73	Question 113	
Question 34	Question 74	Question 114	
Question 35	Question 75	Question 115	
Question 36	Question 76	Question 116	
Question 37	Question 77	Question 117	
Question 38	Question 78	Question 118	
Question 39	Question 79	Question 119	
Question 40	Question 80	Question 120	

Answers for Model Question Paper 3

Question Number	Answer	Question Number	Answer	Question Number	Answer
Question 1	C	Question 41	D	Question 81	A
Question 2	D	Question 42	A	Question 82	B
Question 3	C	Question 43	C	Question 83	A
Question 4	B	Question 44	A	Question 84	B
Question 5	D	Question 45	B	Question 85	D
Question 6	A	Question 46	D	Question 86	B
Question 7	A	Question 47	C	Question 87	A
Question 8	D	Question 48	B	Question 88	C
Question 9	D	Question 49	A	Question 89	D
Question 10	D	Question 50	D	Question 90	C
Question 11	D	Question 51	B	Question 91	A
Question 12	A	Question 52	C	Question 92	C
Question 13	A	Question 53	A	Question 93	C
Question 14	B	Question 54	A	Question 94	D
Question 15	D	Question 55	A	Question 95	A
Question 16	A	Question 56	A	Question 96	B
Question 17	D	Question 57	B	Question 97	C
Question 18	A	Question 58	A	Question 98	D
Question 19	D	Question 59	C	Question 99	A

Question 20	B	Question 60	B	Question 100	B
Question 21	B	Question 61	D	Question 101	C
Question 22	D	Question 62	D	Question 102	D
Question 23	A	Question 63	C	Question 103	A
Question 24	D	Question 64	B	Question 104	C
Question 25	A	Question 65	C	Question 105	D
Question 26	B	Question 66	D	Question 106	B
Question 27	D	Question 67	A	Question 107	B
Question 28	D	Question 68	B	Question 108	A
Question 29	D	Question 69	C	Question 109	B
Question 30	B	Question 70	A	Question 110	D
Question 31	D	Question 71	B	Question 111	B
Question 32	A	Question 72	B	Question 112	C
Question 33	D	Question 73	A	Question 113	D
Question 34	C	Question 74	B	Question 114	A
Question 35	D	Question 75	A	Question 115	C
Question 36	A	Question 76	C	Question 116	C
Question 37	C	Question 77	D	Question 117	B
Question 38	D	Question 78	C	Question 118	B
Question 39	B	Question 79	B	Question 119	D
Question 40	A	Question 80	B	Question 120	A

Model Question Paper–4

- This Practice test is based on the actual syllabus for Agile certification examination
- Please set yourself a time of 3 hours
- Mark your answer on the template sheet provided at the end
- Strictly stop the test when the clock hits 3 hours
- The correct answers are provided at the end of this question set
- Give one mark for each correct answer
- There is no negative marking for wrong answers
- Practice this test multiple times for better results
- All the very Best!

Questions

Question 1. An Agile coach has gathered the team together who will be working in next 10 iterations. The most important value that the Agile coach will depend on for the team to thrive is:

1. Trust
2. Sitting together
3. Real customer involvement
4. Ubiquitous language

Important Points

1. An information radiator displays information in a place where passersby can see it. With information radiators, the passersby don't need to ask questions; the information simply hits them as they pass.
2. A good information radiator
 1. Is large and easily visible to the casual, interested observer
 2. It takes very little energy to view the display.
 3. Is understood at a glance
 4. Changes periodically, so that it is worth visiting
 5. Is easily kept up to date
 6. Information radiators can be used on any project, large or small
3. William Pietri took a stab at defining how you could identify a bad team space easily. He listed the following points
 1. People wearing headphones
 2. Stale artifacts on the walls
 3. Workspace as information desert
 4. Minimal interaction
 5. Furniture as barrier
 6. Sad or ugly spaces
 7. Seating by job description
 8. Space and furniture as status markers
 9. No laughter, no fun
4. Osmotic communication means information flows into the background hearing of members of the team, so that they

pick up relevant information as in osmosis.

5. Osmotic communication makes the cost of communications low and the feedback rate high, so that errors are corrected extremely fast, and knowledge is disseminated quickly.
6. Although osmotic communication is valuable for larger projects, it is, of course, increasingly difficult to attain as the team's size grows.
7. Retrospective planning game: team members brainstorm the task necessary to accomplish a goal and signup for the tasks they will tackle in order to complete the project.
8. Debate is not part of retrospective
9. A more effective tool to use in gathering data for project retrospectives would be timeline.
10. Signal card containing a small amount of work or story to develop: Kanban
11. Issue of too much WIP addressed by Kanban
12. In planning phase-setting a fixed time limit to overall development efforts is called as time-boxing
13. Time-boxes are used as a form of risk management, especially for tasks that may easily extend past their deadlines.
14. In Scrum, we have 6 formal time-boxed events.
 1. Releasing Planning
 2. Sprint Planning
 3. Sprint
 4. Sprint Review
 5. Sprint Retrospective
 6. Daily Scrum
15. Time-boxing can greatly improve productivity and effectiveness.
16. Time-boxes help manage complexity by determining what can be achieved in a short duration.
17. In release burn-down chart, the region below the horizontal axis indicates the overall work added to release.

