



# Driving Out Technical Risk by Blending Architecture, Process, and Project Discipline

Software Engineering Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213

James McHale, Robert Nord

In collaboration with:  
Luis Carballo (Bursatec)



## Copyright 2012 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

### NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

This material has been approved for public release and unlimited distribution except as restricted below.

The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013 and 252.227-7013 Alternate I.

Internal use:\* Permission to reproduce this material and to prepare derivative works from this material for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

External use:\* This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at [permission@sei.cmu.edu](mailto:permission@sei.cmu.edu).

\* These restrictions do not apply to U.S. government entities.



# A New Trading Platform

---

## FINANCIAL TIMES

---

January 9, 2012 12:45 am

### Mexico's BMV to be among fastest bourses by May

*“Mr Téllez [BMV president] said that the new system’s initial configuration will be able to handle 200,000 transactions per second, and take less than 100 microseconds to process a single order.*

*That catapults the BMV into the same league, in terms of latency, as large exchanges such as Nasdaq OMX, NYSE Euronext, Deutsche Börse and the London Stock Exchange.”*



# Experience and Results

Architecture coaching, coupled with the discipline of the Team Software Process, built a competent architecture team and an architecture with successful evaluation quickly – less than six months.

The project objectives were met.

- Schedule – finished on time.
- 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.

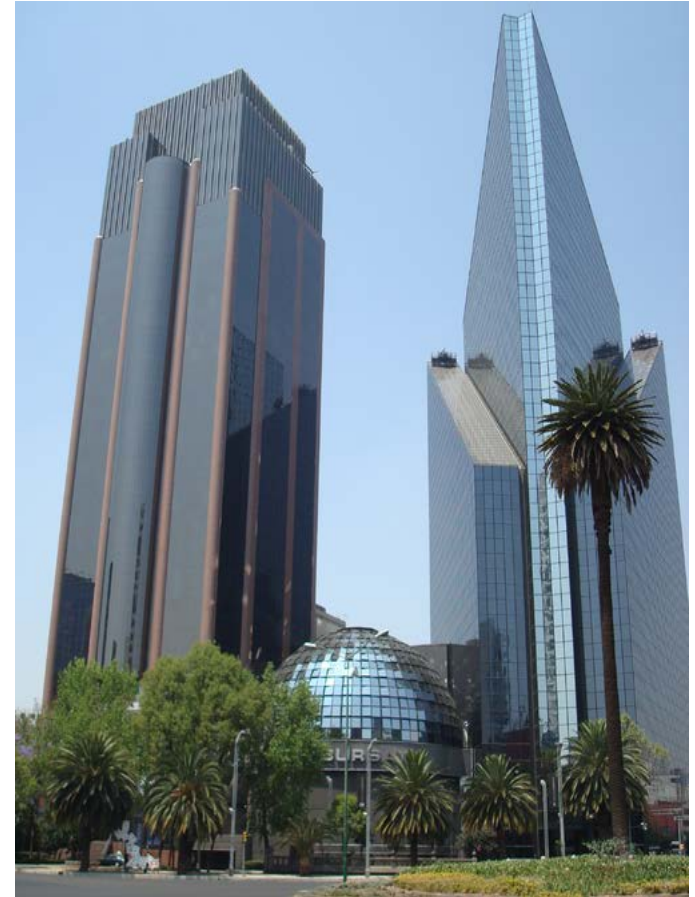


# The Opportunity

---

## Background:

- Bolsa Mexicana de Valores (BMV) operates the Mexican financial markets.
- 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.
- Bursatec committed to deliver a trading engine in 8-10 quarters.
- Bursatec approached the SEI for support during design and development to improve its software delivery capability.

