

Agile Planning and Estimation : From Guesswork to Predictable Delivery

In today's **VUCA world**, requirements evolve rapidly due to shifting business priorities, competitive pressures, and the constant need to reduce time-to-market.

Agile Project Management embraces this reality through principles such as **welcoming change, even late in development** and **continuous delivery of value to customers**.

However, **poor planning and unreliable estimation** can quickly erode the benefits of Agile, leading to delivery risks and loss of stakeholder confidence.

As an Agile Project Manager, ensuring continuous planning, effective risk management, and realistic forecasting (both effort and financials, especially in T&M engagements) enables predictable, real-time delivery instead of guesswork.

Why is Agile different from Traditional (Water fall) Project management?

Traditional/Predictive way of projects has an (a) upfront plan (b) clear scope and objectives (c) structured phase gate reviews & (d) well known outcome.

Predictive project management works best when requirements are stable, change is minimal, stakeholders expect fixed outcomes, and risk appetite is low.

Traditional Vs Agile

S.No	Traditional	Agile
1	Fixed Scope	Evolving Scope
2	Detailed up-front Plan	Progressive Plan
3	Dates are Committed	Dates are Forecast
4	Change is a Risk	Change is expected
5	PM owns the Plan	Team owns the plan

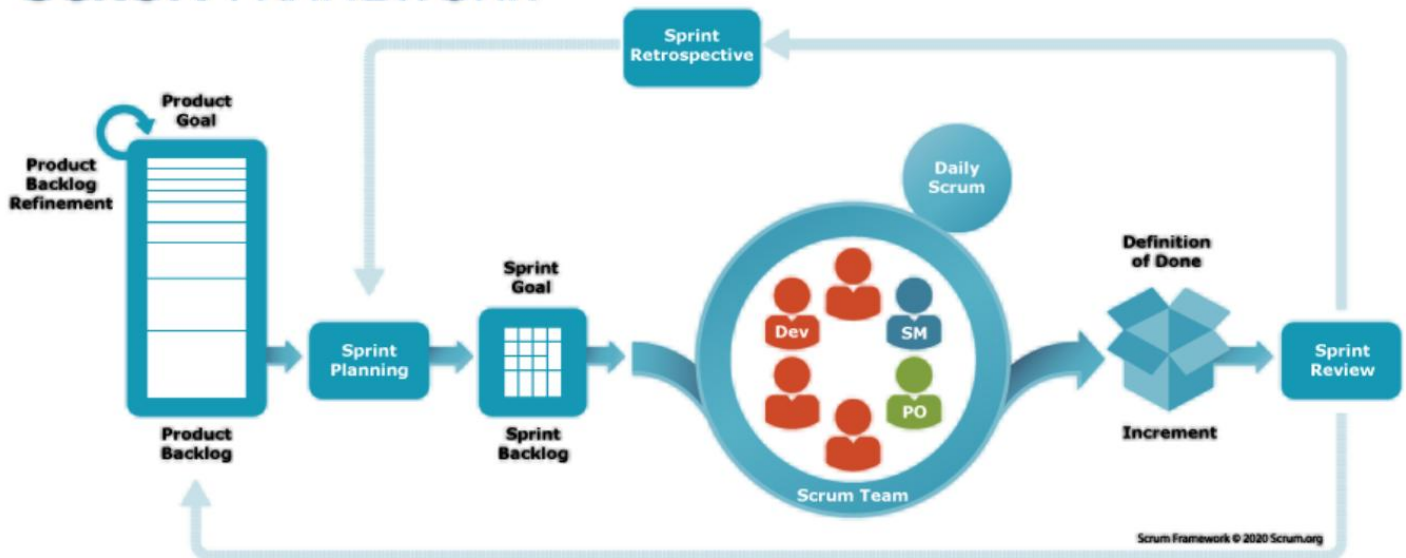
Agile Planning – Different Levels

Agile planning is not a single event but a **continuous activity across multiple horizons**, allowing teams to adapt plans as learning increases.

