TSP^{SM} and Architecture in the Real World

James McHale, SEI

(with thanks to Luis Carballo, Bursatec and Robert Nord, SEI)

NDIA CMMI Technology Conference 2011

The Most Pervasive CMMI V1.3 Change?

Certainly the most publicized model changes are at the higher maturity levels.

But consider…

*Requirements Development, Specific Practice 3.2 – Establish and maintain a definition of required functionality **and quality attributes.***

Quality attributes are mentioned dozens of times now throughout the informative material of the model.

Quality Attributes and Architecture

From the Glossary:

> *quality attribute – A property of a product or service by which its quality will be judged by relevant stakeholders. Quality attributes are characterizable by some appropriate measure. Quality attributes are non-functional, such as timeliness, throughput, responsiveness, security, modifiability, reliability, and usability. They have a significant influence on architecture.*
>
> *architecture – The set of structures needed to reason about a product. These structures are comprised of elements, relations among them, and properties of both.*

In the most basic sense, quality attributes, whether expressed or implied, are what drive architectural decisions.

Put another way, architecture decisions express quality attributes, *whether they are stated or not*.

An Opportunity for Architecture

Background:

- Bolsa Mexicana de Valores (BMV) operates the Mexican financial markets under license from the federal government.

- Bursatec is the technology arm of the BMV.

- BMV desired a new trading engine to replace the existing stock market engine and integrate the options and futures markets.

- The BMV performed a build vs. buy analysis, and decided to replace their three existing trading engines with one in-house developed system.
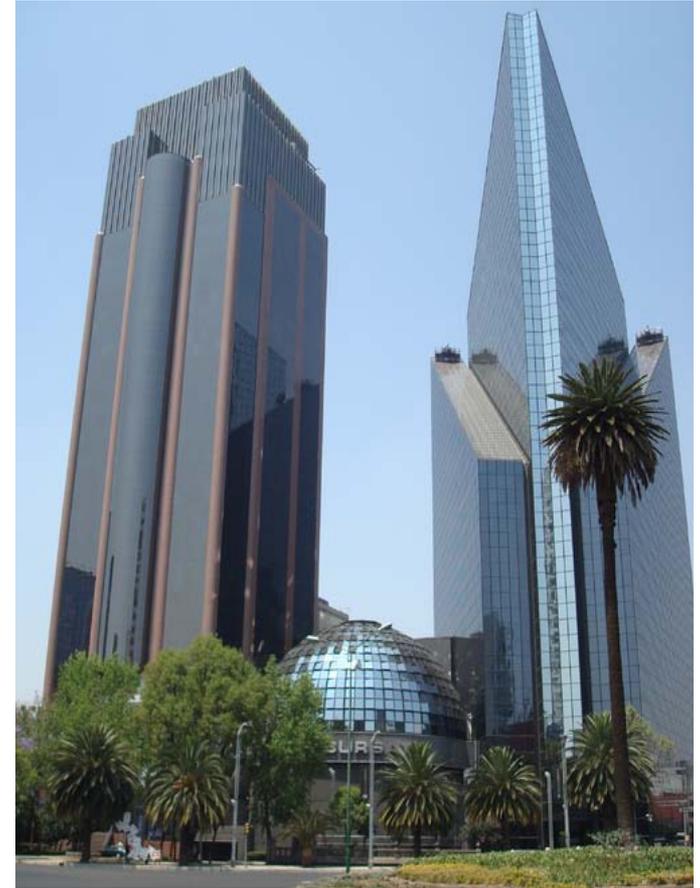
The Project -1

Bursatec committed to deliver a trading engine in 8-10 quarters:

- High performance (as fast or faster than anything out there)

- Reliable and of high quality (the market **cannot** go down)

- Scalable (able to handle both spikes and long-term growth in trading volume)

Bursatec approached the SEI for support during design & development.