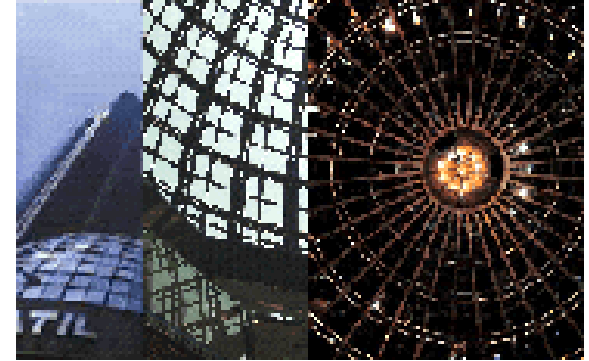


# The Project -1

---

## Business Goals:

- 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)



# The Project -2

## 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

## 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.



# Trading Engine Quality and Other Attributes

---

## Quality Attributes

- Under 1ms processing latency
- Horizontal scalability
- Redundant high availability system
- Warm dual redundant system
- Automatic testing framework (one day turnaround attribute)
- Localize business rules changes in specific modules

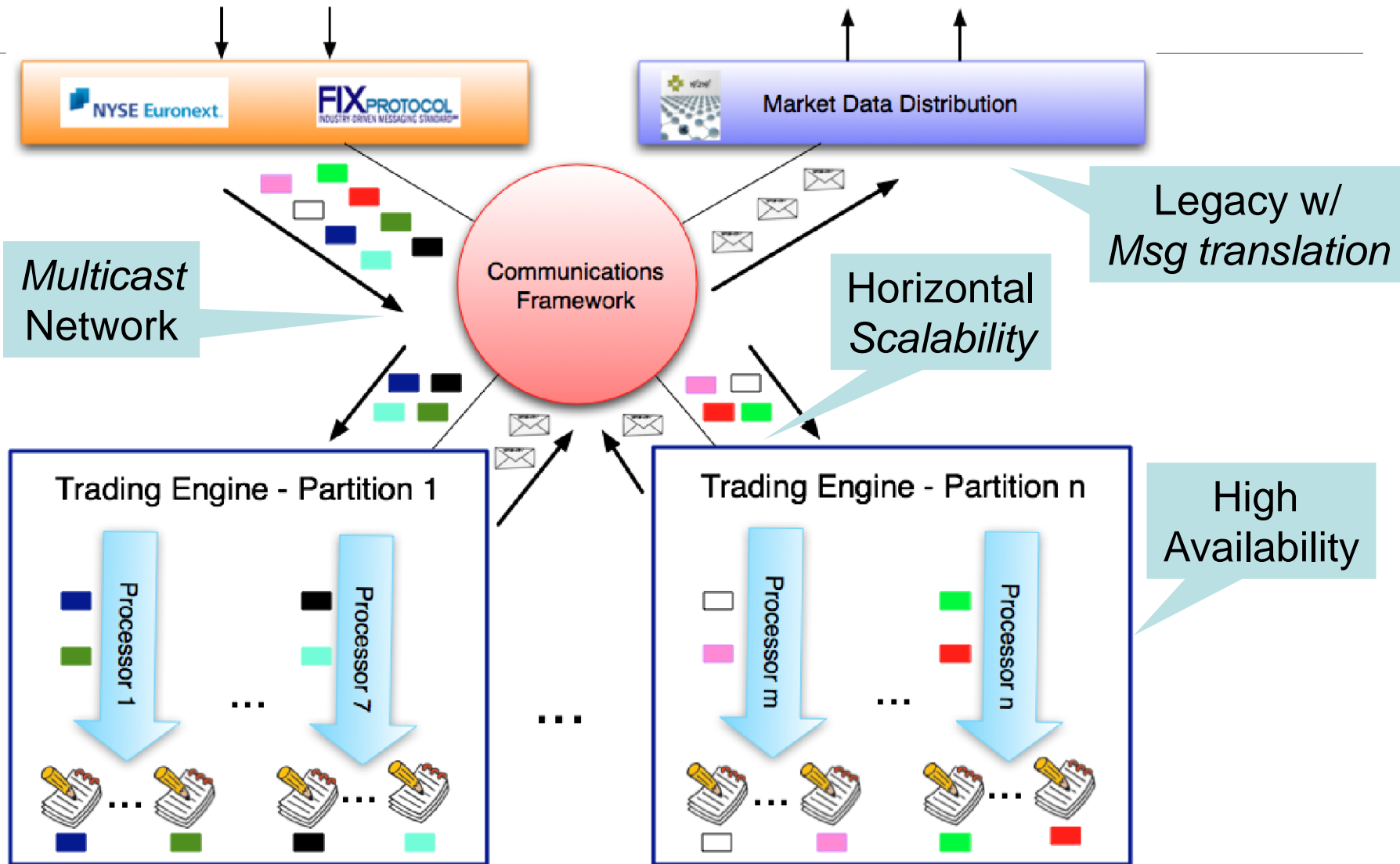
## Other Attributes

- 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.





