Architecture + TSP = High Quality + Fast

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Felix Bachmann, James McHale,
Robert Nord

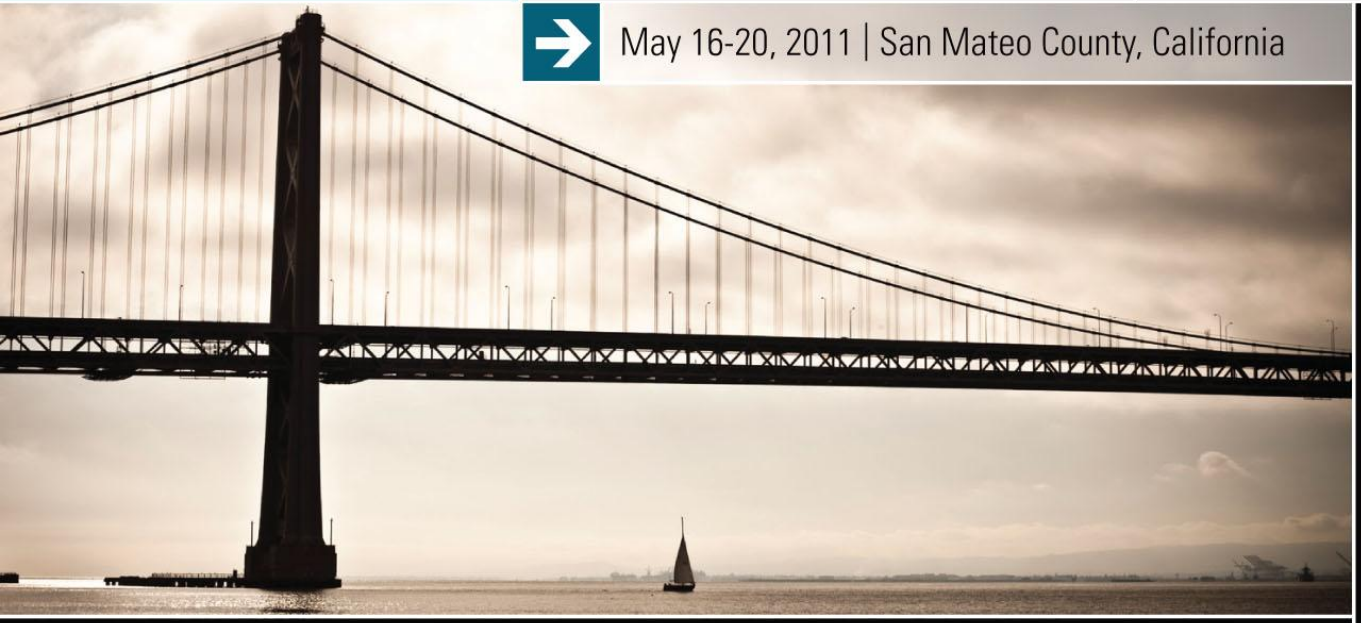
March 17, 2011

Twitter: #seiwebinar

SATURN 2011

Seventh Annual SEI Architecture Technology User Network Conference

Architecting the Future



May 16-20, 2011 | San Mateo County, California

The SEI Architecture Technology User Network (SATURN) Conference brings together experts to exchange best architecture-centric practices in developing, acquiring, and maintaining software-reliant systems.

7 Things You Need to Know About the Next 7 Years in Architecture.


Architecture is Not Just for Architects


Architecture, Agile Development, and Business Agility


Soft Skills for Architects


Service-Oriented Architecture (SOA) and Cloud Computing


Architectural Knowledge Management


Architecting to Meet Tomorrow's Global Challenges


Model-Driven Architecting

www.sei.cmu.edu/saturn/2011

Software Engineering Institute | Carnegie Mellon
in collaboration with IEEE Software

TSP SYMPOSIUM 2011

6th Annual Software Engineering Institute (SEI)
Team Software Process (TSP) Symposium

September 19-22, 2011 • Georgia Tech Hotel and Conference Center • Atlanta, Georgia

KEYNOTES

Capers Jones, Author, Researcher, and Leading Authority on Software Quality, Risks, and Best Practices

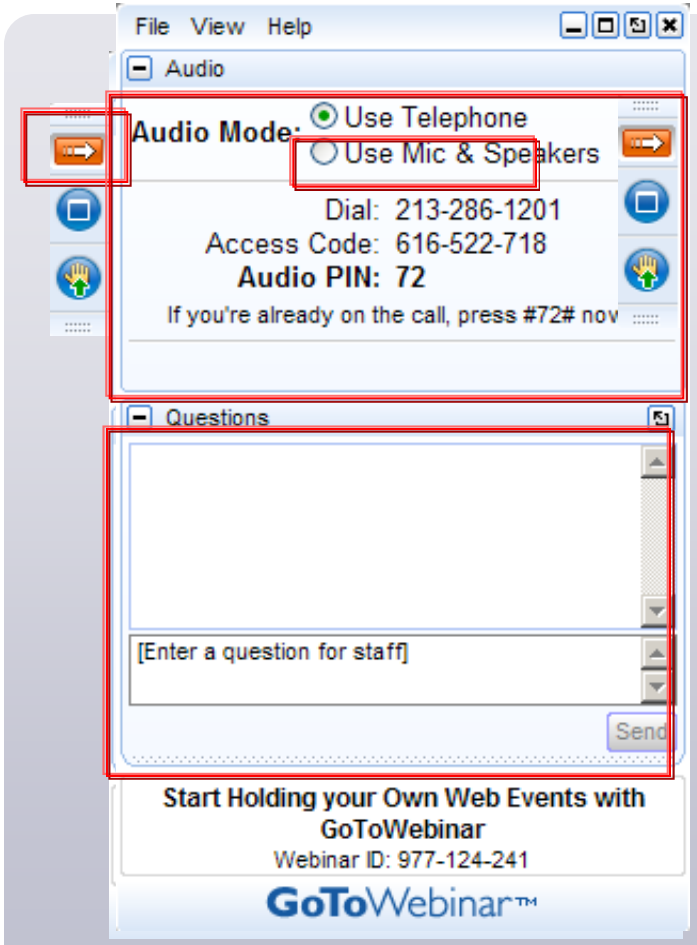
Michael Sapenter, CGI Federal, Inc.
Carl Wyrwa, Beckman Coulter, Inc.

Software Engineering Institute
Carnegie Mellon

CALL FOR PARTICIPATION

Twitter: #seiwebinar

How to Participate Today



Open and close your Panel

View, Select, and Test your audio

Submit text questions

Q&A addressed at the end of today's session

About the Presenters

Felix Bachmann

- Senior member of the technical staff.
- Certified ATAM Leader, instructor of ATAM Evaluator Training.
- Co-author “Documenting Software Architectures: Views & Beyond.”

James McHale

- Senior member of the technical staff.
- Authorized TSP coach, SCAMPI Lead Appraiser candidate.
- Instructor of PSP and TSP course suite, Introduction to CMMI.
- Co-author technical reports relating TSP to process improvement models such as CMM and CMMI.

Robert Nord

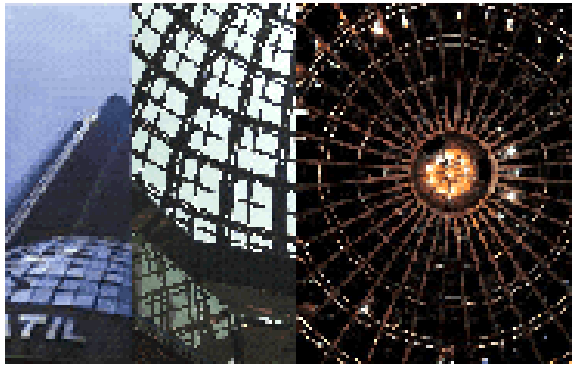
- Senior member of the technical staff.
- ATAM Evaluator, instructor of Software Architecture Design & Analysis.
- Co-author technical reports relating architecture practices to life cycle development processes such as the TSP, the Rational Unified Process, and Agile practices.