18. The bars in release burn-down chart represent the amount of work in the release.
19. If features are being removed the bottom of the bar is raised to reflect this change.
20. When new stories are added to its work stack: burn-down chart report burn-up chart
21. Release burn-down chart vertical axis indicates (story points remaining in the project)
22. Process tailoring: When seeking to implement Agile methodology into the work flow of teams who already have established and entrenched process.
23. User stories originated as a part of XP
24. Triangulating is a process in Agile estimation which is estimating a story based upon its relationship with other stories.
25. Questionaries don't lend themselves to follow-up questions and hence are not the primary technique for trawling for new user stories.
26. User stories should be written by customers of a project.
27. A well-written user story has three components, namely, card, conversation and confirmation.
28. Story points represent cost and value points represent benefit.
29. Zero point estimate: the story requires trivial effort
30. Normal time estimate for research story: 1 day
31. The emphasis in user stories should not be on written communication rather increase collaboration, encourage differing of detail and emphasize on verbal communication.
32. Allow as many rounds as it takes to converge while playing poker.
33. Three assumptions that must be made when estimating ideal days are that the story being estimated is the only thing you will work on, that everything you need will be on hand when you start and there will be no interruptions.

34. In Agile project, the output of negotiation contracts to procure vendors and contractors.
35. With respect to Agile governance, executives are most interested in investments.
36. Governance in Agile environment means making decision in an uncertain environment.
37. Agile project governance team would be interested in both investment decisions and risks.
38. Concepts prevent duplication of code—DRY (Dont repeat yourself)
39. Which information radiator is MOST useful in aligning the team's effort in the right direction?: project priority chart.
40. Most essential for developing collective code ownership: collaboration.
41. For actively cultivating innovation, it requires visualization and commonality.
42. Exploration process are designed to deliver innovation reliably
43. Five key business objectives of Agile project management are
Continuous innovation, product adaptability, improved time to market, people and process capability, and reliable results.
44. A buffer iteration is used to allocate contingency time for final finishing features.
45. A hardening iteration is used for preparing items such as marketing materials.
46. Release backlog features are measured in story point, while iteration backlog items are estimated in hours.
47. Cumulative flow diagram shows visual of all the story points completed through every iteration.
48. Benefit of short iterations: They allow the team to receive timely feedback.
49. Three levels of planning in Agile: release planning, iteration planning and daily planning.

50. Exploration process is designed to deliver innovation reliably
51. Effective release plan is based on timeframe
52. The primary objective of Agile risk planning is to enable the team to meet their long-term commitments.
53. The primary tool used in Agile release planning is index cards.
54. An example for index card-based planning is blitz planning.
55. Release plan helps to develop a better understanding of the project viability.
56. Learners at detaching stage look for limitations of the procedure.
57. Daily stand-up is concise discussions of project status with the entire team.
58. Daily stand-up meetings, daily interaction with product team and stakeholder coordination are ways of encouraging collaboration and coordination
59. Standing helps the meeting to be short.
60. To report the status to the customer all of the following are being used: formal documentation, informal documentation and daily stand-ups.
61. Success of a project is determined by the maximum value that was delivered to the customer.
62. The following demonstrates a measure of value delivered to the customer: extrinsic quality and releaseable product.
63. Three performance parameters used for Agile projects are value, quality and constraints (value is the primary goal).
64. Business goals, Agile values, organization, product pack and process are major components of an Agile scaling model (ASM).
65. Agile teams generally achieve low bug rates
66. ROI, customer satisfaction are the metrics that can be standardized across the team.
67. On an Agile, ‘The Team’ determines how much of the product backlog can be delivered in the upcoming sprint.

68. Tool utilized by distributed team to maintain high level of communication is Wikishare portal.
69. In high performance team, leaders manage principles.
70. The entire team is change controllers in an Agile project.
71. Temperature reading is used to determine what the team wants.
72. First step in building high performance team is to set the expectation.
73. Agile team's productivity metrics: cycle time, release burn-downs and customer support.
74. Programmers should master the practice of spike solution.
75. In high performance team, principles manage the team.
76. The purpose of the risk mitigation is to reduce the impact.
77. An output of catastrophes brainstorming session is a risk census (Risk register).
78. Transition indicator is a signal when a risk is close to occurring.
79. An Agile coach facilitates by creating mediums to collect team generated ideals and innovations. These collection medium are referred as 'Containers'.
80. Coaching style that can be applied where team members have trouble with compliance is teaching
81. The code of professional conduct for Agile coaches holds that coaches exist for the coaches.
82. Period time for release is 12 months.
83. When the team delivered 23 story points, the top will move down 23 points in release burn-down bar.
84. Conflict resolution allows teams to preserve trust amongst members
85. Preferred level of conflict within team is LEVEL 1.
86. When one person complains about another: Three step intervention approach of conflict is being used.
87. The sum of knowledge all of the people in the team is called TACIT knowledge.
88. Three ways to get initial value for velocity: use historical values, run initial iteration and use that velocity, and take

a guess.

89. Commit build process should have both unit test and integration test.
90. The test which validates the functionality to be delivered and meets customer expectations: acceptance test.
91. Exploratory testings help to catch bugs related to security and concurrency in the application.
92. The purpose of ruthless automated testing: to ensure product quality remains high throughout the development process.
93. XP team are advised to integrate their code a minimum of once per day
94. In XP, customers are most qualified to prioritize product features.
95. Most difficult hurdles that managers must overcome in cost accounting and reporting of project is capitalization vs. expense.
96. Relative weighting priority of feature is determined by dividing the priority % by the cost %.
97. Erg second is a measure of cost to get questions answered within a team.
98. Lost opportunity cost: The cost of fixing errors introduced due to team members not communicating with one another.
99. To improve time to market an Agile, the project manager should consider both the number of features and the depth of the features.
100. Kano model's most important product is threshold features/MUST have features(threshold, linear, excitor in order).
101. Change report is generated at the end of each sprint.
102. Change report is BEST described as differences between the product backlogs at the start of previous sprint and at the start of new sprint.
103. Design principles that provides a guideline to design relationship between concepts in a design: Decoupling.