# System Context of the Trading Engine



# Complicating Factors

---

Given the context, one would expect risks due to:

- Large project – scope beyond the organization’s recent experience
  - # of person-months
  - # KLOC/function points
  - # of interconnecting platforms
  - # of individual projects
- Inexperience – available staff talented but young and key implementation technologies never used together formally
- Constant stream of new requirements and changes to business rules



# Solution Integrates High-Value Architecture and Team Practices

---

## Team Software Process

- Proven technology
- Management and measurement across the project lifecycle
- Building high-performance teams
- Key managers familiar with technology through word-of-mouth and literature.

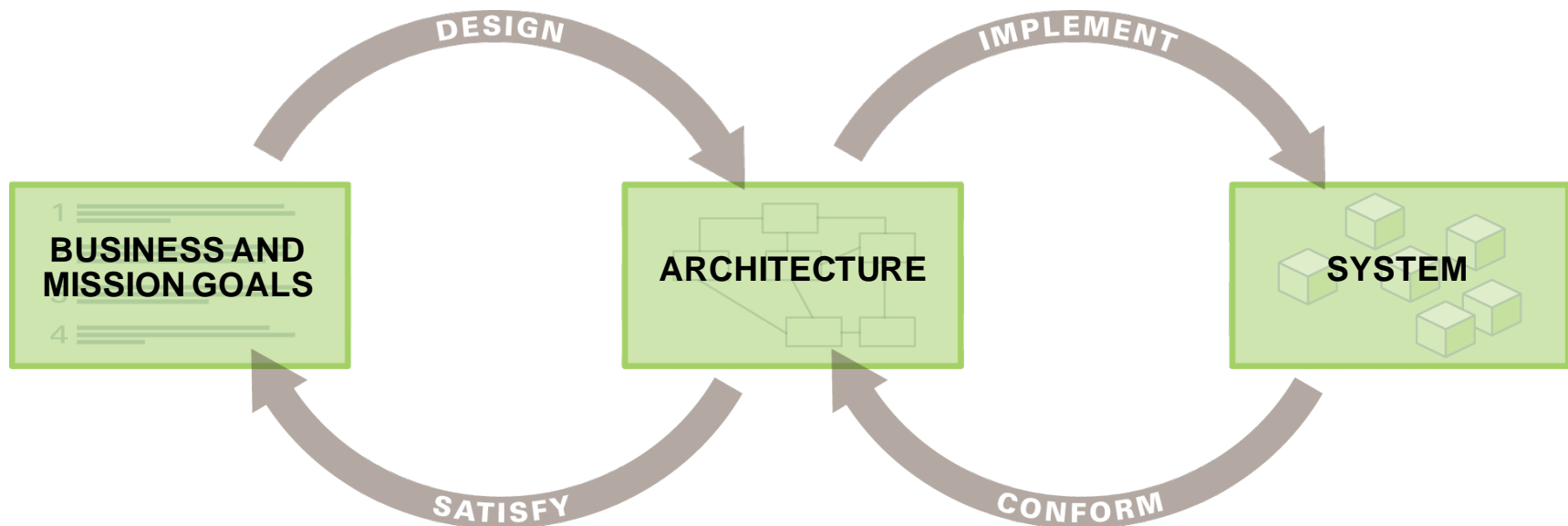
## Architecture-Centric Engineering

- Proven technology
- Technical aspect of the early project lifecycle activities
- Architecting to meet business objectives
- Key managers familiar with technology via training courses.