S.No	Types Planning	Description
1	Product Vision and Roadmap	Defines product future state
2	Release Planning	Features and Capabilities expected
3	Sprint Planning	Team selects what they can complete within selected timeline
4	Daily Planning (Stand-up's)	Short, focused, adopting to plan
5	Sprint Retrospective	Sprint review and continual learning

It is very important to understand the Scrum (Agile) Framework and various phases,

SCRUM FRAMEWORK



The Scrum framework provides the structural foundation for Agile delivery, defining clear roles, events, and artefacts that support transparency, inspection, and adaptation.

Role of Agile Project Manager

Rather than a traditional Command-and-Control Manager, the Agile Project Manager acts as an **integrator and enabler**, ensuring impediments are identified and removed, risks are managed, and teams can operate at full capacity through transparency and continuous feedback.

Who Own the Planning in Agile?

When it comes to ownership of planning, Product Owner, Scrum Master, Development Team and Project Manager share equal responsibilities across.

Agile Project manager doesn't assign work (Scrum Master does), but adds a significant value towards Forecasting, Budgeting, Stakeholder Management, Data Security and Privacy, validating and aiding to remove impediments, blockers in the Project.

Agile Estimation:

One of the biggest mindset shift from Predictive/Traditional project management to Adaptive/Agile Project Management is understanding 'What Estimation really means?'

Agile methodology does not need any kind of Estimation – Misconception

- ✓ Businesses still need Forecasting
- ✓ Budgets still need planning
- ✓ Stakeholders need visibility

In Agile, **estimation is not a commitment or a mathematical calculation—it is a conversation** that builds shared understanding and supports forecasting.

Rather than asking 'How long will this take', Agile ask 'How complex is this compared to other work'

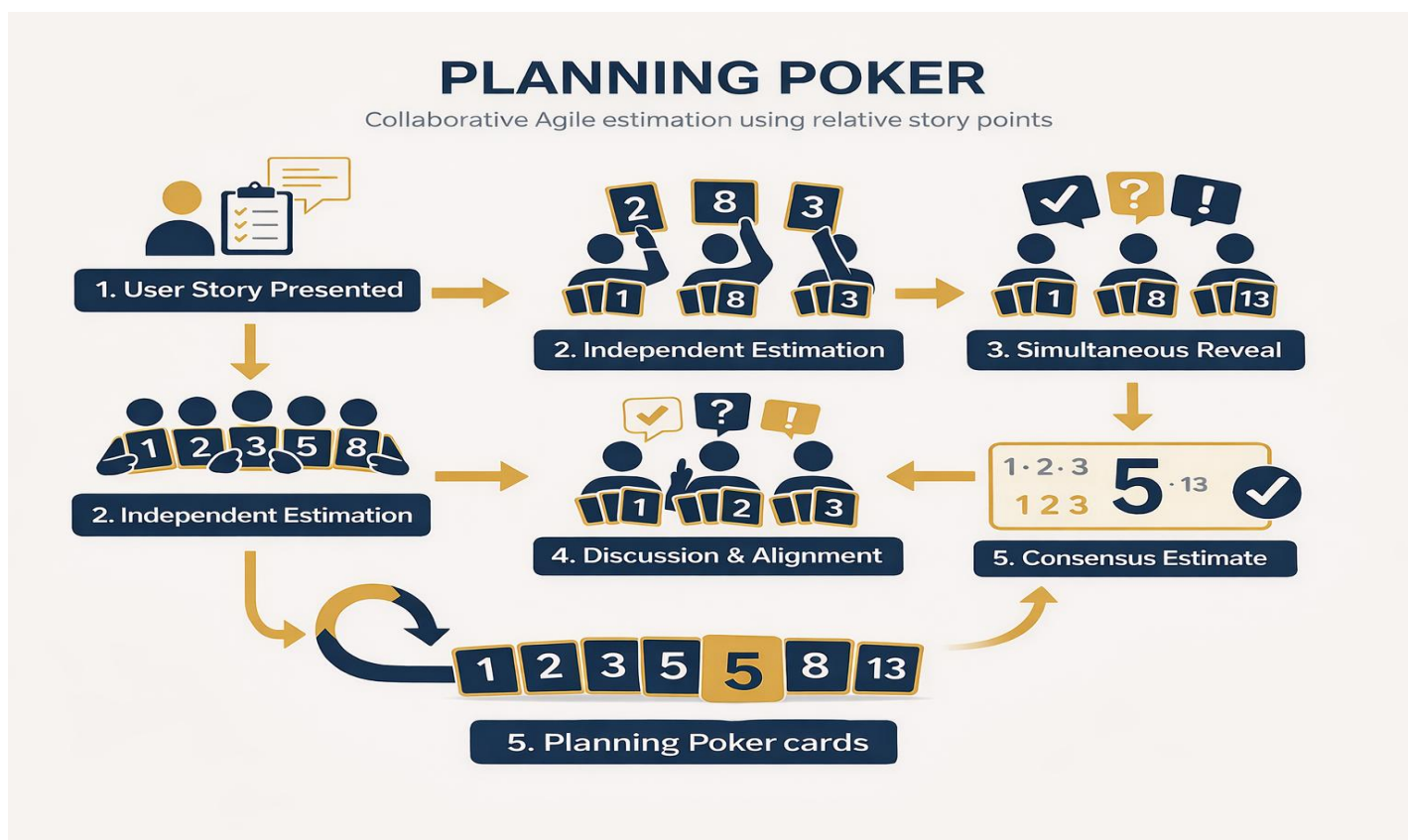
It is important that Agile PM understands the following concepts and starts the estimation

