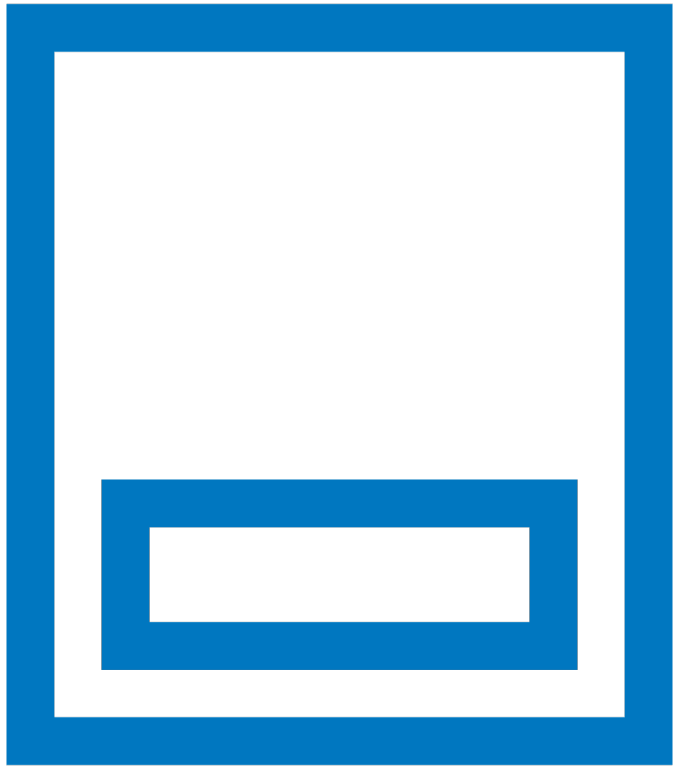


Story Point Analysis

DEVELOPMENT VIEW

SAP ABAP Development

- **Story Point Effort:** A range of 3-8 developer-days per story point is plausible.
- Factors pushing to the higher end include:
  - **Legacy Code:** Entrenched, poorly documented codebases slow down everything.
  - **Complex Integrations:** Interfacing with non-SAP systems adds overhead.
  - **Customization:** Heavy customization vs. standard SAP functionality takes more time.
- **Illustrative Examples:**
  - "A story point to create a simple ALV report on standard SAP data might take 3-4 developer-days."
  - "A story point to develop a custom interface to a third-party system with intricate data mapping could reach 6-8 developer-days.

.NET Full Stack Development

- **Story Point Effort:** A wide range of 2-6 developer-days per story point.
- The frontend/backend split is key:
  - **Frontend-Heavy:** Complex UIs, rich interactions push this higher.
  - **Backend-Heavy:** Intense data processing, complex logic add effort.
- **Illustrative Examples:**
  - "A story point for a basic CRUD screen with minimal UI complexity might take 2-3 developer-days."
  - "A story point involving a sophisticated search feature with backend algorithms and a dynamic UI could take 5-6 developer-days."

C++ Programming Language

- Story Point Effort: Ranges from 3-8 developer-days per story point, influenced by:
  - Code Quality: Clean, modern codebases are faster to work with.
  - Domain: Scientific computing vs. game logic have different complexity profiles.
  - Performance Focus: Optimization needs add significant effort.
- Illustrative Examples:
  - "A story point to refactor a well-contained module in a legacy C++ project might take 3-5 developer-days."
  - "A story point to implement a new, performance-critical algorithm with strict memory constraints could take 6-8 developer-days."

MySQL

- Story Point Effort: Tends to be on the lower end at 1-4 developer-days per story point. Exceptions include:
  - Query Optimization: Tackling performance issues on large datasets gets complex.
  - Scaling: Design for massive scale or complex replication adds effort.
- Illustrative Examples:
  - "A story point to write simple CRUD queries for a well-designed table might take 1-2 developer-days."
  - "A story point focused on optimizing poorly performing queries on a large dataset could take 3-4 developer-days."

Snowflake Data Warehouse

- **Story Point Effort:** A tentative range of 1-5 developer-days per story point.
- Factors influencing this massively include:
  - **Data modelling complexity:** Highly normalized schemas take more effort.
  - **Query Sophistication:** Simple reads are low-effort, intricate analytics are higher.
  - **Transformation Needs:** If the workflow involves complex ETL within Snowflake? This adds time.
- **Illustrative Examples:**
  - "A story point to build a well-structured reporting dashboard on pre-modelled data might take 1-2 developer-days."
  - "A story point to create a new data mart with multiple transformations and optimized queries could take 3-5 developer-days."

Talend ETL

- Story Point Effort: Range of 2-6 developer-days per story point is plausible, influenced by:
  - Source/Target Complexity: Numerous systems, messy data sources increase effort
  - Transformation Logic: Are you doing simple mappings or complex business rules?
  - Volume: Large datasets naturally add some overhead even with efficient tools.
- Illustrative Examples:
  - "A story point involving a simple extract and load from a few structured sources might take 2-3 developer-days."
  - "A story point requiring complex data cleansing, deduplication, and loading into a highly normalized target could take 4-6 developer-days."