*Architecture drove the work breakdown structure (WBS) and provided a robust framework for requirements management.*



# Architecture Drives the Lifecycle



Two iterative processes based on the architecture of the system:

*Design cycles (1, 2)*

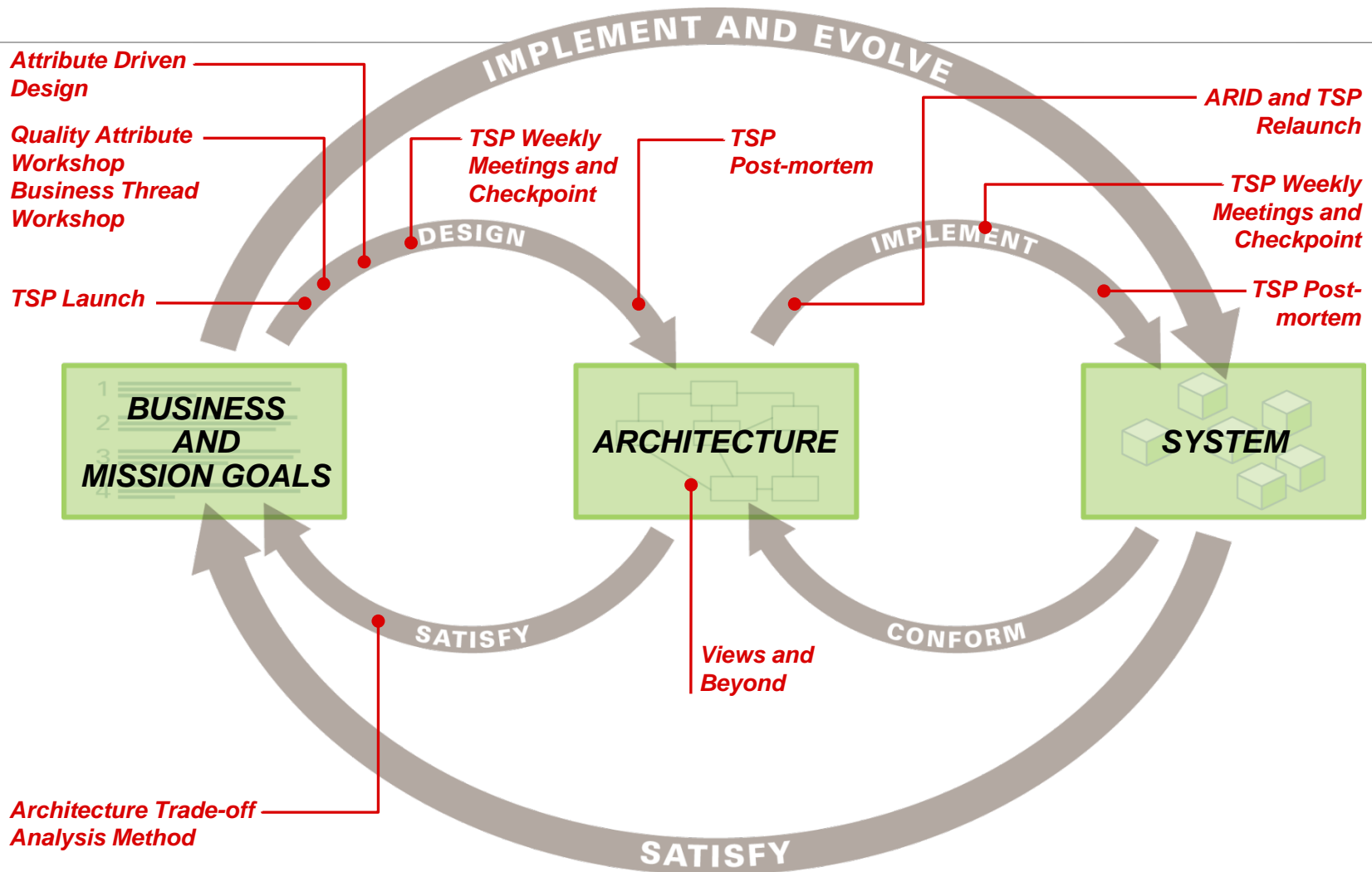
The goal is to design a system that ensures business success.