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 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



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 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 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

The Proposed Solution

Engineering process using Architecture-Centric Engineering (ACE) practices

- Proven technology
- Strongly addresses technical aspect of the early project lifecycle (requirements, architecture design)
- Key managers familiar with technology via training courses

Management process using the Team Software Process (TSP)

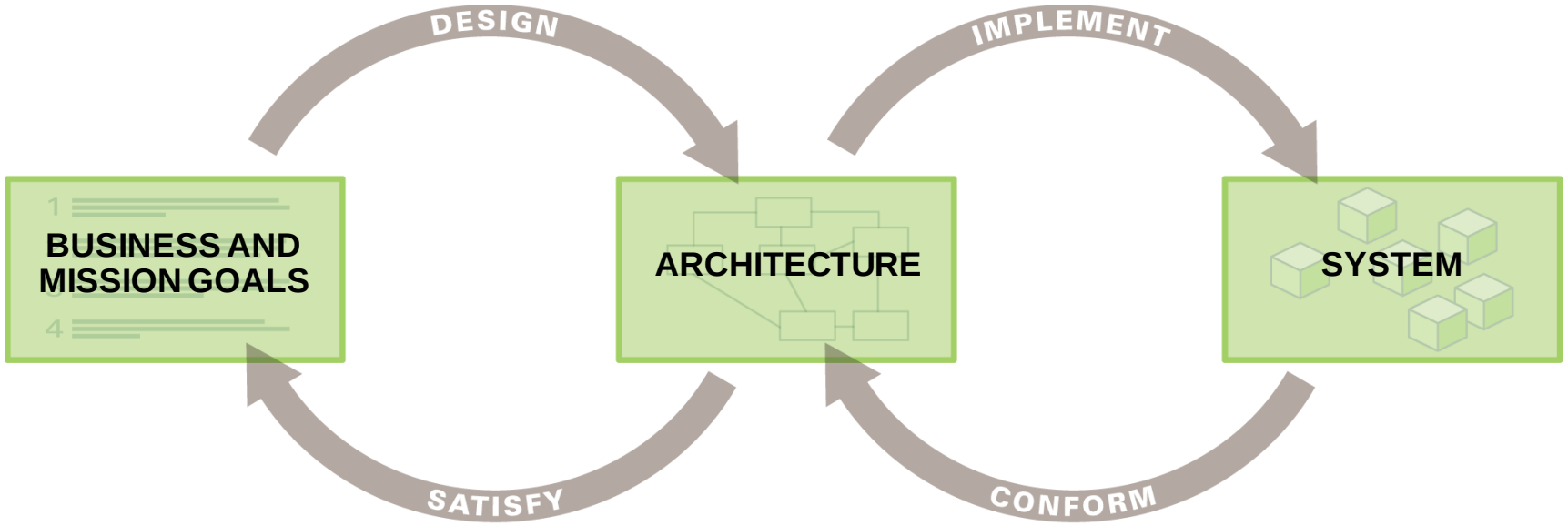
- Proven technology
- Strongly addresses management and measurement across the project lifecycle, especially later phases (implementation, test)
- Key managers familiar with technology only through word-of-mouth and literature

Polling Question #1

Have you used:

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

The Engineering Process

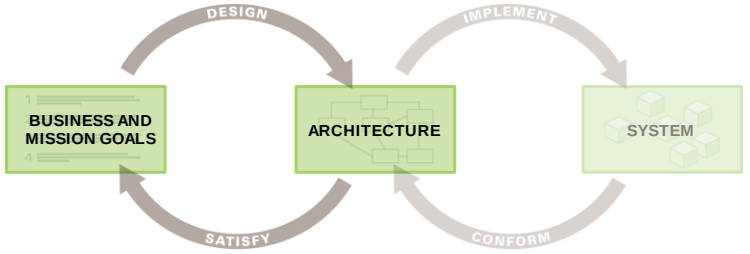


Two iterative processes based on the architecture of the system:

Design cycle. The goal of the iterations shown on the left side is to design a system that ensures business success.

Implementation cycle. The goal of the iterations shown on the right side is to implement the system according to the design.

The Engineering Process – Entry Condition



The entry condition for the design cycle includes:

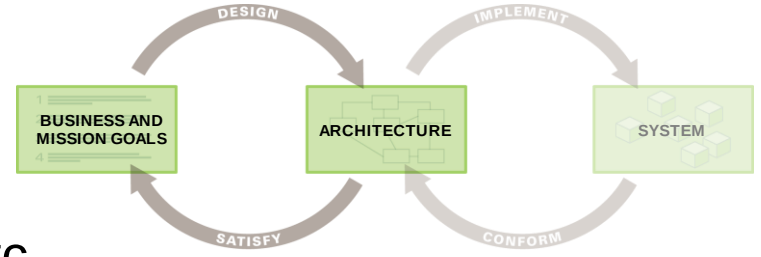
- The availability of defined, prioritized, and measurable quality attribute requirements
- A measurable definition of the important quality attribute requirements
- At least five high important quality attribute scenarios

During the design cycle these quality attribute scenarios will be

- Refined to more specific cases
- Transformed into an architecture design supporting these scenarios
- Annotated with a list of known risks showing the degree of support by the current architecture

The Engineering Process – Design Cycle

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



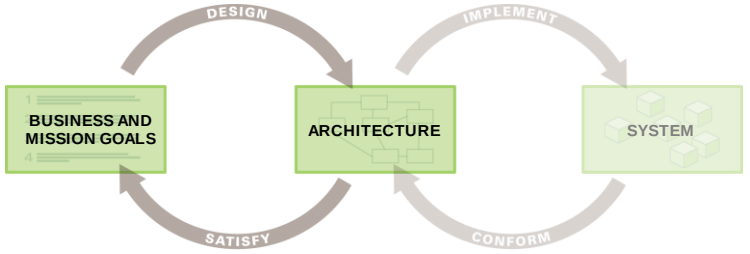
The methods used in the design cycle are a combination of a

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

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

The Engineering Process –
Do Not Plan a Separate Documentation Task

Documentation and reviews are not separate tasks. They are part of the design process.



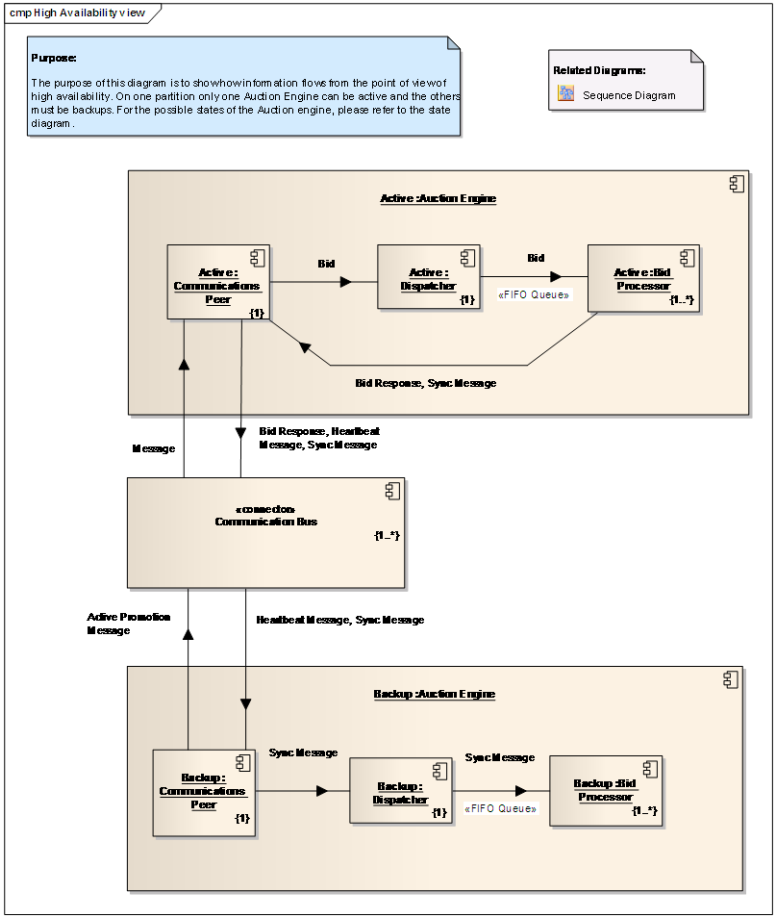
At no point in time is there a need to plan separate tasks for documentation and reviews during the design of the architecture.

