Team Process Data Warehouse – Goals and High-Level Requirements

Background

- TSP[SM] is used by teams working in a wide variety of problem domains (e.g. software, hardware, services). Since these activities are not limited to software, the name "Team Process Integrated" and the acronym "TPI" are used in these documents to describe the full range of TSP-inspired high-maturity processes, and to avoid improper use of Carnegie Mellon service marks.
- TPI teams use a variety of project execution methodologies (e.g. waterfall, Scrum, Agile, etc.)
- TPI teams collect personal, team, group and organization data spanning projects, deliverables, phases, releases, org changes, etc.
- Data is rich and fine-grained
- Up to now, data analysis activities have mostly been confined to single team projects. This represents a significant lost opportunity

Goals

- This initiative will endeavor to create a cohesive set of database and software **components** that any organization can install and set up to create a **private, local repository** of TSP/TPI metrics
    - This initiative **will not** endeavor to install specific physical repositories. In particular, this initiative will **not** endeavor to create a "global database" of TSP/TPI metrics.
    - This initiative will only endeavor to create a set of reusable software components. It will be up to particular organizations to install these components on computers within their network, for the purpose of performing private analyses of their own data.
- Maintain an open architecture, embracing extensibility and customizability
- Aggregate data from different sources, including different TSP/TPI tools
- Analyze data from many projects across an organization, for example to:
    - measure ROI of process improvement activities
    - measure historical performance for use in planning activities
- Maintain history of changes to data, and view trends across time
- Generate charts and reports using third-party reporting tools
- Feed data into other systems for example to facilitate status reporting

[SM] Team Software Process and TSP are service marks of Carnegie Mellon University. Carnegie Mellon University has neither contributed to nor evaluated the contents of this document.

Data Flow

- The warehouse should be capable of aggregating data from multiple sources, to include:
    - Multiple different TSP/TPI tools
    - Other enterprise data sources, such as corporate directories or ALM systems
- In keeping with industry best practices, the data warehouse will **not** be designed to provide back-end transactional storage for TSP/TPI tools or other source systems. Those tools/systems will still need their own, independent transactional data stores. Instead, ETL processes will Extract data from source systems, Transform the data as necessary, and Load it into the data warehouse.
    - To the extent possible, the warehouse should anticipate/allow ETL processes that operate continuously, loading data in near-real-time.
- Data in the warehouse is made available on a read-only basis for analysis and reporting purposes. External logic should not expect to be able to alter values in the warehouse and see those changes propagate back into the source systems (e.g. the originating TSP/TPI tools)

Time History

- The warehouse shall retain a history of changes to certain data, making it possible to
    - View the data as it appeared at some historical point in time
    - Determine the changes that have occurred between two points in time
    - View trends in the way data has changed over time

Query Goals

- At a minimum, the warehouse should include enough data to perform all of the following "traditional TSP" analyses:
    - Planned-vs-actual time by process phase
    - Planned-vs-actual defects injected and removed by process phase
    - Planned-vs-actual size
    - Time/size estimating error
    - Planned-vs-actual Productivity, defect density, review rate
    - Planned-vs-actual Phase yield, process yield
    - Planned-vs-actual AF/R, DRL, PQI, COQ
- These analyses should be possible for various subsets of data, including:
    - The entire organization
    - A single sub-organization or group of organizations
    - A particular project, subproject, or group of projects
    - A particular subteam and/or iteration within a project
    - A particular component or subcomponent within a project
    - A particular milestone within a project
    - A set of data matching user-defined labels

Data Privacy

- When querying data, it shall be possible to determine the number of individuals, teams, projects, and organizations whose data contributed to a particular report.
- The warehouse will NOT provide facilities for analyzing the data of a particular individual. (This restriction is intentional, to discourage improper use of personal metrics.)
- Reporting mechanisms shall include configurable privacy thresholds. For example, a report could censor itself if a particular drill-down would show private data for a single individual. These thresholds should be configurable on a per-user basis, so coaches could potentially see more granular data than senior managers.

Technical Requirements

- To the extent possible, different components of the warehouse should be designed with portability in mind. For example, database components should ideally support the use of a variety of different relational database platforms, including zero-cost solutions whenever possible.
- It should be possible to backup/restore the data in the warehouse, to guard against data loss or corruption.
- Scalability:
  - The warehouse should be capable of supporting very large organizations, to include data collected by tens of thousands of people over a span of many years
  - The warehouse should not be overly complex to set up for small organizations with simple data analysis needs. For these small organizations, it should be possible to setup and install a local data warehouse repository with minimal effort and expense.