*Implementation cycles (3, 5, 6)*

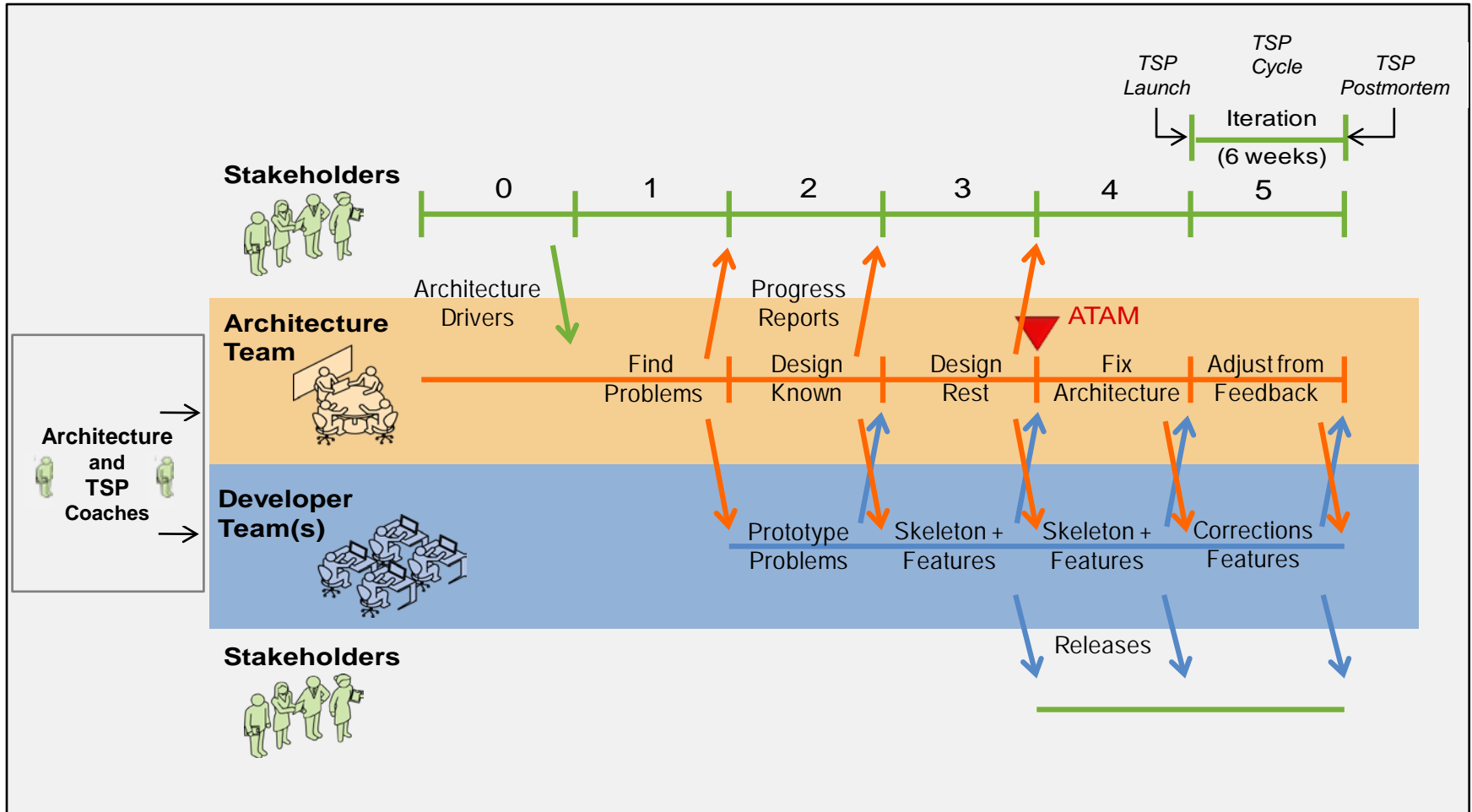
The goal is to implement the system according to the design.