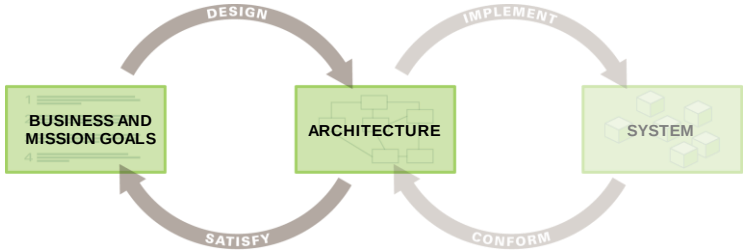
Documentation and reviews are done continuously while designing.

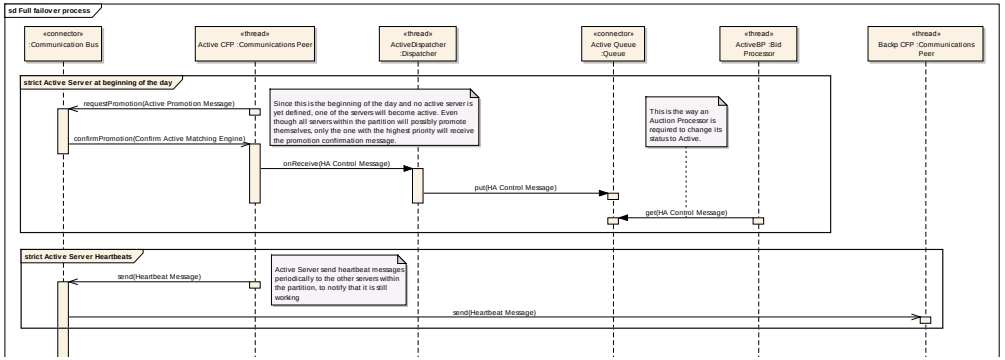
- Designing an architecture means providing evidence that the quality attribute requirements are supported.
- Evidence is best provided in written form (documentation).
- Written evidence can easily be evaluated by peers not involved in the project.

Providing evidence results in high confidence that the designed architecture is appropriate.

The Engineering Process – Documentation as Evidence





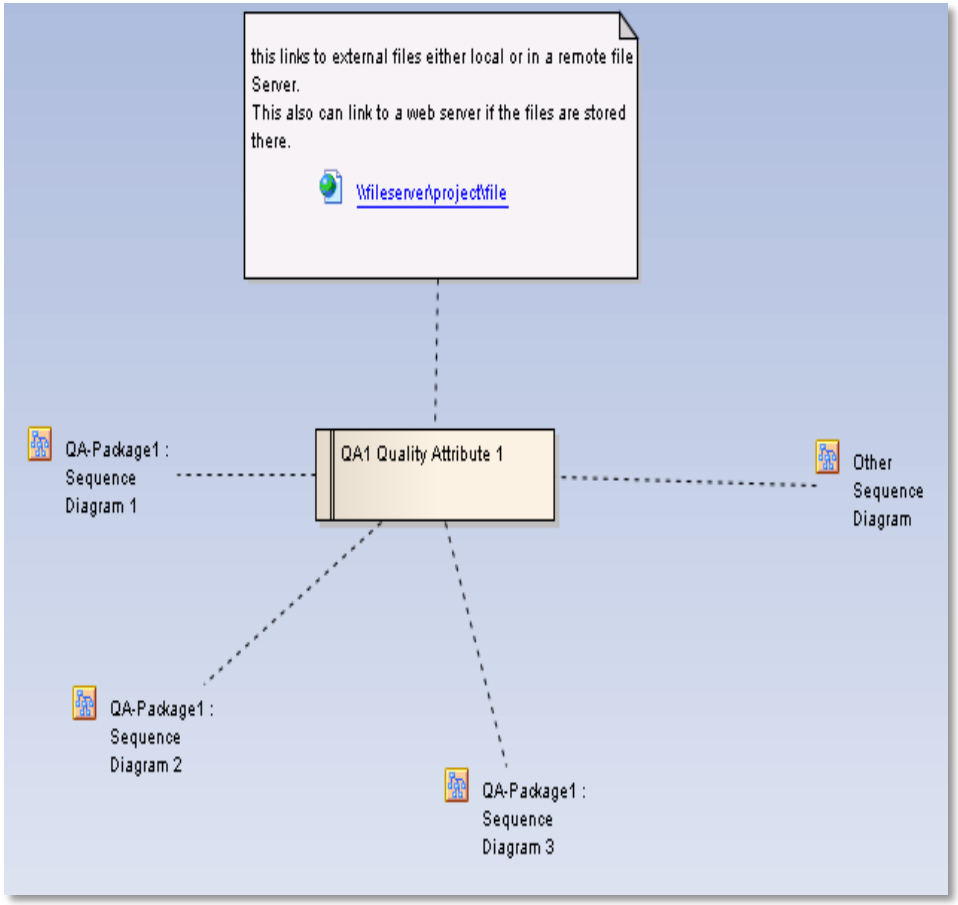


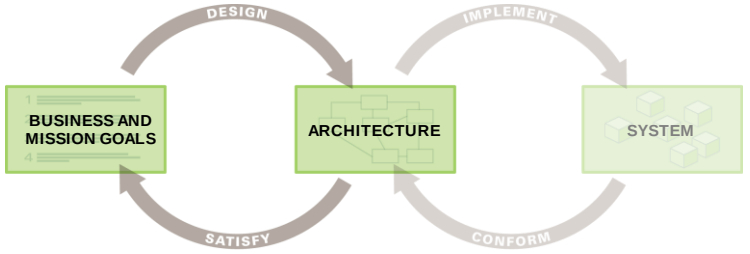
Create diagrams that show how the architecture supports the scenarios.

UML modeling tools are appropriate.

UML models are NOT for stakeholders other than architects and developers.

The Engineering Process –
Organize Documentation Around Scenarios



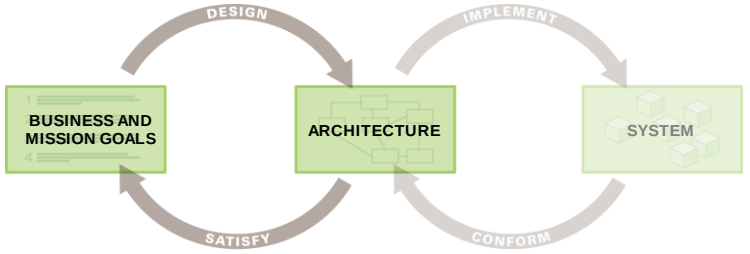


Organize the documentation in such a way that everything that supports a quality attribute scenario can be found quickly.

Having a tool that supports the linking of documents is very helpful.

The Engineering Process –
Continuously Review by Peers

Review the progress made in the architecture at least every two weeks.



Use ATAM techniques for the peer reviews.

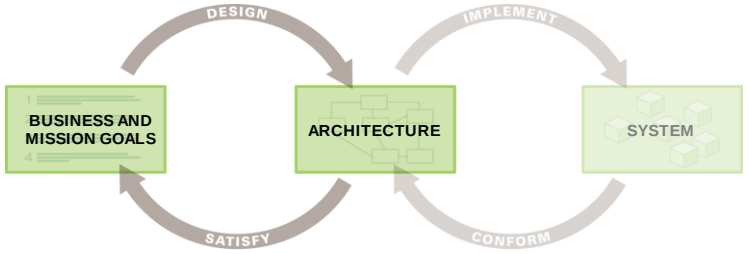
Every review takes two to three hours and requires the presence of one or two questioners, one scribe, and the architect(s).

- Two scenarios are picked for the peer review.
- A list of architecture approaches is elicited and written down by the scribe – if not already existent.
- Every problematic architecture decision is written down as a risk.
- Every unknown answer is written down as a to-do item.

If there are unmitigated risks and/or open to-do items, then the review will be repeated.

The Engineering Process –
Risk Items per Scenario as Progress Measure

Stakeholders understand scenarios and risks.