104. The most effective and efficient way according to the Agile manifesto to convey the information throughout the team is to use face-to-face communication.
105. Product owner represents the stakeholder and the business.
106. Don't wait when issues comes ... discuss with the team immediately.
107. The Agile management tool that allow teams to track pipeline delays and queue sizes: Cumulative flow diagram.
108. Continuously practising the application of active listening frameworks helps to develop active listening skills within Agile groups.
109. Version control system is accessible to both the development team and the customer.
110. The period time for roadmap is 2 years.
111. The period time for wave is 3 months.
112. Specific roles ensure customer collaboration.
113. Group thinking should be avoided while generating ideas.
114. Project trade-off matrix is used to view the relative importance of the project constraints and determines whether they are fixed, flexible or acceptable.
115. Complex adaptive system theory (CAS): the tacit and explicit information exchanges between stakeholders involved in the project.
116. Best approach to estimation is always the combination of expert opinion, analogy and disaggregation.
117. In Agile environment projects are divided into two areas: production and exploration.
118. Exploration means: unknown problem and known solution, unknown problem and unknown solution, known problem and unknown solution.
119. Strategy for creating trust in Agile team are customer programmer empathy, programmer tester empathy, eating together and team continuity.

120. The preparation of business case can assist in obtaining project approval.
121. A spike is best described as a brief learning period.
122. Agile philosophy for failure of project: Fail fast.
123. Collaboration involves working together to jointly produce a deliverable or make a decision, whereas communication involves sharing of information.
124. The best place to store sandbox is on local development machine.
125. Version control with concurrent model allows multiple developers to edit the same file simultaneously; emphasis on user stories should not be on technical competencies.
126. Time travelling to fix bugs: Checking previous versions to identify the root cause of a bug.
127. Project data sheet is a one page summary of key project management information, business objectives and product capabilities.
128. Think of vendors as part of the same team: ‘They are also us’.
129. Keeping related concepts closer together is called cohesion.
130. Six sigma technique adopted by Agile PMO: Pareto diagrams and RCA.
131. Sitting together reduces time to market by one-third.
132. Health check and questionnaires can be used to find problems in a process.
133. Learners at level 1 look for one procedure that works.
134. Multi-stage integration builds for: running additional tests for performance, load or stability
135. Five failure modes: People making mistakes, preference in conservative failures, inventing over researching, forming habits and inconsistencies.
136. Agile enterprise framework has 4 layers: portfolio governance layer, project management layer, iteration management layer and technical practice layer.

137. Speculate phase: develop a capability and/or feature based release plan to deliver on the vision.
138. Reports generation should be considered as a separate story.
139. Vertical market software are developed for many organisations.
140. The correct order of APM delivery frame work: envision, speculate, explore, adapt and close.
141. In high performance company, core values and distinguishing purpose remain constant while operating practices and business strategies change.
142. The role of PMO: Provides insight to stakeholders of value delivered.
143. Shotgun debugging: trying many solutions at once.
144. Exciter feature: add price premium to the product.
145. Cost of delivering an Agile project is best determined by delivery team during release planning phase.
146. Report that is more detailed than vision statement: Roadmap.
147. Team does not know whether the design will work out or not. It is recommended to do _____(Spikes) to find out which will work out for them.
148. Person best suited and most appropriate to monitor all risks in Agile projects: Project manager.
149. Agility is the ability to manage flexibility and stability.
150. To serve as an example of a skilled and capable Agile practitioner, one must be a model of Agile practices.
151. Types of successes are personal success, organizational success and technical success.
152. Organizational success is measured by ROI
153. Apart from revenue and cost savings, sources of organisational value include:
 1. Competitive differentiation
 2. Brand projection
 3. Enhanced customer loyalty
 4. Satisfying regulatory requirements

5. Original research
 6. Strategic information
154. Agile projects release their most valuable features first and release new versions frequently, which dramatically increase value.
155. Agile manifesto is a collection of 4 values and 12 principles.
156. Agile manifesto values
1. Individuals and interactions over processes and tools
 2. Working software over comprehensive documentation
 3. Customer collaboration over contract negotiation
 4. Responding to change over following a plan
157. XP emphasizes face-to-face collaboration
158. In XP, customers and testers create the customer tests for a story around the same time that programmers implement the story.
159. This work is driven by test-driven development (TDD), an activity that inextricably weaves together testing, coding, design and architecture.
160. TDD produces automated unit and integration tests.
161. They use exploratory testing to look for surprises and gaps in the software.
162. When the testers find a bug, the team conducts root-cause analysis and considers how to improve their process to prevent similar bugs from occurring in the future.
163. A well-functioning XP teams produce only a handful of bugs per month in a completed work.
164. Customers' most important activity in XP is release planning.
165. XP uses requirements documents only as memory aids for customers.
166. Typically, product managers, domain experts, interaction designers and business analysts play the role of the on-site customer.