SEI's role—provide methods, techniques, and guidance to improve Bursatec's software delivery capability:

- Training and coaching for the system architects

- Training and coaching for the development team
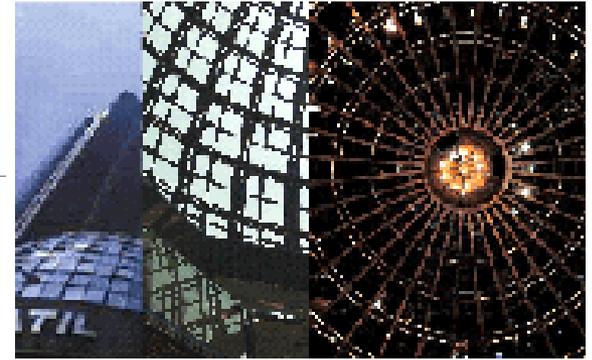
The Project -2

Architecture Decisions (to satisfy quality attributes):

- Development in Java (lower TCO)
- Low Latency Communication Multicast Network
- In memory data storage during trading session.
- Hot-Hot High Availability configuration.
- Parallel processing in JVM
- Horizontal scalability

Functional Requirements:

- Order routing with FIX protocol.
- Interconnect to current legacy systems.
- Combined Cash and Derivatives markets with a single Control Workstation.
- Separate Market Data and Index calculation system.

A Partial List of Potential Problems

Complicating factors:

- Pressure – managers replaced when commitments are not met

- Inexperience - available staff talented but young

- Large project - scope of the project beyond the organization's recent experience

  - # of person-months

  - # KLOC/function points

  - # of interconnecting platforms

  - # of individual projects

- Key implementation technologies never used together formally

- Constant stream of new requirements/changes to business rules

Trading Engine Quality and Other Attributes

Quality Attributes

- Under 1ms processing latency

- Horizontal scalability

- Redundant HA system

- Warm DR system

- Automatic testing framework (one day turnaround attribute)

- Localize business rules changes in specific modules

Other Attributes

- Backward compatible with current systems

- Combined platform for both markets

- Run on Commodity hardware

- 86 order type/attribute combinations (30 in current system)

- Real time updates to status of system via Control Workstation.

The Proposed Solution –
Integrates High-Value Architecture and Team Practices

Architecture-Centric Engineering

- Proven technology.

- Strongly addresses critical technical aspects of the early project lifecycle activities.

- Specific focus on architecting to meet business objectives.

- Key managers familiar with technology via training courses.

Team Software Process

- Proven technology.

- Strongly addresses management and measurement across the project lifecycle.

- Specific focus on building high-performance teams.

- Key managers familiar with technology only through word-of-mouth and literature.

TSP has a large "out-of-the-box" CMMI footprint. Architecure drove the work breakdown structure (WBS) and provided a robust framework for requirements management.

Software Engineering Institute | **Carnegie Mellon**

Architecture Drives the Lifecycle
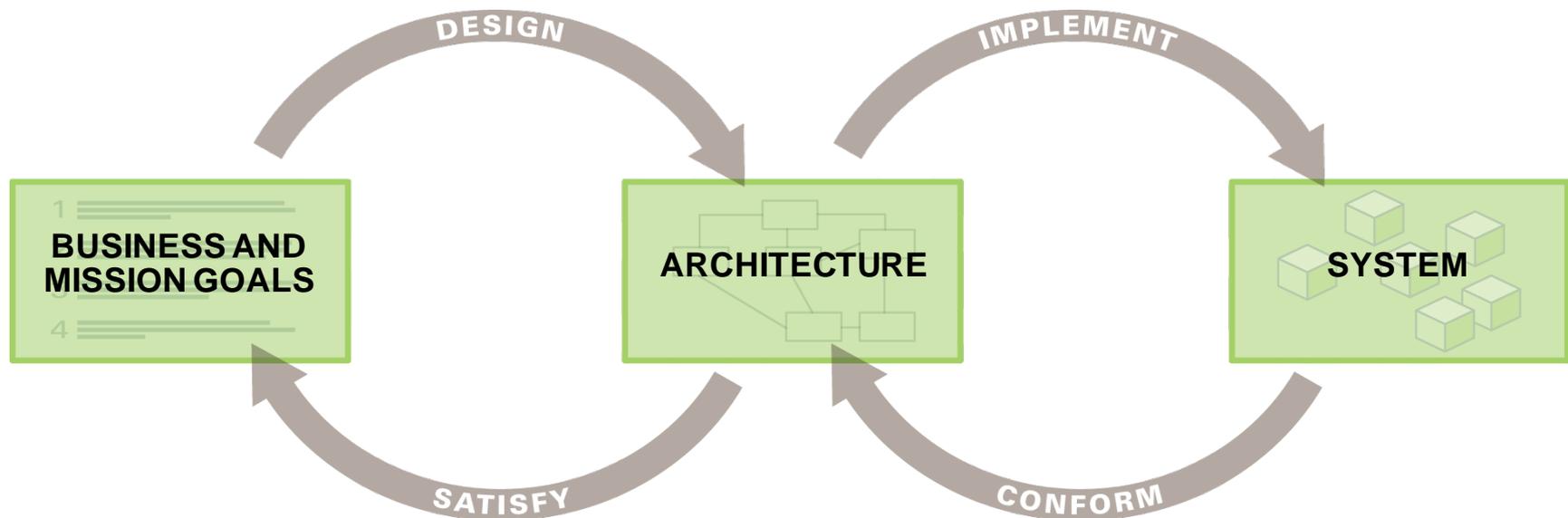
Two iterative processes based on the architecture of the system:

Design cycles (1, 2)
The goal is to design a system that ensures business success.

Implementation cycles (3, 5, 6)
The goal is to implement the system according to the design.

QAW/BTW – Building Quality Attribute Scenarios

The Quality Attribute Workshop (QAW) and Business Thread Workshop (BTW)

- bring together important internal and external stakeholders

- develop and validate key quality attribute scenarios that **quantitatively** define the most important **non-functional** requirements

- QAW focuses on developing quality attribute scenarios

- BTW focuses on business context to validate scenarios

Attribute-Driven Design (ADD) Method

ADD uses quality attribute scenarios to drive architectural design.

The process was time-boxed two ways.

- Six-week boxes to focus on

    — initial architectural (v1) while training architect team

    — refined architecture (v2) for early review or ATAM[1]

    — "complete" (not final) architecture (v3) for use by developers[2]

- Two-week boxes that focused on

    — developing the architecture

    — preparing for and performing ATAM-based peer-reviews with the "architecture coach"

1. *Development team was launched at this point*

2. *ATAM actually occurred at this point*

Views and Beyond for Architecture Documentation

"View and Beyond is not a method, but a collection of techniques:

1. Find out what architecture information stakeholders need.

2. Provide that information to satisfy the needs.

3. Capture the information in views, plus beyond-view information.

4. Package the information in a useful form to its stakeholders.

5. Review the result to see if it satisfied stakeholders' needs."

From the SEI class *Documenting Software Architectures,*
http://www.sei.cmu.edu/training/p33.cfm.

Active Review of Intermediate Designs (ARID)

An ARID was held in conjunction with a TSP relaunch.

The purpose of ARID is to

- put the architectural documents into the hands of developers

- ensure that the documents are fit for development use (right information recorded at sufficient level of detail)

- provide early "live" feedback to the architecture team

Architecture Trade-off Analysis Method (ATAM)

ATAM

- brings together a system's stakeholders

- evaluates the existing architecture with respect to the quality attribute scenarios

- focuses on surfacing architectural risks

- promotes & requires adequate documentation of the architecture

As mentioned previously, two-day ATAM-based peer-reviews were used by the architecture coach during development.