Do not try to show progress to the project manager and other stakeholders by showing them how your UML diagrams evolved. They will not see any progress.

Stakeholders understand their quality attribute scenarios and they understand risks.

To show progress, show the quality attribute scenarios with the attached list of risks and to-do items from the peer reviews.

Over time this list shrinks and therefore shows progress to the project manager.

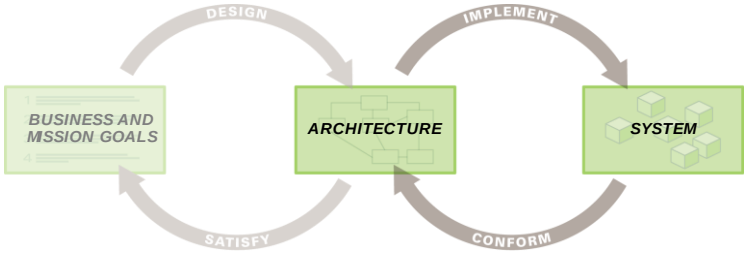
Polling Question #2

What architecture practices have you used? (check all that apply)

- Architecture design driven by quality attribute requirements
- Documentation of multiple views to guide architecture design
- Architecture evaluation
- Documentation of module views to guide implementation
- Architecture conformance

The Engineering Process – Implementation Cycle

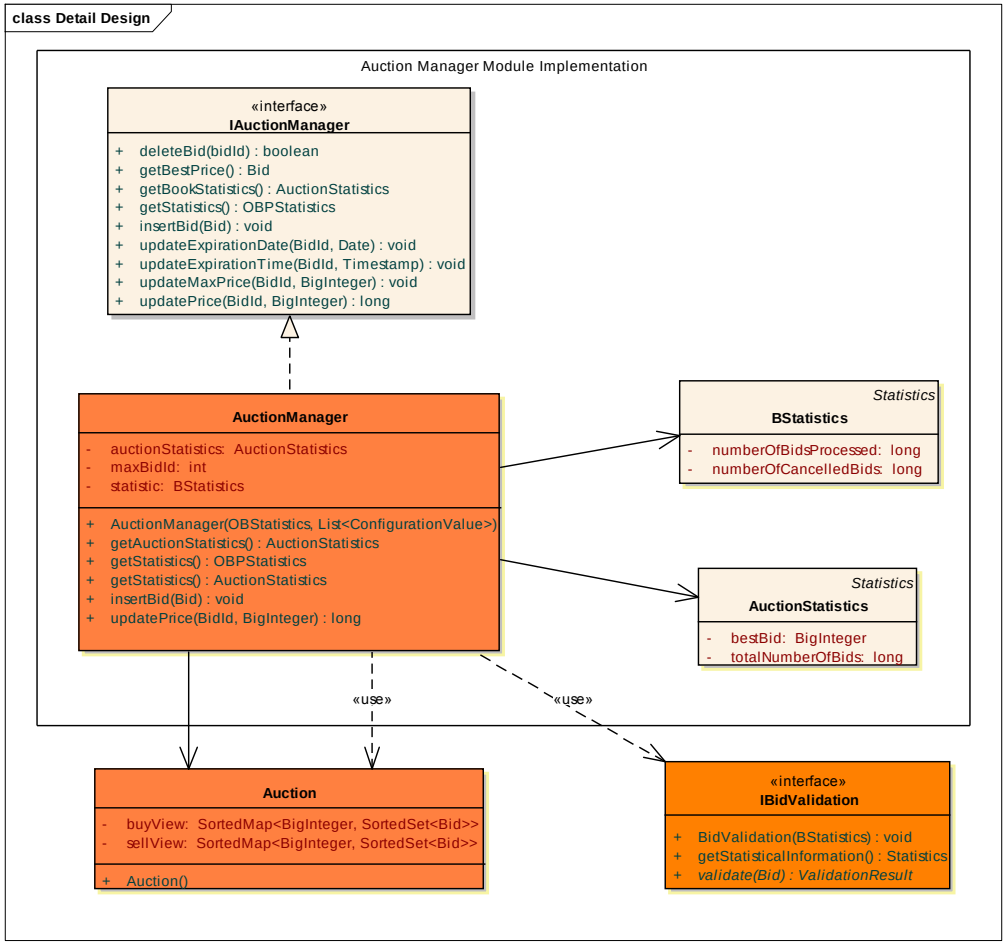
The implementation cycle is centered around establishing communication between architects and developers.

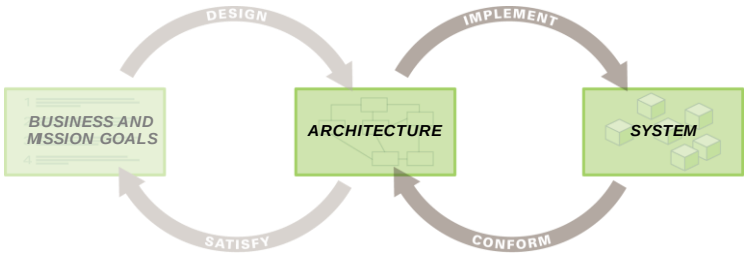


An architecture that shows how well the scenarios are supported is not enough to be implemented correctly.

- 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.

The Engineering Process – Module Views



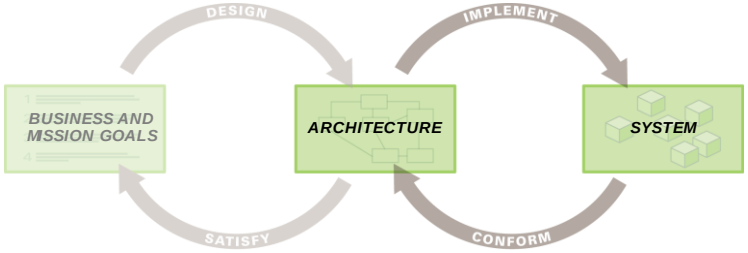


Every module (implementation unit) has a description showing the module's:

- Interface
- Parts (if already known)
- Allowed usage of other modules

The Engineering Process –
Active Reviews for Intermediate Designs (ARID)

An active design review (ARID) established understanding of the architecture quickly and efficiently.

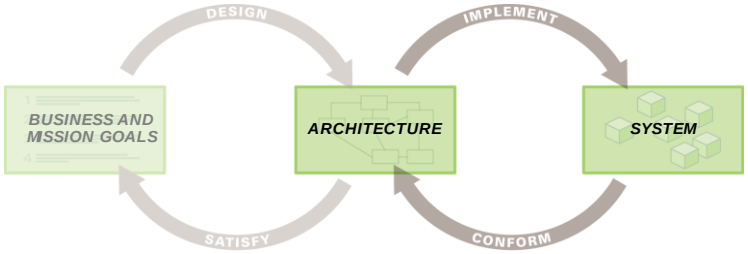


Have a two to three hour workshop with the developers and architects to transition from developing the architecture to using the architecture.

- Provide the documentation to the developers.
- Present one or two use cases to the developers.
- Ask developers to sketch what they think they have to do to implement the use case.
- Developers can ask architects any question. Architects note the questions, to further improve the documentation.
- After two hours the developers describe what they would have to do.
- Architects note any discrepancy, to improve the architecture.

The Engineering Process –
Conformance Reviews

Continuously check that the implementation follows the architecture (e.g., every second week).



Have a two to three hour review workshop with the developers and architects. Developers provide documentation (generated by tool) showing:

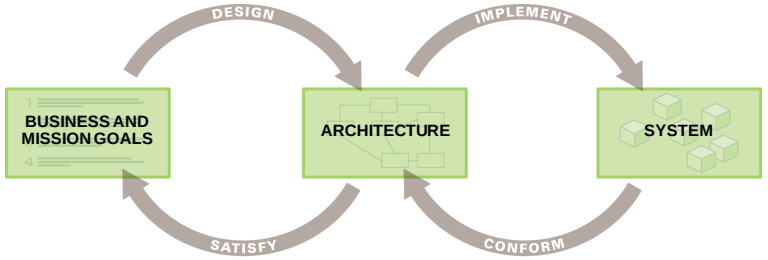
- Classes / functions that implement the module – the module classes.
- Uses relations the module classes have with other classes
- Module classes in the package / sub-system hierarchy.