167. As a rule of thumb, start with two on-site customers (including the product manager) for every three programmers. As a rule of thumb, start with one tester for every four programmers.
168. The product manager has only one job on an XP project, - to maintain and promote the product vision.
169. The best product managers have deep understandings of their markets.
170. Interaction designers(UI) perform such tasks as interviewing users, creating user personas, reviewing paper prototypes with users, and observing usage of actual software.
171. If the customers' job is to maximize the value of the product, then the programmers' job is to minimize its cost.
172. XP leaders are called coaches.
173. The coach differs from your mentor. Your mentor is someone outside the team who you can turn to for advice.
174. At a minimum, however, XP prefer to see one person clearly designated as 'product manager' (who may do other customery things) and one person clearly defined as 'programmer-coach' (who also does programmery things).
175. Refactoring is the process of changing the structure of code—rephrasing it—with changing its meaning or behaviour.
176. Technical debt is the total amount of less-than-perfect design and implementation decisions in your project.
177. Stories are customer-centric, describing the results in terms of business results.
178. Velocity has no relation to productivity.
179. TOC—Theory of Constraints. Programmers are the constraint.
180. Agility—the ability to respond effectively to change—requires that everyone pays attention to the process and practices of development. This is mindfulness.
181. XP's applicability is based on organizations and people, not types of projects.

182. For XP

1. Prerequisite #1: Management Support
2. Prerequisite #2: Team Agreement
3. Prerequisite #3: A Collocated Team
4. Prerequisite #4: On-site Customers
5. Prerequisite #5: The Right Team Size
6. Prerequisite #6: Use All the Practices

183. For XP

1. Recommendation #1: A Brand-New Codebase
2. Recommendation #2: Strong Design Skills
3. Recommendation #3: A Language that's Easy to Refactor
4. Recommendation #4: An Experienced Programmer-Coach

184. XP strongly recommends 40 hr week - 5 working days of 8 hr each. Regular overtime is the signal of a problem in the project.

185. The first team is often very successful, inspiring the organization to use XP on more projects, but then this second wave of XP projects struggles, called second adaptor syndrome.

186. Other than change itself, the biggest challenge in applying XP to an existing project is not writing tests, refactoring or cleaning up your bug database. The biggest challenge is setting aside enough time to pay down technical debt.

187. Technical debt is a big challenge in applying XP to an existing project.

188. The more slack you provide for paying down technical debt, the lower your velocity will be, but the less time it will take for your velocity to rise again.

189. Think of velocity as cash flow: the more principal you pay on your debt, the less cash you have each week, but the more quickly you can stop paying interest.

190. Fortunately, as your technical debt decreases, your velocity will rise again.

191. XP assumes that you use iterations, not phases, which makes using XP in a phase-based environment difficult.
192. Assess your agility: Thinking, planning, developing, releasing, collaborating.
193. XP does not require experts. It does require a habit of mindfulness.
194. Pair programming doubles the brainpower available during coding, and gives one person in each pair the opportunity to think about strategic, long-term issues.
195. Energized work acknowledges that developers do their best, most productive work when they are energized and motivated.
196. An informative workspace gives the whole team more opportunities to notice what's working well and what isn't.
197. Root-cause analysis is a useful tool for identifying the underlying causes of your problems.
198. Retrospectives provide a way to analyse and improve the entire development process.
199. To practice communicating and switching roles while pairing, consider ping-pong pairing.
200. An informative workspace broadcasts information into the room.
201. Gaming occurs when people try to improve a number at the expense of overall progress.
202. Iteration retrospectives, release retrospectives, project retrospectives and surprise retrospectives.
203. Mute mapping is a variant of affinity mapping in which no one speaks. Its a great way to categorize a lot of ideas quickly.
204. Trust is essential for the team to thrive (We work together effectively and without fear).
205. Sitting together leads to fast, accurate communication.
206. Real customer involvement helps the team understand what to build.

- 207. A ubiquitous language helps team members understand each other.
- 208. Stand-up meetings keep team members informed.
- 209. Coding standards provide a template for seamlessly joining the team's work together.
- 210. Iteration demos keep the team's efforts aligned with stakeholder goals.
- 211. Reporting helps reassure the organization that the team is working well.
- 212. Trust
 - 1. Team Strategy #1: Customer–Programmer Empathy
 - 2. Team Strategy #2: Programmer–Tester Empathy
 - 3. Team Strategy #3: Eat Together
 - 4. Team Strategy #4: Team Continuity
- 213. Consider the team as a resource rather than people as a resource.
- 214. The more successful XP is the more these worries grow. Alistair Cockburn calls them organizational antibodies.
- 215. When a problem does occur, you should usually be able to solve it by using slack, not overtime.
- 216. In the case of a software team, hustle is energized, productive work.
 - 1. Organizational Strategy #1: Show Some Hustle
 - 2. Organizational Strategy #2: Deliver on Commitments
 - 3. Organizational Strategy #3: Manage Problems
 - 4. Organizational Strategy #4: Respect Customer Goals
 - 5. Organizational Strategy #5: Promote the Team
 - 6. Organizational Strategy #6: Be Honest
- 217. Tunnel vision—you can get so caught up in the daily details of the project that you lose track of your real customers' interests.
- 218. Unlike custom development, vertical-market software is developed for many organizations. Like custom development, however, it's built for a particular industry and it's often customized for each customer.

219. Horizontal-market software: software that's intended to be used across a wide range of industries. (Instead, find other ways to involve customers: focus groups, user experience testing, community previews, beta releases and so forth.)
220. Stand-up Meetings—The primary virtue of the stand-up meeting is brevity.
221. Progress Reports: Reports on the progress of the team, such as an iteration demo or a release plan.
 1. Vision statement
 2. Weekly demo
 3. Release and iteration plans
 4. Burn-up chart
 5. Roadmap
 6. Status e-mail
222. Management Report: high level status of the project
 1. Productivity
 2. Throughput
 3. Defects
 4. Time usage
223. Throughput is the number of features the team can develop in a particular amount of time.
224. Reports to avoid
 1. Number of stories
 2. Velocity
 3. Code quality
225. A normal-sized team will typically work on 4 to 10 stories in every iteration.
226. ‘Done done’ ensures that completed work is ready to release.
227. Version control allows team members to work together without stepping on each other’s toes.
228. A ten-minute build builds a tested release package in under 10 minutes.
229. Continuous integration prevents a long, risky integration phase.