- on-the-job training for architecture team

- forced adequate documentation from the start

- fewer risks surfaced at formal ATAM than expected for size/scope of project

ACE / TSP Design, Analysis, and Implementation
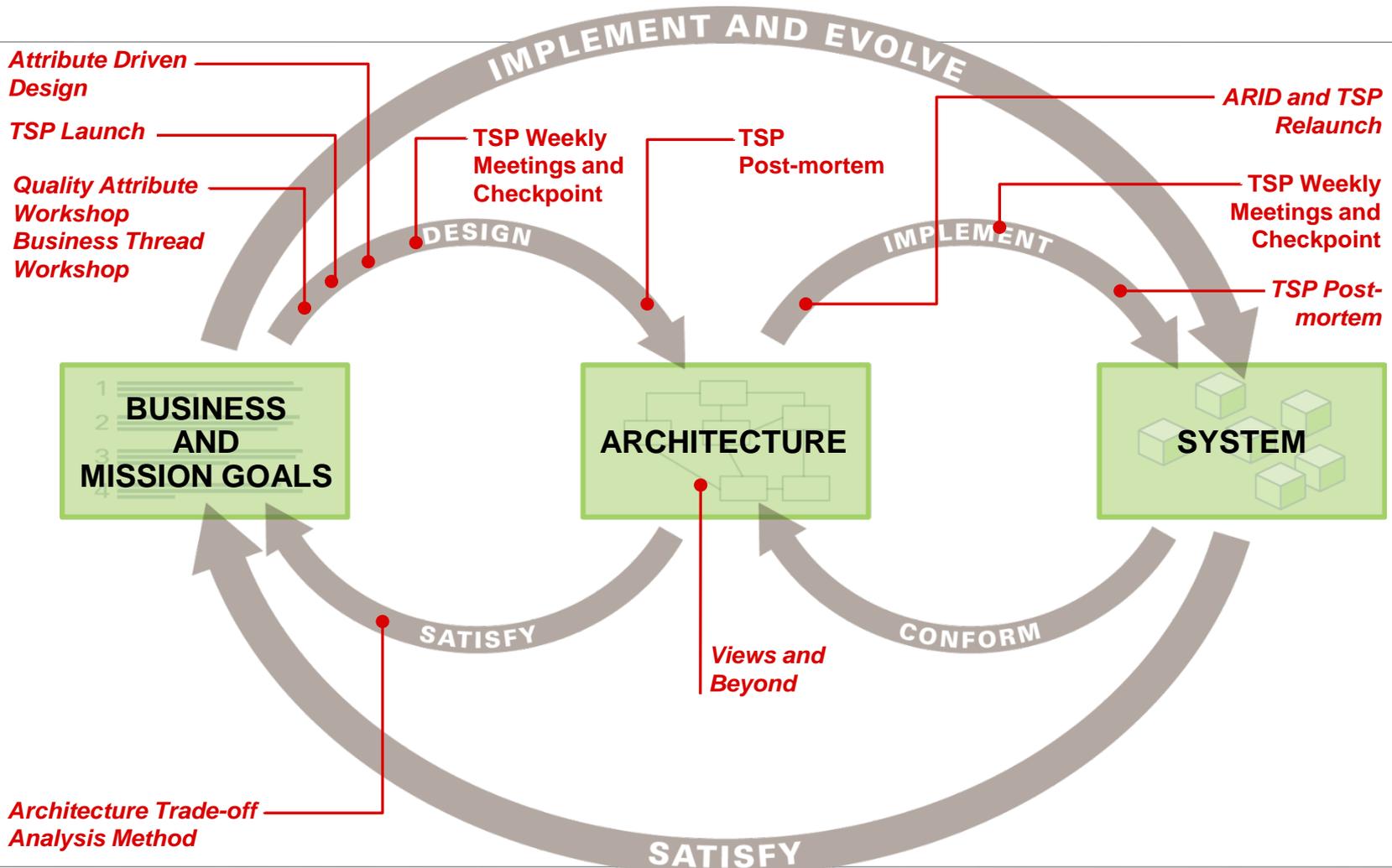
Trading Engine

Multicast Network

Horizontal Scalability

Legacy w/ Msg translation

HA

Special TSP Roles for Architecture

TSP defines certain standard roles on a software development team.

- "Staff" roles - planning, quality, process, support
- "Line" roles – customer interface (requirements), design, implementation, test

Planning and performing these roles have a large CMMI footprint.

The team defined three special roles to address critical architecture issues.

- Lead architect – a coming "standard" role
- Performance manager – the #1 quality attribute scenarios
- Garbage collection manager – the #1 technical risk to performance

Project History

Cycle 1 (Architecture) – Completed Jan. 2010 (on time), demonstrated architecture coaching for the first time, evaluation of comm. packages, built test framework

Cycle 2 (Infrastructure implementation) – Completed Apr. 2010 (on time), included successful ATAM in Mar. 2010 (documentation noticeably thorough, no significant new architectural risks discovered)

Cycle 3 (Basic functions and main performance loop) – Completed July 2010 (on time), good (not great) quality, performance exceeding requirements by more than a factor of 5

Cycle 4 (Non-TSP cycle, outside evaluation by world-class experts) – Completed Aug. 2010, JVM & high-speed redundant communications

Cycle 5 (Full normal operations, complete performance loop) – Completed Jan. 2011 (on time)

Cycle 6 (Full functionality incl. startup, shutdown, & maintenance modes) – Completed July 2011 (additional scope extended scheduled June finish)

Current Project Status – cont.