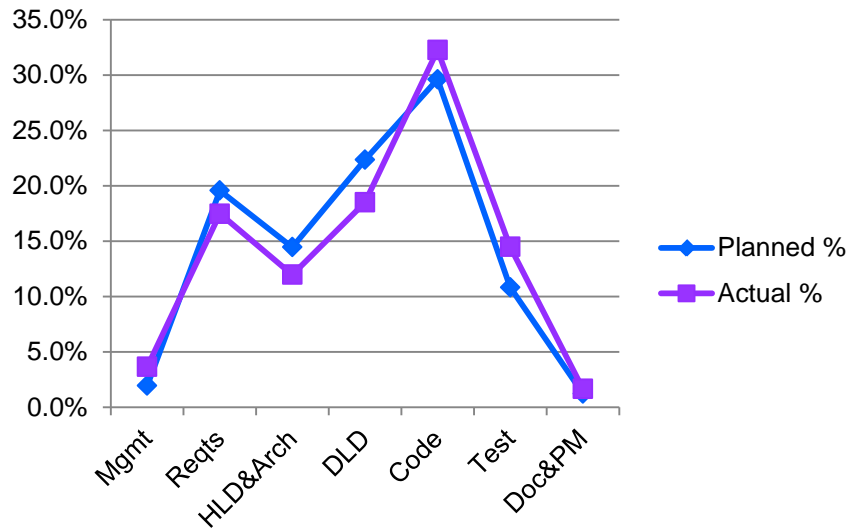
# ACE / TSP Design, Analysis, and Implementation



# Example Design and Implementation Strategy



# Select Process Data



## Effort distribution by phase blocks (% of total task hours)

- ~208 eKLOC in 24 months
- Complete functionality of previous system and new functionality
- Latency target 1msec, achieved 0.1 msec

- Architectural design practices were 12% of the total cost but were key in meeting the technical requirements and are estimated to have reduced the implementation costs by 10%-15% (due to avoided functionality and clean design)
- Only 15% effort in testing – compared to normally equal distribution between coding and testing; higher than usual quality achieved



# Accomplishments

---

## Performance

- Latency and throughput metrics greatly exceeded initial expectations (0.1 msec. vs. 1.0 msec.)

## Quality

- Very low defect count in validation test. Error density 0.1 error per KLOC compared to “normal” of 0.5-1.0
- Defects encountered have not modified the architecture
- Testing framework allowed a smooth continuous integration

## Cost & Schedule

- Team achieved EVERY Milestone (internal and external) on time and budget (including unplanned new functionality), with the planned number of people. No “forced march”.





# Key Takeaways

---

Investment in early architecture and team practices drive the lifecycle and plays a role in managing risk.

Earlier identification and resolution of defects reduces the cost of rework.

Iterative and incremental approach fosters collaboration and facilitates handoffs reducing the cost of delay.

Architectural practices and TSP provide a disciplined framework for measuring and managing structured intellectual activity related to the product, process, and project.



# Questions?

---



For more information: Combining Architecture-Centric Engineering with the Team Software Process, Technical Report, CMU/SEI-2010-TR-031, December 2010



# Contact Information

---

## TSP Initiative

James W. Over  
TSP Initiative Lead  
[jwo@sei.cmu.edu](mailto:jwo@sei.cmu.edu)

Jim McHale  
TSP Mentor Coach  
[jdm@sei.cmu.edu](mailto:jdm@sei.cmu.edu)

## Business Development

David Scherb [dscherb@sei.cmu.edu](mailto:dscherb@sei.cmu.edu)  
Greg Such [gsuch@sei.cmu.edu](mailto:gsuch@sei.cmu.edu)

SEI website at [www.sei.cmu.edu](http://www.sei.cmu.edu) (~/tsp or ~/architecture)

## RTSS Program

Linda Northrop  
RTSS Program Director  
[lmn@sei.cmu.edu](mailto:lmn@sei.cmu.edu)

Felix Bachmann  
Architecture Mentor Coach  
[fb@sei.cmu.edu](mailto:fb@sei.cmu.edu)



---

# ADDITIONAL INFORMATION



# Project History

---

## Cycle 1 – Architecture

- Completed Jan. 2010, demonstrated architecture coaching, evaluation of comm. packages, built test framework

## Cycle 2 – Infrastructure implementation

- Completed Apr. 2010, included successful ATAM (documentation noticeably thorough, 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 & high-speed redundant communications