230. Collective code ownership allows the team to solve problems no matter where they may lie.
231. To achieve zero bugs result, XP uses a potent cocktail of techniques
 1. Write fewer bugs by using a wide variety of technical and organizational practices.
 2. Eliminate bug breeding grounds by refactoring poorly designed code.
 3. Fix bugs quickly to reduce their impact, write tests to prevent them from reoccurring, then fix the associated design flaws that are likely to breed more bugs.
 4. Test your process by using exploratory testing to expose systemic problems and hidden assumptions.
 5. Fix your process by uncovering categories of mistakes and making those mistakes impossible.
232. Collective code ownership spreads responsibility for maintaining the code to all the programmers.
233. A typical alternative to collective code ownership is strong code ownership, in which each module has a specific owner and only that person may make changes. A variant is weak code ownership, in which one person owns a module but others can make changes as long as they coordinate with the owner.
234. Vision reveals where the project is going and why it's going there.
235. Release planning provides a roadmap for reaching your destination.
236. The planning game combines the expertise of the whole team to create achievable plans.
237. Risk management allows the team to make and meet long-term commitments.
238. Iteration planning provides structure to the team's daily activities.
239. Slack allows the team to reliably deliver results every iteration.

240. Stories form the line items in the team's plan.
241. Estimating enables the team to predict how long its work will take.
242. The vision statement documents three things: what the project should accomplish, why it is valuable and the project's success criteria.
243. Release early, Release often.
244. A minimum marketable feature, or MMF, is the smallest set of functionality that provides value to your market, whether that market is internal users (as with custom software) or external customers (as with commercial software).
245. MMFs provide value in many ways such as competitive differentiation, revenue generation and cost savings.
246. Adapt Your Plans
 1. Keep Your Options Open
 2. Planning at the Last Responsible Moment
 3. The Planning Game
247. Suppose you are creating a system that gets data from a user, validates the data and writes it to a database. You might initially create a story for each step: 'Get data', 'Validate data', and 'Write data to database'. These are sometimes called horizontal stripes.
248. A better approach is to create stories that do all three tasks but provide narrower individual utility. For example, you might create the stories 'Process customer data', 'Process shipping address', and 'Process billing information'. These are vertical stripes.
249. There are two basic types of plans: scope-boxed plans and time-boxed plans.
250. A scope-boxed plan defines the features the team will build in advance, but the release date is uncertain.
251. A time-boxed plan defines the release date in advance, but the specific features that release will include are uncertain.
252. XP assumes that customers have the most information about value: what best serves the organization.

- 253. Programmers have the most information about costs: what it will take to implement and maintain those features.
- 254. Create a risk census, that is, a list of the risks your project faces that focuses on your project's unique risks.
- 255. Risk exposure reflects how much time or money you should set aside to contain the risk.
- 256. $\text{Risk_adjusted_points_remaining} = (\text{iterations_remaining} - \text{risk_exposure}) \times \text{velocity} / \text{risk_multiplier}$
- 257. For example, if you are using a rigorous approach, your release is 12 iterations away, your velocity is 14 points, and your risk exposure is one iteration, you would calculate the range of possibilities as:
 1. Points remaining = $(12 - 1) \times 14 = 154$ points
 2. 10 % chance: $154/1 = 154$ points
 3. 50 % chance: $154/1.4 = 110$ points
 4. 90 % chance: $154/1.8 = 86$ points
 5. In other words, when it is time to release, you are 90% likely to have finished 86 more points of work, 50% likely to have finished 110 more points, and only 10% likely to have finished 154 more points.
- 258. Some teams have a dedicated phone for emergency support requests; this is, of course, the bat-phone.
- 259. The amount of slack you need does not depend on the number of problems you face. It depends on the randomness of problems.
- 260. A good rule of thumb is to spend 10% of the iteration on technical debt.
- 261. Dedicated research time is an excellent way to encourage learning and adding additional slack into your iterations (People improvement).
- 262. Stories represent customer value and are written in the customers' terminology. (The best stories are actually written by customers.)
- 263. Stories describe an end-result that the customer values, not implementation details.

- 264. Stories have clear completion criteria. Customers can describe an objective test that would allow programmers to tell when they have successfully implemented the story.
- 265. Special type of stories
 - 1. Documentation stories
 - 2. 'Non-functional' stories
 - 3. Bug stories
 - 4. Spike stories (Research to do something)
 - 5. Estimating
 - 6. Meetings
- 266. Velocity relies upon a strict iteration time-box. To make velocity work, never count stories that aren't 'done done' at the end of the iteration. Never allow the iteration deadline to slip, not even by a few hours.
- 267. You can have accurate estimates if you:
 - 1. Estimate in terms of ideal engineering days (story points), not calendar time
 - 2. Use velocity to determine how many story points the team can finish in an iteration
 - 3. Use iteration slack to smooth over surprises and deliver on time, every iteration
 - 4. Use risk management to adjust for risks to the overall release plan
- 268. Improving Velocity
 - 1. Pay down technical debt
 - 2. Improve customer involvement
 - 3. Support energized work
 - 4. Provide needed resources
 - 5. Offload programmer duties
 - 6. Adding programmers
- 269. Incremental requirements allows the team to get started while customers work out requirements details.
- 270. Customer tests help communicate tricky domain rules.
- 271. Test-driven Development allows programmers to be confident that their code does what they think it should.