Architects and developers compare the generated documentation against the architecture document.

For every discrepancy, it is decided to either fix the architecture or the code.

The Engineering Process – Iterations

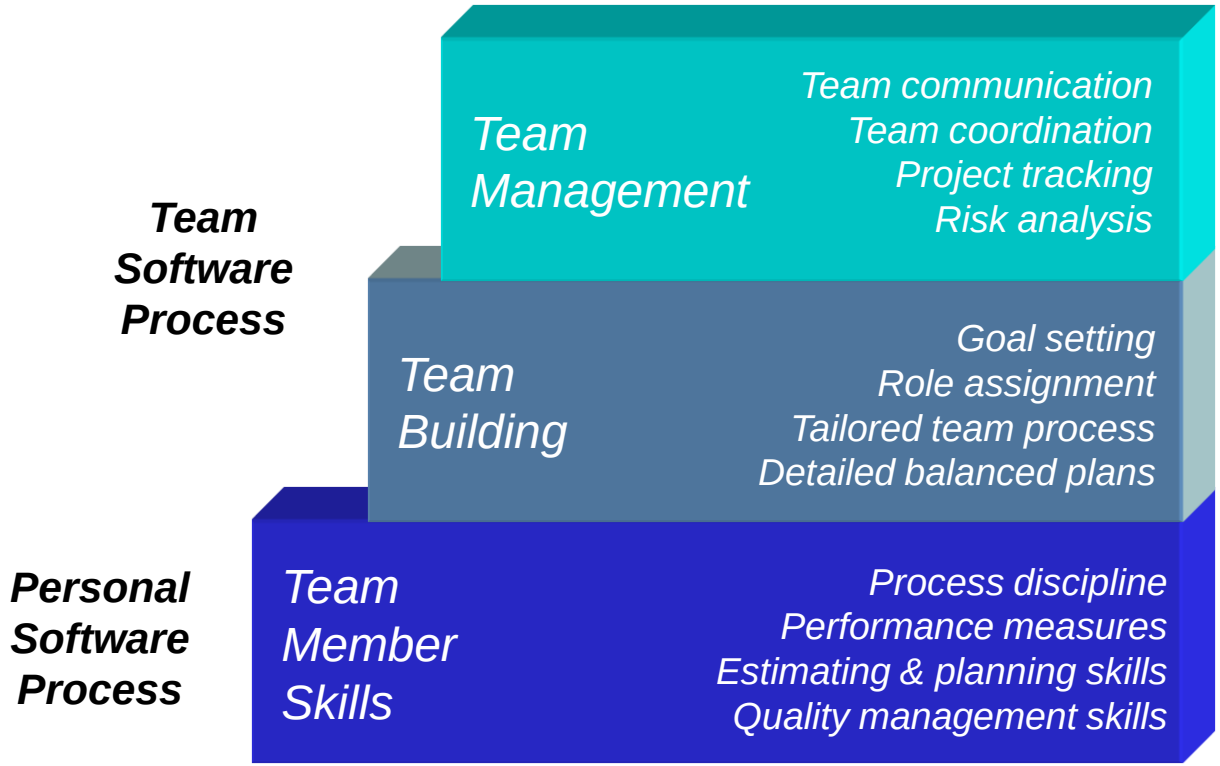
Design and implementation cycles can be run one after the other or in alternating iterations



The duration of an iteration depends on the nature of the project. Iterations should be between four to eight weeks.

The Management Process –
Team Software Process (TSP)

TSP builds high-performance teams from the bottom-up.



The Management Process –
Personal Software Process (PSP)

The Personal Software Process (PSP) discipline is essential to TSP.

Individual developers, including architects

- select, adapt, or define their own processes
- estimate, plan, and track their work based on these processes
- take their own measurements and use this data to manage both the schedule and the quality of the work
- constantly improve their processes based on results and the data

PSP training is the basic training for these skills.

Polling Question #3

What PSP / TSP training have you participated in? (check all that apply)

- *PSP Developer Training*
- *TSP Team Member Training*
- *TSP Team Leader Training*
- *TSP Coaching*
- *TSP Executive Seminar*

The Management Process –
TSP Measurement Framework

Schedule

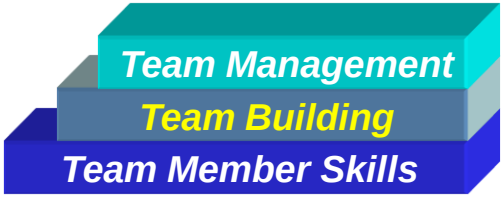
Effort

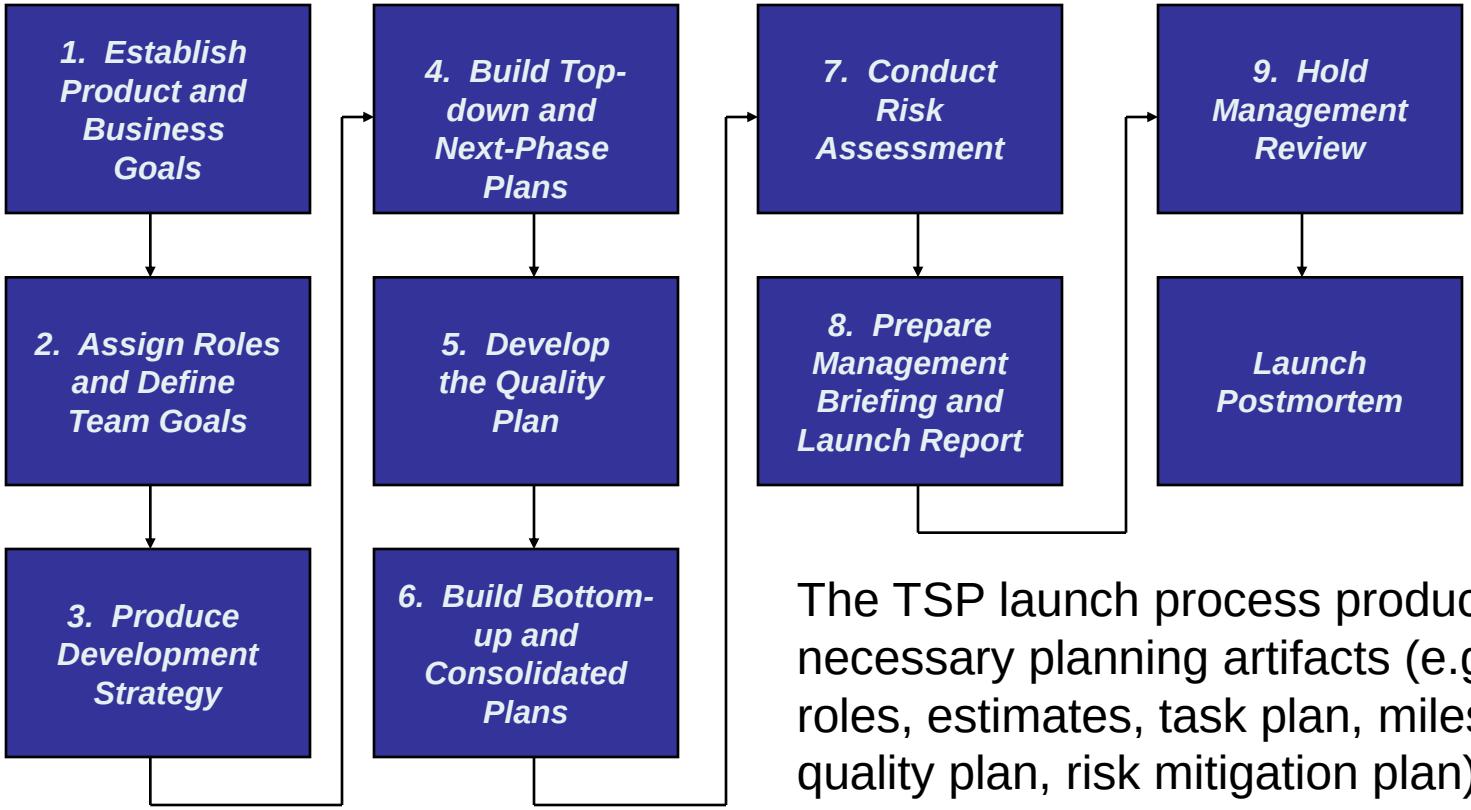
Size

Quality

- *Four base measures*
- *Apply to all processes and products*
- *Estimates are made during planning*
- *Directly measured by team members while working*

The Management Process – TSP Launch

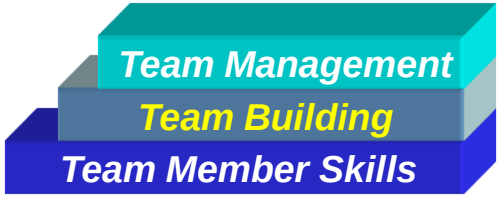




The TSP launch process produces necessary planning artifacts (e.g., goals, roles, estimates, task plan, milestones, quality plan, risk mitigation plan).