## Cycle 5 – Full normal operations, complete performance loop

- Completed Jan. 2011

## Cycle 6 – Full functionality incl. startup, shutdown, & maintenance modes

- Completed July 2011 (additional scope extended scheduled June finish)
- 



# Project History -2

---

Cycle 7 – System Test / Integration Test w/ Legacy Systems (starting Aug. 2011)

Cycle 8 – Acceptance Test / Parallel Test (starting Dec. 2011)

Cycle 9 – User Test / Deployment (starting Jan. 2012)

- Testing activities overlapped in part due to the Matching Engine readiness being AHEAD of other interfacing systems
- Includes internal test group, internal operations, brokerage firms testing (functional, HA, throughput ,and DRP tests)
- Operations testing detected (as of Jan. 2012) < 50 unique defects in 200+ KLOC

Go-Live Scheduled May 2012 (NOW!)



# Getting Started

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

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



# 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.

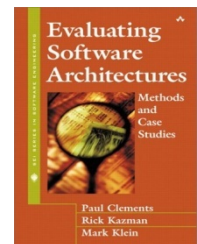
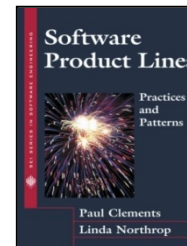
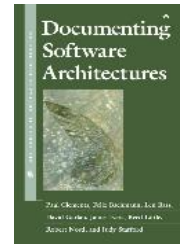
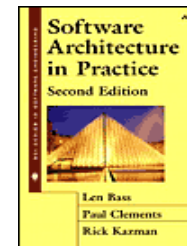
- Contractors
- Organizational boundaries
- Internal conflicts





# Architecture Training

	Certificate Programs		Certification
	Software Architecture Professional	ATAM Evaluator	ATAM Leader
<i>Six Courses</i>			
Software Architecture Principles and Practices*	✓	✓	✓
Documenting Software Architectures	✓		✓
Software Architecture Design and Analysis	✓		✓
Software Product Lines	✓		
ATAM Evaluator Training		✓	✓
ATAM Leader Training			✓
ATAM Observation			✓

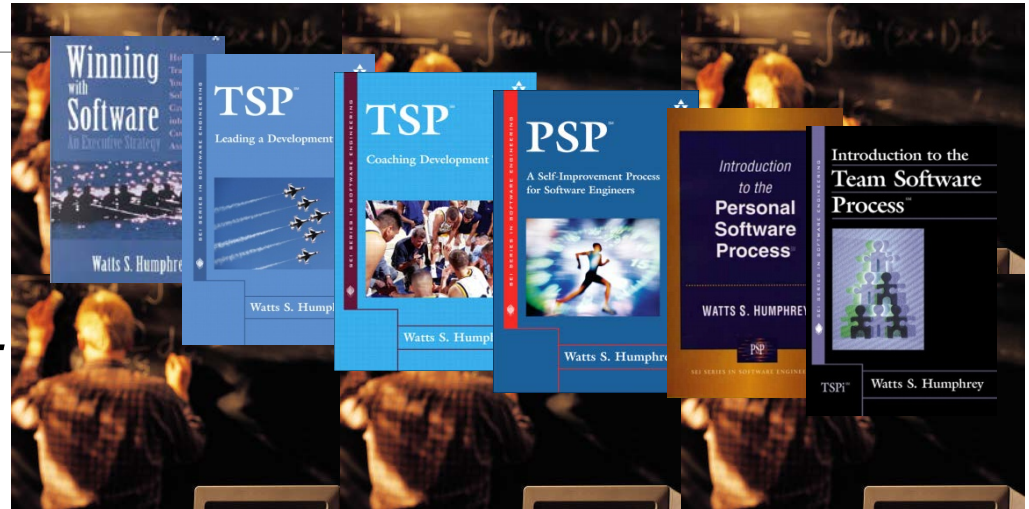


✓ : required to receive certificate / certification  
 \*: available through e-learning



# PSP<sup>SM</sup> and TSP Training

*Personal Software Process (PSP<sup>SM</sup>) 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.



## NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

This Presentation may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at [permission@sei.cmu.edu](mailto:permission@sei.cmu.edu).

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