Cycle 7 – System Test / Integration Test

- ***ALL QUALITY ATTRIBUTES HAVE BEEN DEMONSTRATED AT OR BETTER THAN SPECIFIED LEVELS.***
- On Time (expected Oct. 2011 finish)
- Integration Test with Legacy systems

Cycle 8 – Acceptance Test / Parallel Test

- Internal user testing / certification
- Scheduled to start in 4Q'2011

Cycle 9 – User Test / Deployment

- Brokerage firms testing , including functional, HA, throughput and DRP tests
- Scheduled to start late 2011

Go-Live Scheduled 2Q'2012

Select Process Data

Measured size through cycle 7 (actual)

- ~208 eKLOC in 24 months

Effort distribution through cycle 6 (% of task hours)

Cycle 1	Cycle 2	Cycle 3	Cycle 5	Cycle 6
14.4	4.9	19.4	32.5	28.8

Effort distribution through cycle 6 (% by "block activities")

Mgt	Req	Arch	DLD	Code	Test	Other
3.7	17.5	12.0	18.5	32.2	14.5	1.5

25.3% of all recorded task hours through cycle 6 were some form of review or inspection, 48% requirements or design.

Current Project Status

- Very low defect count in System Test

- Defects encountered have not modified the Architecture

- Unit Test in place with high code coverage

- Testing Framework allowed a smooth continuous integration

- Regression tests done within the same day (except for multiday orders)

- Static analysis tools for Inspections and Architecture Integrity

- Latency and throughput metrics exceeded initial expectations

Key Takeaways

Architecture and TSP were focused on core of the system (Matching Engine)

Other key components would have benefitted with TSP such as:

- Message Format translator

- Trading Terminal

Most of the issues encountered have been with the interaction with legacy systems: Reporting, Billing, Market monitoring due to legacy fields.

Requirements / Inspections could be done better (including DLD interfaces with legacy systems) to have a better defect yield.

Future Potential for TSP & Architecture

This is not a complete set of possible TSP adaptations of architecture processes.

> Applying architecture methods to a large legacy system that requires significant enhancements demands different adaptations of the underlying principles.

> Applying SOA (service-oriented architecture) methods would be a related but different set of adaptations.

The presumption is that the appropriate combination of TSP and architecture methods meets the intent of the (new) CMMI practices.

Questions?

?

See the SEI website for information on their architecture-related conference, *SATURN 2012.*

http://www.sei.cmu.edu/saturn/

Contact Information

TSP Initiative

James W. Over

TSP Initiative Lead

jwo@sei.cmu.edu

Jim McHale

TSP Mentor Coach

jdm@sei.cmu.edu

RTSS Program

Linda Northrop

RTSS Program Director

lmn@sei.cmu.edu

Felix Bachmann

Architecture Mentor Coach

fb@sei.cmu.edu

Business Development

David Scherb   dscherb@sei.cmu.edu

Greg Such   gsuch@sei.cmu.edu

SEI website at www.sei.cmu.edu (~/tsp or ~/architecture)

Software Engineering Institute | **Carnegie Mellon**

Backup Slides

ACE Training
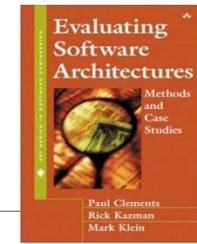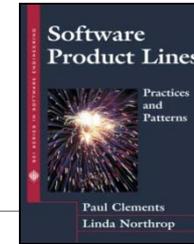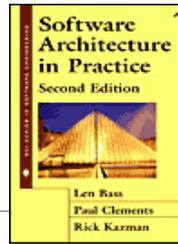
Requirements	CERTIFICATE PROGRAMS		CERTIFICATION
	Software Architecture Professional	ATAM Evaluator	ATAM Leader
Software Architecture: Principles and Practices course	●	●	●
Documenting Software Architectures course	●		●
Software Architecture Design and Analysis course	●		●
Software Product Lines course	●		
Software Architecture: Principles and Practices Exam	●	●	●
ATAM Evaluator Training course		●	●
ATAM Leader Training course			●
ATAM Observation			●