The most important outcome is a committed team.

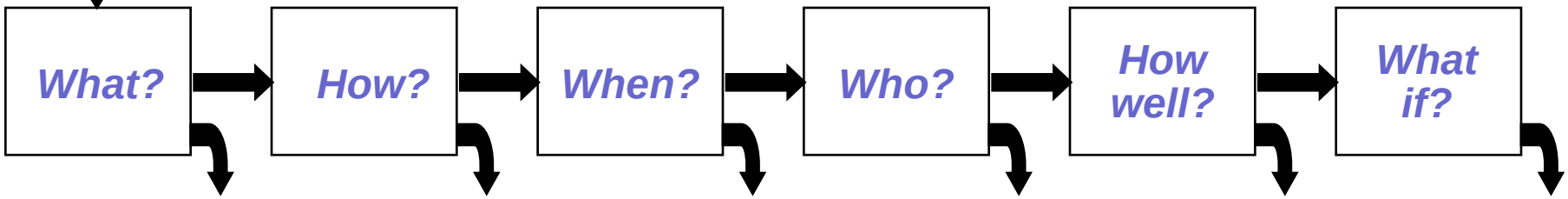
The Management Process – TSP Launch Artifacts



Business needs
Management goals
Product requirements

A TSP team has a lot to do at startup.

The TSP coach is critical to a successful start.



Team goals

Conceptual design

Planned products

Size estimates

Team strategy

Team process

Task hour plan

Schedule plan

Earned-value plan

Team roles

Task plans

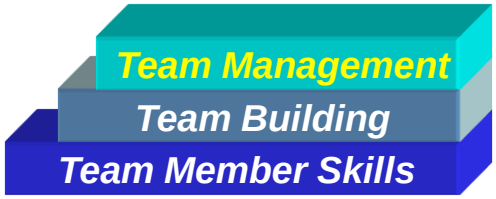
Detailed plans

Quality plan

Risk evaluation

Alternative plans

The Management Process –
TSP Launch and Coaching Support

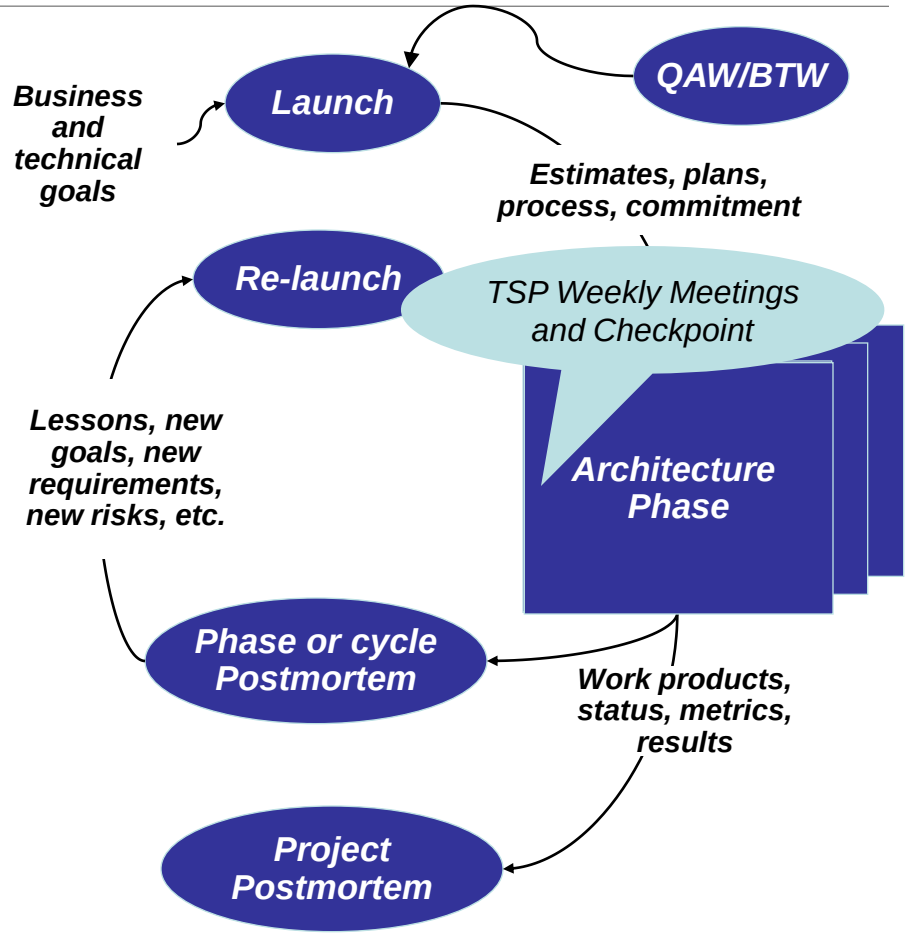


TSP supports an iterative or cyclic development strategy.

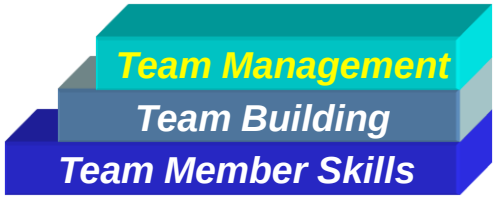
TSP can be introduced starting at any phase or any cycle.

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.



The Management Process –
Weekly Team-working Framework



Team member's track plans daily

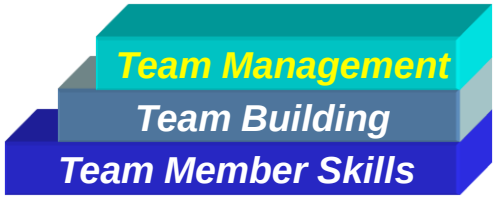
Bob	Tom	Sally	John	Tyra	Pablo	Gloria	Abhinav

Team plans are consolidated and reviewed weekly

Galileo	Protec	RSM

Management reviews are typically monthly

The Management Process – Weekly Status



Team members meet each week to assess progress.

- Role managers present evaluation of the plan and data
- Goal owners present status on product and business objectives
- Risk owners present status on risk mitigation plans and new risks
- Team members present status on their plans

Plan deviations are addressed each week.

