The Opportunity

Background:

- Bolsa Mexicana de Valores (BMV) operates the Mexican financial markets under license from the federal government.
- Bursatec is the technology arm of 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 three existing trading engines with one in-house developed system.
Using Architecture-Centric Engineering on a TSP Project – Carballo, Mchale, Nord

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 spikes and long-term growth in trading volume)

Architecture Decisions:

- Development in Java (lower Total Cost of Ownership)
- Low Latency Communication Multicast Network
- In memory data storage during trading session
- Hot-Hot High Availability configuration
- Parallel processing in Java Virtual Machine (JVM)
- Horizontal scalability

The Project -2

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.

Bursatec approached the SEI for support during design and 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
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 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

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

<table>
<thead>
<tr>
<th>Team Software Process</th>
<th>Architecture-Centric Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proven technology.</td>
<td>Proven technology.</td>
</tr>
<tr>
<td>Strongly addresses management and measurement across the project lifecycle.</td>
<td>Strongly addresses technical aspect of the early project lifecycle activities.</td>
</tr>
<tr>
<td>Specific focus on building high-performance teams.</td>
<td>Specific focus on architecting to meet business objectives.</td>
</tr>
<tr>
<td>Key managers familiar with technology only through word-of-mouth and literature.</td>
<td>Key managers familiar with technology via training courses.</td>
</tr>
</tbody>
</table>
Polling Question

Have you used:

- Architecture practices with TSP (or CMMI)
- Architecture practices (without TSP)
- TSP/PSP (without architecture practices)
- Neither

The Engineering Process

Two iterative processes based on the architecture of the system:

*Design cycle.*
The goal is to design a system that ensures business success.

*Implementation cycle.*
The goal is to implement the system according to the design.
The Engineering Process – Design Cycle

Designing a software system is defining structures that support the quality attribute requirements, such as performance, availability, extensibility, etc.

The methods used in the design cycle are a combination of a

- Attribute Driven Design (ADD) method to transform the quality attribute requirements into an appropriate design
- Views & Beyond (V&B) documentation method to document the design to be used for evaluation and development
- Peer review process to continuously check the design to identify weaknesses, using Architecture Tradeoff Analysis (ATAM) techniques.

Other activities, such as quality attribute modeling might be necessary.

The Engineering Process – Implementation Cycle

The implementation cycle is centered around establishing communication between architects and developers.

An architecture that shows how well the scenarios are supported is not enough to be implemented correctly.

- Module views help to describe the architecture clearly enough for implementation.
- Active design reviews (ARID) help to communicate the architecture effectively to the developers.
- Conformance reviews help to continuously check if code and architecture are synchronized.
TSP Launch and Coaching Support

TSP supports an iterative or cyclic development strategy.
TSP can be introduced starting at any phase or any cycle.
Each cycle starts with a launch or re-launch and ends with a postmortem.
The TSP coach guides the team through each launch, re-launch, and postmortem, and provides weekly coaching support during the cycle.

ACE / TSP Design, Analysis, and Implementation

BUSINESS AND MISSION GOALS

ARCHITECTURE

SYSTEM

IMPLEMENT AND EVOLVE

SATISFY
Using Architecture-Centric Engineering on a TSP Project – Carballo, Mchale, Nord

ACE / TSP Design, Analysis, and Implementation

BUSINESS AND MISSION GOALS

ARCHITECTURE

SYSTEM

Attribute Driven Design
Quality Attribute Workshop
Business Thread Workshop
TSP Launch
TSP Weekly Meetings and Checkpoint
TSP Post-mortem
ARID and TSP Relaunch
TSP Weekly Meetings and Checkpoint
TSP Post-mortem
Views and Beyond

Architecture Trade-off Analysis Method

ACE / TSP Design, Analysis, and Implementation

BUSINESS AND MISSION GOALS

ARCHITECTURE

SYSTEM

Attribute Driven Design
Quality Attribute Workshop
Business Thread Workshop
TSP Launch
TSP Weekly Meetings and Checkpoint
TSP Post-mortem
ARID and TSP Relaunch
TSP Weekly Meetings and Checkpoint
TSP Post-mortem
Views and Beyond

Architecture Trade-off Analysis Method
Example Design and Implementation Strategy

Bursatec Overall Schedule – Phase I

(based on an initial notional schedule by SEI)

Software Engineering Institute | Carnegie Mellon
Using Architecture-Centric Engineering on a TSP Project – Carballo, Mchale, Nord

Some Early Challenges -1

Size estimates and size measurements

- TSP does a conceptual design for initial estimates and structuring of project plans.
- ACE calls for initial architectural concepts or approaches, parts of which will often survive into the final architecture.
- Bursatec had already done a lot of analysis for the buy vs. build decision, including effort estimates and initial architectural concepts.

The launch fell back on these initial estimates plus a number of user stories based on the quality attribute scenarios from the QAW.