S.no	Concept	Description
1	Complexity	A measure of how difficult, uncertain, or risky a piece of work is compared to other work.
2	Velocity	The average amount of work a team completes in a sprint
3	Capacity	The actual amount of work a team can take on in a sprint based on availability
4	Backlog	An ordered list of all work items required to deliver a product, prioritised by business value
5	User Story	A short, value-focused description of functionality
6	Backlog Refinement (Grooming)	A continuous activity where upcoming backlog items are clarified, split, and estimated
7	Acceptance Criteria /DOD	Mutual agreement on what “Complete” means to ensure consistent quality
8	Story point	A relative unit used to estimate effort, complexity, and uncertainty—rather than time.

What is that one method which is very commonly accepted?

The most widely accepted Agile estimation technique is Planning Poker, especially in sprint-based development environments.

** - Assumption made – Project uses Sprint based development method, planning poker works best during Sprint level Estimation



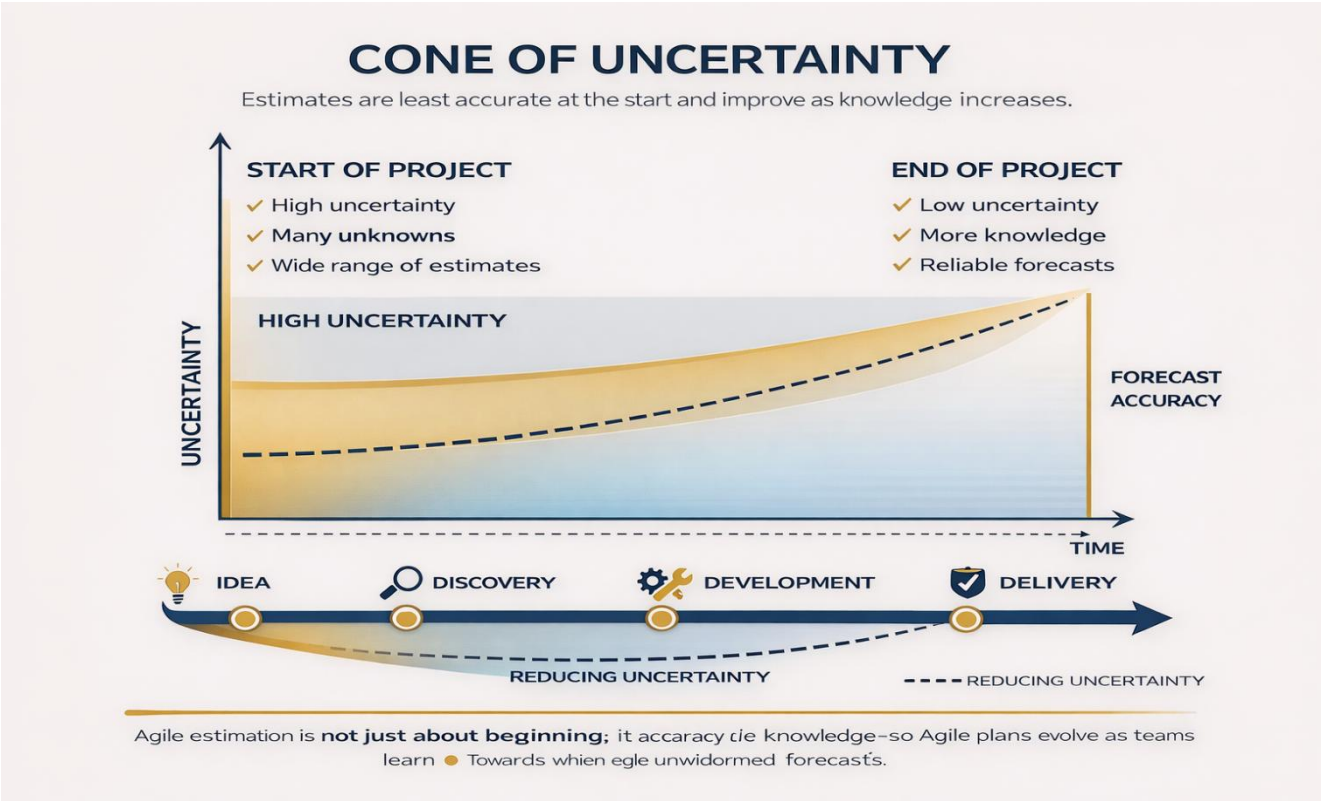
Common Mistakes to avoid during Estimation

- 1) Treating Story points as hours, rather assume them as relative complexity and uncertainty
- 2) Estimates are forecasts and not commitments
- 3) Respect independent estimates and consciously set aside hierarchy, bias, and ego.
- 4) Always engage in conversations to resolve
- 5) Avoid re-negotiation during an active sprint

The cone of Uncertainty:

The **Cone of Uncertainty** illustrates that estimates are least accurate at the start of a project and improve as knowledge increases.