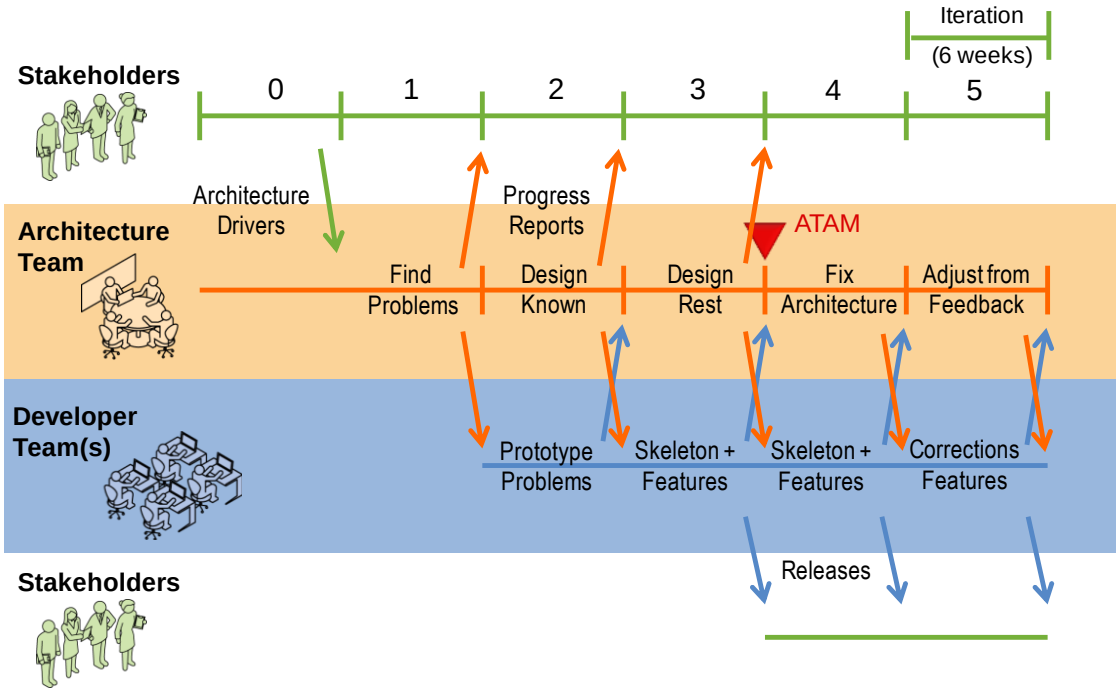
Significant deviations, such as new requirements or a staffing change would trigger a replan.

Performance Data Reviewed

- Baseline Plan Value
- Plan Value
- Earned Value
- Predicted Earned Value
- Earned Value Trend
- Plan Task Hours
- Actual Task Hours
- Tasks/Milestones completed
- Tasks/Milestones past due
- Tasks/Milestones next 2 weeks
- Effort against incomplete tasks
- Estimation Accuracy
- Review and Inspection Rates
- Injection Rates
- Removal Rates
- Time in Phase Ratios
- Phase and Process Yield
- Defect Density
- Quality Profile (QP)
- QP Index
- Percent Defect Free
- Defect Removal Profile
- Plan to Actual Defects Injected/Removed

Managing the Engineering Process – Iteration 1

A possible sequence of iterations for one architecture and one developer team …

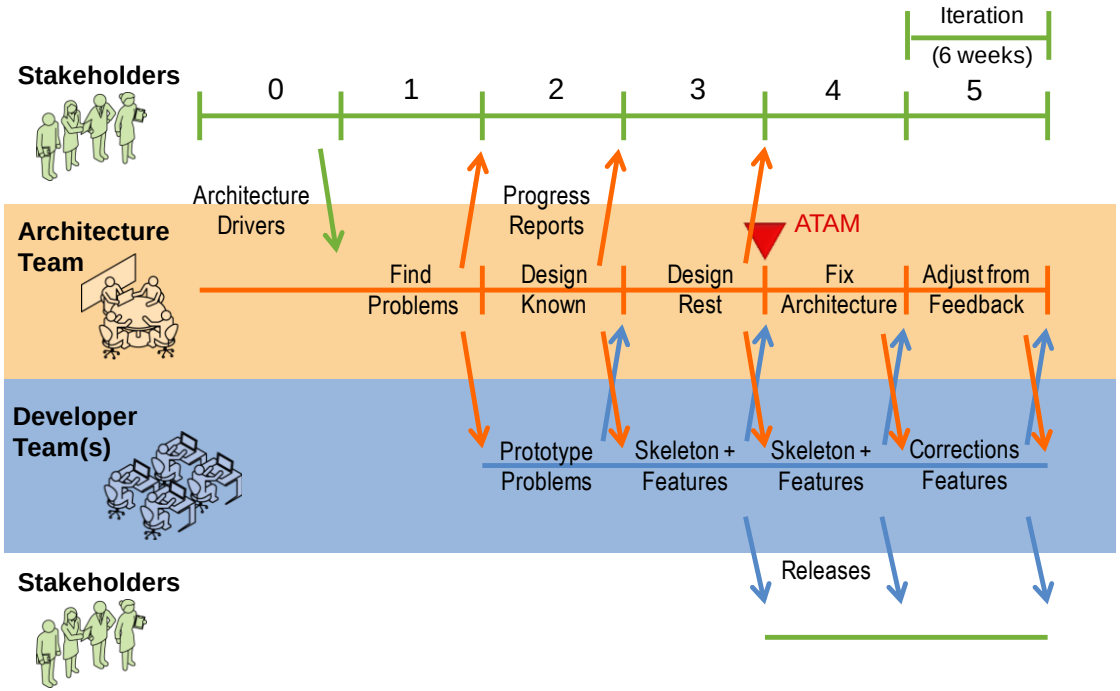


Iteration 1

- Architects - Use the five most important quality attribute requirements to design the architecture in enough detail to separate what is known from what is unknown.
- Developers – Prepare development environment.

Managing the Engineering Process – Iteration 2

Design and implementation cycles continued …

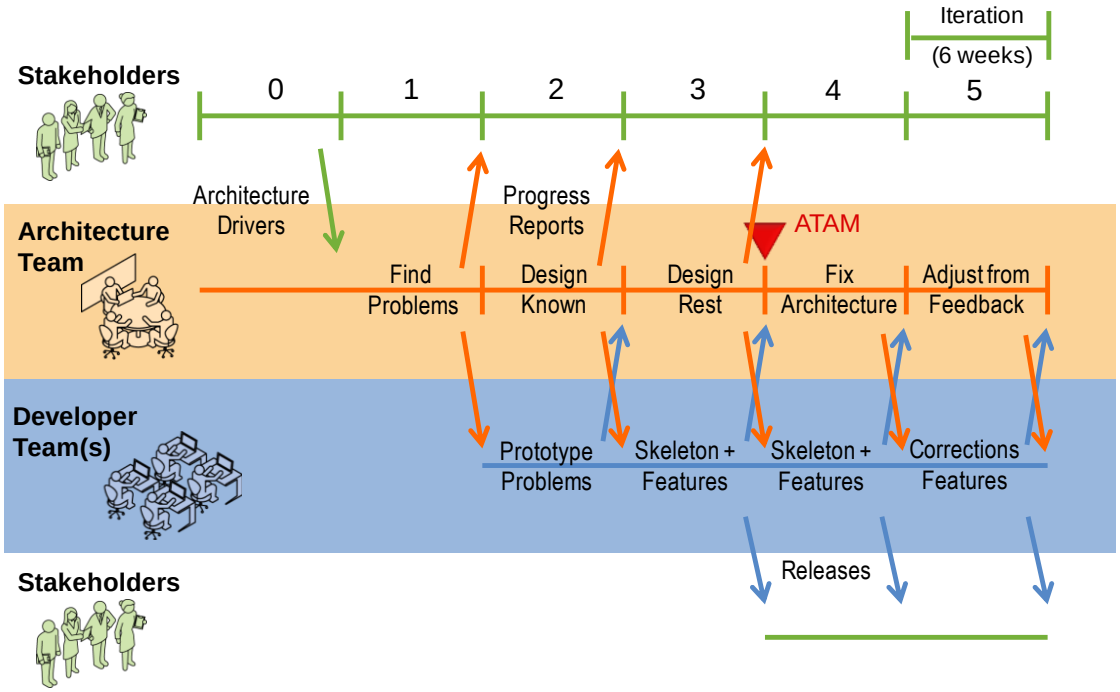


Iteration 2

- Architects – Describe the well understood parts of the architecture in enough detail to be implemented.
- Developers – Create prototypes that provide answers to identified architecture problems.