AWS

- Story Point Effort: This is very difficult to generalize due to the sheer range of AWS services used.
  - Infrastructure as Code (IaC): Mature IaC setups allow faster provisioning (lower effort).
  - Serverless vs. Traditional: Serverless *can* mean less effort per story point, but not always.
  - Service Complexity: Basic S3 storage is simpler than setting up an elaborate data pipeline.
- Illustrative Examples (With Caveats):
  - "A story point involving deploying a pre-configured EC2 instance using existing IaC templates might take 1-2 developer-days."
  - "A story point to design and implement a new serverless function with integrations might take 3-5 developer-days."

Mulesoft

- Story Point Effort: A potential range of 2-5 developer-days per story point affected by:
  - Integration Count: Each new system you connect to adds effort.
  - Complexity: Basic pass-through integrations are lower effort than intricate transformations.
  - Custom Code: Heavy reliance on custom Java within Mulesoft increases effort

Outsystems

- **Story Point Effort:** Low-code promises *potential* for 1-3 developer-days per story point, Caveats:
  - **Suitable Use Cases:** Outsystems excels in specific scenarios; complex logic still takes time.
  - **Customization Needs:** Heavy customization negates some of the low-code speed.
  - **Team Maturity:** Teams well-versed in the platform are far more efficient.

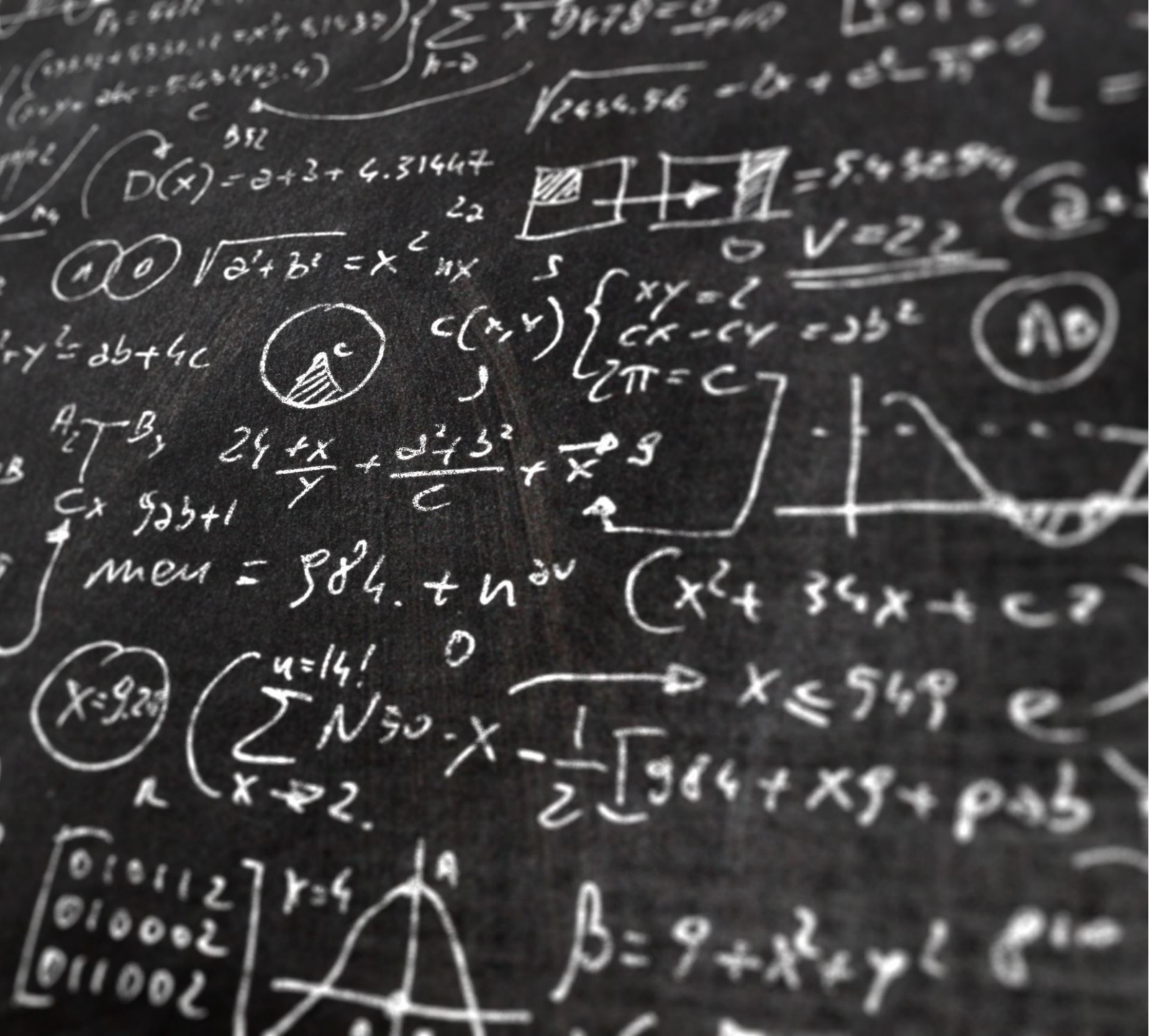
Sample : Summarized view for Questions mentioned in email for complete survey.