PSP^{SM} training

Personal Software Process (PSP^{SM}) training is *essential* to successful TSP implementation.
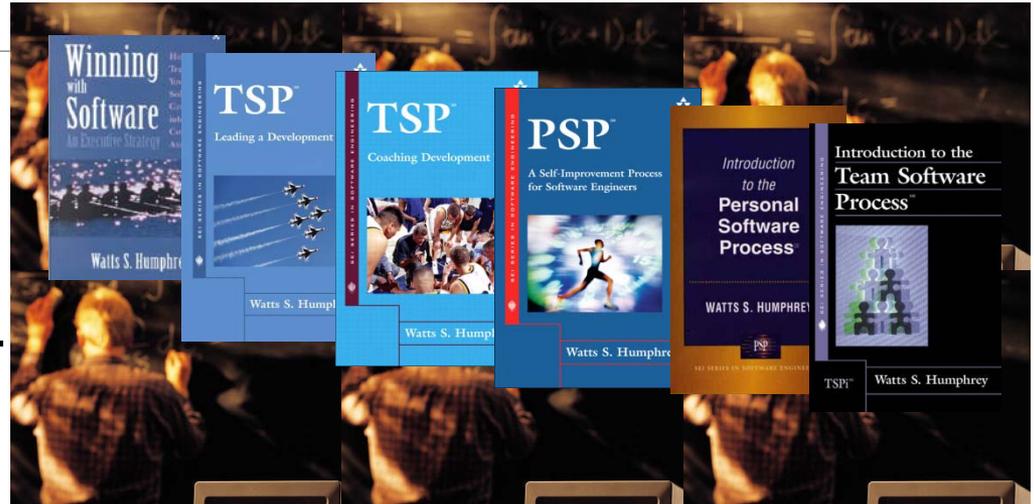
- *TSP Executive Seminar* (1 day for top-level execs, middle managers)

- *TSP Team Leader Training* (3 days for team leads, affected managers)

- *PSP Fundamentals* (5 days for software developers)

- *TSP Team Member Training* (3 days for other disciplines)

Building a High-Performance Engineering Team

The Team Software Process (TSP) is a development process for engineering teams

- Meet planned commitments
- Produce high-quality products
- Deliver working software on time/cost

The TSP provides a disciplined, measured approach to engineering.

Focus on quality, cost, and schedule performance by improving the management and engineering of software at the team and individual level.

Team Management
Team communication
Team coordination
Project tracking
Risk analysis

Team Building
Goal setting
Role assignment
Tailored team process
Detailed balanced plans

Team Member Skills
Process discipline
Performance measures
Estimating & planning skills
Quality management skills

The TSP Launch Process

1. Establish Product and Business Goals

2. Assign Roles and Define Team Goals

3. Produce Development Strategy

4. Build Top-down and Next-Phase Plans

5. Develop the Quality Plan

6. Build Bottom-up and Consolidated Plans

7. Conduct Risk Assessment

8. Prepare Management Briefing and Launch Report

9. Hold Management Review

Launch Postmortem

The TSP launch process produces necessary planning artifacts, e.g. goals, roles, estimates, task plan, milestones, quality plan, risk mitigation plan, etc.

The most important outcome is a committed team.