Managing the Engineering Process – Iteration 3

Design and implementation cycles continued …

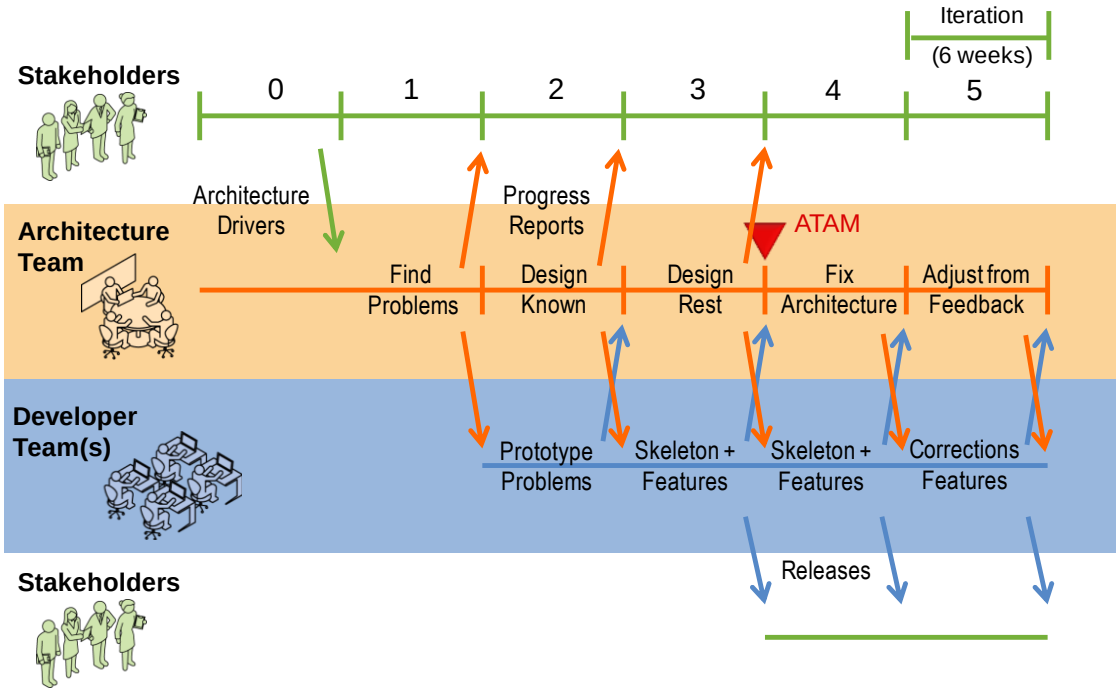


Iteration 3

- Architects – Design the rest of the architecture in enough detail to be implemented, using the results of the prototyping.
- Developers – Build a "skeleton" system of the known parts of the architecture. Implement one simple feature in the skeleton system (1st deliverable to be shown to the stakeholders).

Managing the Engineering Process – Iteration 4

Design and implementation cycles continued …

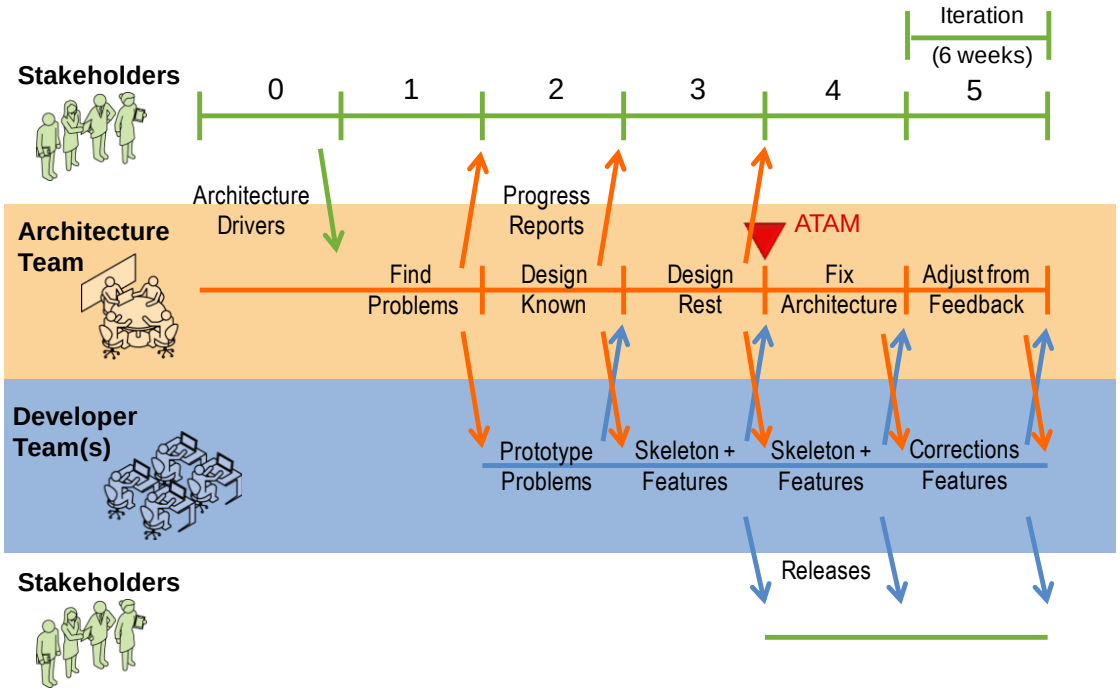


Iteration 4

- Architects – Perform an architecture evaluation ATAM and make necessary adjustments in the architecture.
- Developers – Implement skeleton of the remaining system.

Managing the Engineering Process – Iteration 5

Design and implementation cycles continued …



Iteration 5

- Architects – Architecture tasks are fading out now. One architect continues to solve discovered issues.
- Developers – Implement the next feature in the skeleton system (2nd deliverable to be shown to the stakeholders).

This type of iteration now repeats until all features are implemented.

TSP and ACE – Principles in Common

On the surface, TSP and ACE are very different disciplines.

- TSP is a self-directed management and measurement process.
- ACE provides sophisticated technical development methods.

Common principles allow TSP and ACE to work well together.

- emphasis on business and quality goals
- emphasis on engineering excellence
- emphasis on defined processes and process discipline
- emphasis on teamwork

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!

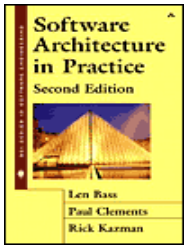
Architecture can be coached.

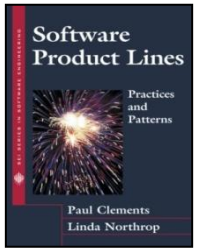
TSP provides a disciplined framework for measuring and managing any structured intellectual activity.

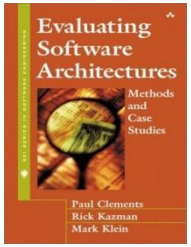
Architectural awareness helps to structure the team and the work in addition to the product.

Architecture Training to Support the Engineering Process

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





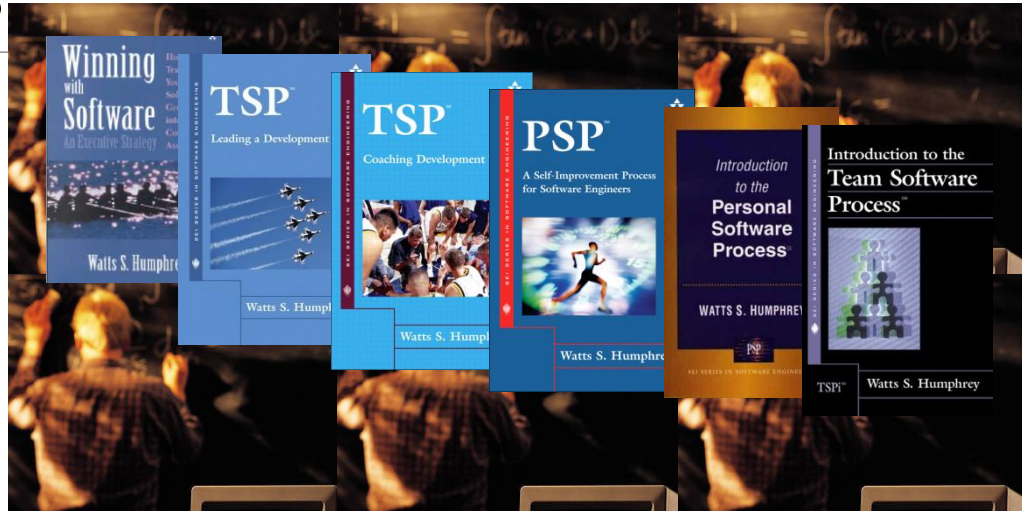


✓ : required to receive certificate / certification

*: available through e-learning

PSP[SM] and TSP Training to Support the Management Process

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)

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)

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.

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.