272. Refactoring enables programmers to improve code quality without changing its behaviour.
273. Simple design allows the design to change to support any feature request, no matter how surprising.
274. Incremental design and architecture allows programmers to work on features in parallel with technical infrastructure.
275. Spike solutions use controlled experiments to provide information.
276. Performance optimization uses hard data to drive optimization efforts.
277. Exploratory testing enables testers to identify gaps in the team's thought processes.
278. Divergent change occurs when unrelated changes affect the same class. It's an indication that your class involves too many concepts. Split it, and give each concept its own home.
279. Shotgun surgery is just the opposite: it occurs when you have to modify multiple classes to support changes to a single idea. This indicates that the concept is represented in many places throughout your code. Find or create a single home for the idea.
280. Data clumps are similar to primitive obsession. They occur when several primitives represent a concept as a group.
281. When you have a class that contains methods but no meaningful per-object state, you have a wannabe static class.
282. When you refactor, proceed in a series of small transformations. (Confusingly, each type of transformation is also called a refactoring.)
283. You Aren't Gonna Need It (YAGNI)
284. 'Don't repeat yourself', or the DRY principle.
285. A spike solution, or spike, is a technical investigation. It's a small experiment to research the answer to a problem.

286. Exploratory testers use the following four tools to explore the software.
1. Tool #1: Charters (Charters are the testing equivalent of a story)
 2. Tool #2: Observation
 3. Tool #3: Note-taking
 4. Tool #4: Heuristics
287. CRUD: create, read, update, delete
288. Experience helps you see how Agile methods work.
Mindfulness helps you understand your experiences.
289. Improve the Process
- Test-driven development, exploratory testing, real customer involvement, iteration demos and frequent releases all provide information about the project, from code to user response.
290. Rely on People: (Continuous improvement)
291. Build Effective Relationships
292. Let the Right People Do the Right Things
293. Eliminate Waste (improving)
294. Deliver Value
1. Only Releasable Code has Value
 2. Deliver Business Results
 3. Deliver Frequently
295. They say that it has the Quality without a Name (QWAN)
—an ineffable sense of rightness in the design.
296. Talk with end user involvement
297. Building dam, building bridge there is no short cut.
298. For software Agile is possible because of rapid change in business conditions, market conditions and needs to be integrated.
299. We can't call a requirement analysis phase complete in a Agile project. It keeps on going continuously.
300. Delivery software quickly and incremental manner.
301. In Agile projects, user feels software early in the life cycle and opportunity to get view quickly.
302. Iterative process

1. Specify increment
 2. Build increment
 3. Validate increment
 4. Integrate increment
 5. Test system and continue
303. Advantage Agile
1. Accelerated delivery
 2. User engaged with system continuously
304. Disadvantage Agile
1. Tracing the increment to meet the overall goal.
 2. Hard to maintain if there are no documents.
 3. There is no specification and so validation problem may exist.
 4. Complexity keep on increasing.
305. Prototyping
1. Does not focus on delivery to the customer.
 2. Used to get feedback from the user.
 3. Business logic may not be coded in prototyping.
 4. Experimental systems are developed using this.
 5. Initial version of the system.
 6. It is used in UI, design, testing.
306. Benefits of Prototyping
1. Improve system usability
 2. Closer match to user needs
 3. Design quality improves (try out choices)
 4. Improve maintainability
 5. Reduce development efforts
307. Process of Prototype
1. Define objective of Prototype
 2. Define functionality prototype
 3. Develop prototype
 4. Evaluate prototype
308. Both (Agile, Prototypes) starts from requirements.
309. E-commerce system developed incrementally.
310. Examples for principles of Agile methods
1. Customer involvement

2. Incremental delivery
 3. People not process
 4. System is designed for change
311. XP is a set of guiding principles.
312. Core values of XP
1. Communication
 2. Simplicity
 3. Feedback
 4. Courage
313. The Scrum master is responsible for teaching others how to use the Scrum process to deal with every new complexity encountered during a project.
314. Much of our society is based on processes that work only because their degree of imprecision is acceptable.
315. Laying out a process that repeatedly will produce acceptable quality output is called defined process control.
316. When defined process control cannot be achieved because of the complexity of the intermediate activities, something called empirical process control has to be employed.
317. There are three legs that hold up every implementation of empirical process control: visibility, inspection and adaptation.
318. Visibility means that those aspects of the process that affect the outcome must be visible to those controlling the process.
319. Enumeration of complexity in software development to the three most significant dimensions: requirements, technology and people.
320. Complexity assessment graph—the vertical axis traces requirements complexity, and the horizontal axis traces technology complexity.
321. Scrum cycle repeats until the project is no longer funded.
322. There are only three Scrum roles: the product owner, the team, and the Scrum master.
323. The product owner is responsible for representing the interests of everyone with a stake in the project and its

resulting system.