Early estimates are guesses; later estimates are informed forecasts. This is normal—even for experienced Agile teams. As sprints progress, uncertainty reduces and predictability improves.



Key Takeaways

- ❖ Uncertainty is **highest at the beginning**
- ❖ Accuracy **improves with learning**
- ❖ Agile planning succeeds by **re-forecasting**, not locking dates
- ❖ Transparency builds **stakeholder trust**.

Predictability in Agile is achieved not through rigid plans, but through transparency and continuous learning.

Case Study:

Talking more about Transparency build Stakeholder trust, I have a practical example in one of my Agile project.

The Customer was moving from Predictive method to Adaptive (Agile) method. Client POC was not comfortable with we building the application in small increments, he felt insecure about we finishing the project on time.

In order to tackle this problem, we built a simple excel sheet which was as following,

#	Sprint	Start Date	End Date	Features	Baseline	Forecast
1	Sprint x	dd-mm-yy	dd-mm-yy	<ul style="list-style-type: none">Mention the list features considered for the Sprint	<ul style="list-style-type: none">List down the User stories for the sprint as baselineYou can consider the extract from Azure Devops or JIRA	While giving a demo at the end of the Sprint, list the user stories completed and what features are available for review/testing

We were tracking the Features vs User stories for the sprint and ensured a transparent communication to manage stakeholders expectations.

Over the course of multiple sprints, this transparency helped the client gain confidence in incremental delivery. Stakeholder anxiety reduced significantly, sprint reviews became more focused, and decision-making shifted from opinion-driven to data-driven.