Content Requirements

Organizations

- The warehouse should be capable of capturing data from more than one organization
- Organizations can be recorded in the warehouse as one or more hierarchical org structures
- To the extent possible, the warehouse architecture will not place specific constraints on the definition of an organization:
    - "Organization" could map to various management or reporting entities such as branches, divisions, groups, subsidiaries, companies, or even entire distinct corporations. This decision would be the prerogative of the person who manages a particular data warehouse repository. For example, someone creating a local repository for analysis of corporate data might use "organizations" to represent the corporate reporting structure; while someone creating a warehouse repository to analyze data across the global TSP community might use organizations to represent distinct corporations.
    - The org structure hierarchy can be arbitrarily deep, and can be "ragged" (deeper in some branches of the tree than others).
    - The warehouse will **not** assume that a particular individual belongs to a single organization, or that a particular project team is made up of individuals from a single organization.
- Imported data can be tagged with the originating organization
- It shall be possible to analyze data for a particular organizations and its sub-organizations, for example to:
    - Display aggregated CIO-level data for an entire organization, for example as part of a business intelligence dashboard
    - Drill down to see data for a particular sub-organization
    - Enforce security constraints, for example to restrict a particular user's view of a report to the data from a single organization

Projects

- The warehouse shall store data from many different projects
- A given project may span any number of iterations
- A particular project / iteration may involve individuals from multiple teams and multiple organizations

Teams

- Project work is performed by teams of individuals
- Teams can have subteams (e.g. hardware team, software team, QA)
- Teams and subteams can include members from more than one organization.

Processes

- The warehouse should not presume a specific hard-coded process, but should be amenable to the analysis of data from teams using:
  - Different problem/business domains (e.g. software development, hardware design, production support, documentation, training, services)
  - Different project lifecycles (e.g. agile / Scrum / rapid-iterative / waterfall)
  - Different processes, phases and workflows
  - Different work products and size metrics
- The warehouse should be capable of capturing the processes that various teams are using, and tagging imported data with the appropriate process metadata.
- A mechanism should be provided to map phase types from one process to another, so that:
  - Teams can manage and analyze their data using customized/tailored processes
  - This data can still be mapped to an organizational standard process for analysis and reporting purposes

Taxonomies

- Organizations may want to group data into arbitrary categories and analyze the data by category. For example, an organization might want to:
  - Perform separate analyses of data from new development projects vs. legacy maintenance projects
  - Analyze data separately for web development, rich client application development, or embedded software
  - Analyze the differences between teams that are new to TSP and teams that have been using TSP for many years
  - Analyze the return on investment obtained by putting individuals through a PSP[SM] Advanced course
  - Perform an analysis that excludes data collected by teams or individuals that are known to be willfully ignoring the process
- To support these needs, the warehouse will allow the definition of custom taxonomies. Each taxonomy will have:
  - A name (For example: "Team Maturity")
  - A fixed list of terms for categorizing data (For example: "Learning", "Applying", "Mastering" "Perfecting")
- The warehouse shall allow taxonomy terms to be attached to organizations, projects, teams, individuals, WBS components, or tasks. These attachments can be multivalued.
- Data queries and reports shall support filtering or categorizing data by these taxonomies.

[SM] PSP is a service mark of Carnegie Mellon University.

Metadata

- Organizations may have arbitrary types of custom metadata that they would like to record in the warehouse. For example, when a project component is stored in the warehouse, they might want to annotate it with the unique ID of a defect from a corporate defect tracker. These annotations could be important to enable integration of warehouse data with other systems.
- To meet this need, the warehouse will include a facility to define metadata types
- Free-text metadata values can be attached to organizations, projects, teams, WBS components, or tasks. These attachments can be multivalued.

Work Breakdown Structure

- The Warehouse shall include a description of the hierarchical components and subcomponents that make up each project
- The warehouse shall include a list of the milestones declared for each project including:
    - The name of the milestone
    - The commit date, if applicable
- The Warehouse shall include a list of the tasks present in each project, including:
    - The names of each task
    - The component / subcomponent to which the task belongs
    - The process phase for each task
    - The milestone to which the task belongs

Metrics Data

- The warehouse shall include planned and actual size data for individual components
- The warehouse shall allow free-text notes to be attached to any component or task
- The warehouse shall include raw time log entries, to include:
    - The contributing organization
    - The associated team project, WBS component, and task
    - The process phase
    - Start and end timestamps for the time log entry
    - The number of minutes of delta and interrupt time
- The warehouse shall include defect log entries, to include:
    - The contributing organization
    - The associated team project, WBS component, and task
    - The date and time of the defect log entry
    - The type of the defect (from the defect type standard)
    - The process phases where the defect was injected and removed
    - The number of minutes of fix time
    - The defect description
- The warehouse shall include calculated earned value data for WBS components and tasks, to include:
    - Baseline planned time
    - Baseline completion date

  - Planned and actual time
  - Planned completion date
  - Replanned completion date
  - Forecast completion date
  - Actual completion date
  - Percent complete
  - Percent spent