324. The Scrum master is responsible for the Scrum process.
Teach the product owner how to maximize ROI and meet his or her objectives through the Scrum.
325. Team members are responsible for figuring out how to turn the product backlog into an increment of functionality within an iteration and managing their own work to do so.
326. Pigs are those who are committed to a project: (PO, Team, Scrum master)
327. Chickens are the spectators.
328. Chickens have no direct authority over the project's execution or progress.
329. Each sprint is initiated with a sprint planning meeting, where the product owner and team get together to collaborate about what will be done for the next sprint.
330. Sprint planning is an 8 hours meeting
331. Sprint review is a 4 hours meeting
332. Sprint retrospective is a 3 hours meeting
333. Sprint review is an informal meeting
334. In the product backlog items, the first four columns are the product backlog item name, the initial estimate, the complexity factor and the adjusted estimate.
335. In the sprint backlog, each task takes roughly 4-hours to finish.
336. This requires that the increment consist of thoroughly tested, well-structured, and well-written code that has been built into an executable and that the user operation of the functionality is documented, either in help files or in user documentation. This is the definition of a 'done' increment.
337. While the traditional project manager is responsible for defining and managing the work, the Scrum master is responsible for managing the Scrum process. To put it simply, Scrum masters make Scrum work.
338. One of the ingredients of Scrum is a practice known as sashimi. Sashimi is a Japanese delicacy consisting of thin

- slices of raw fish.
- 339. The minimum plan necessary to start a Scrum project consists of a vision and a product backlog.
 - 340. The daily Scrum is open to everyone.
 - 341. Scrum addresses the various KPAs in level 2 and level 3. 6 level 2 KPAs 7 level 3 KPAs.
 - 342. The Scrum planning process sets stakeholders' expectations.
 - 343. Specific structure of retrospective meeting
 1. Set the stage—working agreement (ground rules)
 2. Gather data—gathering data creates a shared picture of what happened.
 3. Generate insights—time to ask ‘Why?’
 4. Decide what to do—pick the topic items
 5. Close the retrospective—decide
 - 344. Ensure each stage has minimum of 2 activities.
 - 345. Activities to set the stage
 1. Check-in: Help people articulate what they want from the retrospective
 2. Activity: Focus On/Focus Off
 3. Activity: ESVP (Explorer, Shopper, Vacationer or Prisoner)
 - 346. Explorers are eager to discover new ideas and insights. They want to learn everything they can about the iteration/release/project.
 - 347. Shoppers will look over all the available information, and will be happy to go home with one useful new idea.
 - 348. Vacationers aren't interested in the work of the retrospective, but are happy to be away from the daily grind. They may pay attention some of the time, but they are mostly glad to be out of the office.
 - 349. Prisoners feel that they've been forced to attend and would rather be doing something else.
 - 350. Activities to gather data
 1. Activity: Timeline

Color coding feelings to gather both facts and feelings, use colours to represent emotional states. For example:

1. Blue = sad, mad, bad
2. Red = challenged, stalled
3. Green = satisfied, successful, energetic
4. Yellow = cautious, confused
5. Purple = fun, surprise, humour
6. Salmon = fatigued, stressed

351. Activity: Triple Nickels. (Generate ideas for actions or recommendations.)

352. Activities to Generate Insights:

1. Activity: Brainstorming/Filtering
2. Activity: Force Field Analysis
3. Activity: Five Whys
4. Activity: Fishbone
5. Activity: Patterns and Shifts
6. Activity: Prioritize with Dots
7. Activity: Report Out with Synthesis
8. Activity: Identify Themes
9. Activity: Learning Ideas (what is significant in their ideas)

353. Activities to decide what to do:

1. Activity: Retrospective Planning Game
2. Activity: SMART Goals (Specific, Measureable, Attainable, Relevant, Timely)
3. Activity: Circle of Questions
4. Activity: Short Subjects

354. Activities to Close the retrospectives:

1. Activity: +/Delta
2. Activity: Appreciations
3. Activity: Temperature Reading
4. Activity: Helped, Hindered, Hypothesis (HHH)
5. Activity: Return on Time Invested (ROTI)

355. Determine Goal for Retrospective, for example, Find ways to improve our practices. Determine duration: depends on

- length of the iteration, complexity of the work, size of the iteration.
- 356. Encourage Equal Participation, Focus the Conversation
Encourage New Perspectives
 - 357. J. M. Keller, an expert in motivation and learning, developed a criteria for evaluating instructional designs. The criteria are attention, relevance, confidence/competence, satisfaction—ARCS for short.
 - 358. Retrospectives can't solve every problem.
 - 359. Iteration retrospectives focus solely on the team.
 - 360. For a release retrospective consider inviting folks who represent admin support, on-site customers, product owners, the deployment team, the testing group, marketing, technical support, help desk, operations, beta testers and the project manager.
 - 361. Four transitional phases in a change
 1. Loss
 2. Chaos
 3. Transforming idea
 4. Practice and integration
 - 362. A debrief helps your team examine their experience and extract insights. They'll make conscious connections and form new ideas. Debriefing each activity builds towards the insights and decisions of the retrospective.
 - 363. Agility is the ability to both create and respond to change in order to profit in a turbulent business environment.
 - 364. Agility is the ability to balance flexibility and stability (Highsmith 2002).
 - 365. There are two primary sources for Agile values, the declaration of interdependence authored by the founding members of the Agile Project Leadership Network (www.apln.org) and the Manifesto for Agile Software Development authored by many of the founding members of the Agile Alliance (www.agilealliance.org).
 - 366. The Declaration of Interdependence was developed with project leaders in mind, whereas the Agile Manifesto (the

oft-used short name) was developed with software development in mind.