The early architectural concepts helped to get better size estimates and to think about work allocation earlier.

Some Early Challenges -2

Defect management

- ACE focuses on architectural risks, but a TSP coach might be tempted to call them “defects injected in HLD”.

Risk management

- In TSP, risks are usually risks to the project plan – but that might include architectural risks as well!

The good news – at least one team in Mexico hasn’t had any trouble dealing with potentially ambiguous English terminology.
Some Early Challenges -3

Other process issues

- Early design is exploratory in nature, becoming progressively more conventional.
- ARIDs and ATAMs – not just new names for a review or inspection.
- Agile practices – daily meetings, planning poker without the cards, two levels of time-boxing.
- Lead Architect and Performance Manager roles added to TSP to oversee architecture “fitness” both generally and specifically for the most visible quality attribute.
- Architecture conformance checks included in the cycles.

Important Lessons Learned (So Far)

TSP and ACE are not simply compatible, they are complementary.

Learning cycles can be shortened; they cannot be short-cut!

Architecture can be coached.

TSP provides a disciplined framework for measuring and managing any structured intellectual activity.

Architectural awareness helps to structure the team and the work in addition to the product.
Using Architecture-Centric Engineering on a TSP Project – Carballo, Mchale, Nord

Current Project Status

Cycle 1 - Architecture
- completed Jan. 2010
- demonstrated coaching, evaluation of comm. packages, built test framework

Cycle 2 - Infrastructure implementation
- completed Apr. 2010
- successful ATAM (no significant new architectural risks discovered)

Cycle 3 - Basic functions and main performance loop
- completed July 2010
- good 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 and high-speed redundant communications

Cycle 5 - Full normal operations, complete performance loop
- completed Jan. 2011, full round-trip performance proven

Cycle 6 - Full functionality incl. startup, shutdown, and maintenance modes
- scheduled June 2011 (currently on time)

Experience and Results

The ACE methods and architecture coaching, coupled with the discipline of the TSP, built a competent architecture team (and an architecture with successful ATAM!) quickly – less than 6 months.

The project objectives were met.
- Schedule – finished early.
- Quality – early trials and quality metrics suggest reliability and quality goals were met.
- *No known defects carried into final cycle!*
- Performance – a day’s worth of transactions can be processed in seconds.
Using Architecture-Centric Engineering on a TSP Project – Carballo, Mchale, Nord

Select Process Data

Through cycle 5 (count)
- ~100 eKLOC in 68 weeks
- ~58 eKLOC main engine
- ~29 eKLOC infrastructure
- ~13 eKLOC test structure

Through cycle 6 (estimate)
- ~137 eKLOC in 89 weeks

Through cycle 6 (estimate including cycle 5 actuals)
- 11.9% architecture (includes design cycle activities with the exception of QAW/BTW; does NOT include architecture maintenance, conformance, or testing activities)
- 13.1% test (unit, integration, & limited system testing)

eKLOC – effective thousands of lines of code

Questions?

?
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)

ADDITIONAL INFORMATION
Getting Started

TSP/ACE is introduced into an organization on a project-by-project basis.

<table>
<thead>
<tr>
<th>TSP Introduction Steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Start by identifying candidate projects, architects, and internal architecture and TSP coach candidates.</td>
</tr>
<tr>
<td>2. Train senior management.</td>
</tr>
<tr>
<td>3. Train the selected teams and their managers, then launch the project.</td>
</tr>
<tr>
<td>4. Monitor the projects and make adjustments as needed.</td>
</tr>
<tr>
<td>5. Expand the scope to include additional projects and teams.</td>
</tr>
<tr>
<td>6. Create or expand the pool of available SEI-authorized architects, instructors and coaches.</td>
</tr>
<tr>
<td>7. Repeat starting at step 3.</td>
</tr>
</tbody>
</table>

Selecting Pilot Projects

Pick 3 to 5 medium-to large-sized pilot projects.

- 8 to 15 team members
- 4 to 18 month schedule
- Software-intensive new development or enhancement
- Representative of the organization’s work
- Important projects

Select teams with members and managers who are willing to participate.

Consider the group relationships.

- Contractors
- Organizational boundaries
- Internal conflicts
## Architecture Training

<table>
<thead>
<tr>
<th>Certificate Programs</th>
<th>ATAM Evaluator</th>
<th>ATAM Leader</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Six Courses</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software Architecture</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Principles and Practices*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Documenting Software Architectures</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Software Architecture</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Design and Analysis</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software Product Lines</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ATAM Evaluator Training</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ATAM Leader Training</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>ATAM Observation</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

* ✓: required to receive certificate / certification
  ✓: available through e-learning

### PSP℠ and TSP Training

Personal Software Process (PSP℠) 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)
Intellectual Property

Personal Software Process, PSP, Team Software Process, and TSP are service marks of Carnegie Mellon University.

Architecture Tradeoff Analysis Method and ATAM are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.