SAP ABAP Development

- Story Point Effort: "SAP landscapes will be an average range of 4-7 developer days per story point. Projects involving extensive customization or integration with non-SAP systems can skew toward the higher end."
- Y-o-Y Productivity: "While highly context-dependent, focusing on ABAP modernization initiatives have seen up to a %X improvement in story point velocity year-over-year."
- FTE per Pod: "ABAP pods often trend smaller (X-Y FTEs) due to specialized skillsets and the need to tightly integrate with business domain experts."
- Shoring & Pyramids: "We are increasingly offshoring routine ABAP maintenance tasks, while retaining complex development or architecture expertise onshore."

Appendix

- Reference – Microsoft Web Archives



Welcome to the Story Point Estimation

AN OVERVIEW OF PROJECT ESTIMATION TECHNIQUES

Presentation Overview

What is Story Point Estimation?

Story point estimation is a process of estimating the effort required to complete a task or a user story. It is important because it helps teams to plan and prioritize work more effectively.

Estimation Techniques

There are several techniques that can be used for estimating story points, including planning poker, affinity mapping, and t-shirt sizing. We will explore the pros and cons of each technique and how to use them effectively.

Benefits of Story Point Estimation

Story point estimation can help teams to improve project planning by providing a more accurate estimate of how long a project will take to complete. It can also help to identify potential issues and risks early on, allowing teams to address them before they become bigger problems.

What is Story Point Estimation?

Story Point Estimation Technique

Story point estimation is a technique used in agile project management to estimate the effort required to complete a user story. It is a relative sizing technique that is based on the complexity of the story and the effort required to complete it.

Benefits of Story Point Estimation

Story point estimation helps project teams to better understand the scope of work and how long it will take to complete. It also promotes team collaboration and helps to prioritize tasks based on their level of complexity.

Why use Story Point Estimation?

Bias Reduction and Increased Accuracy

Story point estimation is based on the complexity of the story rather than the time required to complete it. This helps to reduce bias and increase accuracy by avoiding the human tendency to underestimate or overestimate time requirements.

Identifying Risks and Dependencies

Story points can help identify risks and dependencies early in the project by highlighting complex stories that may require additional resources or attention, allowing for early mitigation of potential issues.

Improving Planning and Project Progress

Story points can help improve planning and project progress by providing a clear understanding of the scope and complexity of each story, allowing for better allocation of resources and more accurate forecasting of project timelines.

How to Estimate Story Points

Planning Poker

Planning poker is a technique used to estimate the effort required to complete a user story. It involves team members making individual estimates and then discussing the reasons behind their estimates until a consensus is reached.

Affinity Mapping

Affinity mapping is a technique used to estimate the effort required to complete a group of related user stories. It involves team members grouping user stories into categories based on their complexity and then using these categories to estimate the overall effort required.

T-Shirt Sizing

T-shirt sizing is a technique used to estimate the relative size of user stories. It involves team members assigning each user story a size based on its complexity, with sizes represented by t-shirt sizes such as small, medium, large, and extra-large.

Planning Poker

Planning poker is a collaborative estimation technique that involves the entire team. Each team member is given a set of cards with numbers representing story points.

Affinity Mapping

Affinity mapping is a technique that helps teams to group stories based on their size and complexity, allowing for a better understanding of the work that needs to be done.

T-Shirt Sizing

Simplifying Complexity

T-shirt sizing simplifies the estimation process by assigning a size to each story based on its complexity, making it easier for teams to quickly estimate the time and effort required to complete a project.

Quick Estimation

T-shirt sizing is a quick technique for estimating the size of stories in a project, making it ideal for projects with a large number of stories that need to be estimated quickly.

Using Story Points for Project Planning

Story points are a useful tool for estimating the time required to complete the project and to identify risks and dependencies. They help teams to plan and prioritize tasks effectively and to deliver high-quality results.

Estimating Project Velocity

Project velocity is the rate at which the team completes stories. It is a key metric that can be used to estimate the timeline for the project and determine how many stories can be completed in a given time period.

Identifying Risks and Dependencies

Story points can be an effective tool for identifying risks and dependencies early in the project. By analyzing the story points, a team can determine which stories may represent a higher risk or a dependency that needs to be addressed before the story can be completed.

Conclusion

Story point estimation is a valuable technique that helps teams to estimate the effort required to complete a project accurately, reduce bias, and identify risks and dependencies early in the project. Incorporating story point estimation into project planning process helps teams to improve project outcomes and deliver high-quality products.