367. The Agile triangle. The measures here are value (to the customer), quality (required to deliver continuous value to the customer) and constraints (scope, schedule, and cost).
368. The envision phase results in a well-articulated business or product vision—enough to keep the next phases bounded.
369. In the speculate phase, the team hypothesizes about the specifications of the product and builds a release plan, knowing that as the project continues both technology and customer specifications will evolve as new knowledge is gained.
370. The explore phase then becomes an iterative operation in which the features and stories are implemented.
371. In the adapt phase, the results of these experiments are subjected to technical, customer and business case review, and adaptive actions are incorporated into the next iteration.
372. Project leaders need to focus on value in several ways: value determination (with product owners), value prioritization (backlog management) and value creation (iterative development).
373. Levels of Listening
 1. Level 1- Internal Listening—Hearing the word and interrupt it in own language
 2. Level 2- Focused Listening—Listening and responding without interrupting in own language
 3. Level 3- Global Listening—While focused understand the team global environment
374. Agile projects gives high IRR comparatively with other traditional projects.
375. Payback Period: Payback period is the unit of time (amount of time) to receive (get back) the initial amount invested. It does not consider the future.

376. Net Present Value (NPV): It is the accurate value of all the future inflow due to business in today's value (Present Value). We need to convert the future value into the present value and then sum of all the present value gives NPV (Net Present Value).
377. Discounted ROI = NPV of Benefits / NPV of Costs
378. ROI = Velocity × Margin
379. The most effective and efficient way according to the Agile manifesto to convey the information throughout the team is to use face-to-face communication.
380. Collaboration involves working together to jointly produce a deliverable or make a decision, whereas communication involves the sharing of information.
381. Time and Material Contract: This is the most suitable method of contracting for Agile projects. In this type of contract, the vendor is being paid for hours spent by the team members.
382. Agile concentrates straightaway into execution (action) part.
383. Iterative nature takes care of identifying risks earlier. Risk process repeats for every iteration.
384. Risk analysis is used to help a team understand uncertainty that could affect the outcome of the project.
385. Four Risk Planning mechanism:
1. Mitigate
 2. Avoid
 3. Evade
 4. Contain
386. Risk management should be an integral part of organizational processes—When you are doing Agile, you are doing risk management, and hence it is a part of the standard process.
387. Risk management should take into account human factors —That's why risk management is done with all stakeholders, and not single-handedly by a project manager.

- 388. eXtreme projects are chaotic, messy and unpredictable; speed and innovation are critical, and planning is chaotic and just-in-time.
- 389. Innovation is critical in eXtreme projects. In fact, it is more than critical; innovation is what eXtreme projects are all about.
- 390. Traditional project management is past-oriented. eXtreme project management is future-oriented.
- 391. The goal of traditional project management is to get it right the first time; the goal of Agile or adaptive project management is to get it right the last time.
- 392. With a task-based WBS, tasks are relatively permanent. With a feature-based WBS, features can be added or deleted during each iteration.
- 393. Coding standards focus on consistency and consensus over perfection.
- 394. You can still cancel the project if you encounter risks and issues in planning phase of the project.
- 395. Feature shell is also the term used to refer feature card.
- 396. There is no formula to define the size of the story.
- 397. The amount of time a user story will take to develop can be more easily estimated in ideal days than elapsed days.
- 398. Ideal days are easier to explain to outside team than user stories.
- 399. A project is usually critical if the company or customer can't survive without it.
- 400. When you create a trade-off matrix, only one item can be the number one priority.
- 401. Prioritizing features lets you stop a project before it is complete and still delivers the critical features.
- 402. When we create feature cards we record acceptance test too.

Index

100 point method, 337
3 C's, 118
40-hour work week, 64
5 why technique, 254–255
5S tool, 247
7S framework, 330

A

ABO, 43
Acceptance testing, 229
Active listening, 83
Activity cost estimate, 175
Actual cost (AC), 195
Adaptive leadership, 275
Adaptive planning, 5, 12, 116
Adaptive software development (ASD), 32
Affinity diagram, 253
Affinity estimation, 165
Agile, 26
 analysis and design, 197–214
 compliance, 343
 contracting methods, 131
 data method, 31
 framework, 1
 hybrid models, 69
 knowledge and skills, 8, 15

manifesto, 27
methodologies, 31–32, 55–70
methods, 27
metrics and estimations, 161–185
modeling, 31, 210
planning, 43
principles and mindset, 2, 9, 25–44
software development, 27
team space, 87
team stages, 291
tooling, 93
traditional ...

Editor—Acquisitions: Neha Goomer

Senior Editor—Production: C. Purushothaman

Copyright © 2017 Pearson India Education Services Pvt. Ltd

This book is sold subject to the condition that it shall not, by way of trade or otherwise, be lent, resold, hired out, or otherwise circulated without the publisher's prior written consent in any form of binding or cover other than that in which it is published and without a similar condition including this condition being imposed on the subsequent purchaser and without limiting the rights under copyright reserved above, no part of this publication may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording or otherwise), without the prior written permission of both the copyright owner and the publisher of this book.

ISBN 978-93-325-7394-9

e-ISBN 9789353067083

First Impression

Published by Pearson India Education Services Pvt. Ltd, CIN: U72200TN2005PTC057128, formerly known as TutorVista Global Pvt. Ltd, licensee of Pearson Education in South Asia.

Head Office: A-8(A), 7th Floor, Knowledge Boulevard, Sector 62, Noida 201 309, Uttar Pradesh, India.

Registered Office: 4th floor, Software Block, Elnet Software City,

TS-140, Block 2 & 9, Rajiv Gandhi Salai, Taramani, Chennai 600 113, Tamil Nadu, India.

Fax: 080-30461003, Phone: 080-30461060

www.pearson.co.in, Email: companysecretary.india@pearson.com

Composer: SRS Technologies, Puducherry

Printed in India