TSP-ACE Development Process
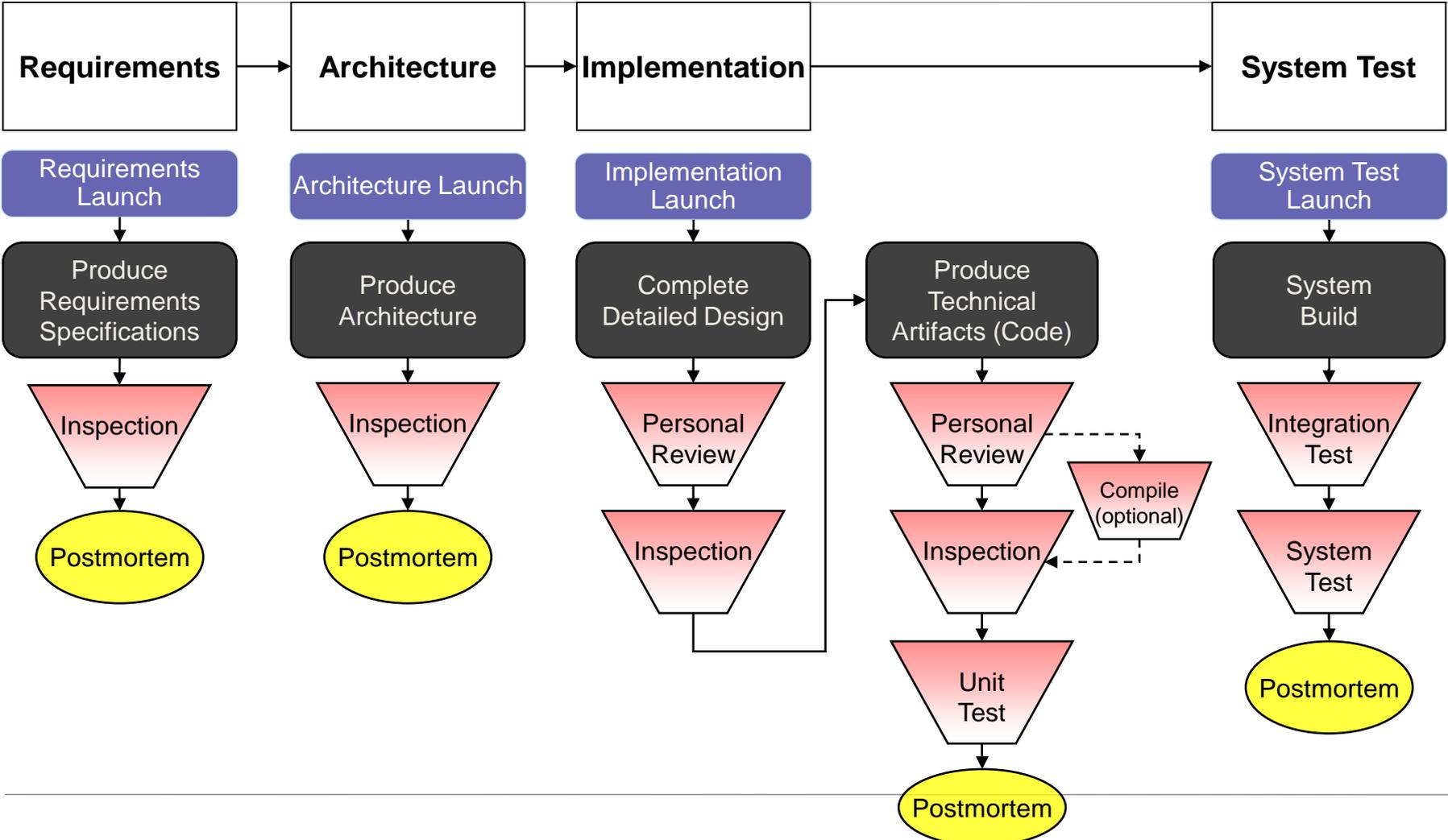
TSP Guidelines for Architecture Methods -1

Training (SEI courses – SAPP, DSA, SADA, ESA)

- *Software Architecture Principles & Practices (2 days or 11 hrs. online)*

- *Documenting Software Architectures (2 days – some concepts overlap with PSP design templates)*

- *Software Architecture Design and Analysis (2 days)*

- *Evaluating Software Architecture (2 days – can be replaced by an architecture coach; recommended for TSP coaches)*

TSP Guidelines for Architecture Methods -2

For first projects:

- An architecture coach is essential for inexperienced teams, replacing ESA training.

- ESA may be sufficient for experienced teams, especially if there is architecture expertise elsewhere in the organization.

- Expertise in defining and capturing quality attributes (QAW) and evaluating architectures (ATAM) is worth the price.

Architectural Process Assets

- Views & Beyond (taught in DSA) informs design standards.

- ADD (a subject in SADA) is the basic architecture design process.

- Lead Architect is more than a design manager.