Introducing Software Architecture Development Methods into a TSP-Based Development Company

Humberto Cervantes (UAM-I)
Isaac Martinez Aceves (Quarksoft)
Jaime Castillo (Quarksoft)
Carlos Montes de Oca (CIMAT/Quarksoft)

Outline

• About the authors and Quarksoft
• TSP and architecture
• Quarksoft’s TSP and architecture
• Introducing architectural development methods into Quarksoft’s TSP
• Lessons learnt and conclusion
About the authors

- Humberto Cervantes
  - Professor and researcher at Universidad Autónoma Metropolitana – Iztapalapa
- Isaac Martinez Aceves
  - Software architect at Quarksoft
- Jaime Castillo
  - Corporate training leader at Quarksoft
- Carlos Montes de Oca
  - Professor and researcher at CIMAT
  - On sabbatical at Quarksoft, innovation department leader

About Quarksoft

- Quarksoft is a leading software development company in Mexico City
  - Founded in 2001
  - Around 280 people distributed over 3 sites
- Rated at CMMi level 3 since 01/2006
  - Development based on the Team Software Process (TSP)
TSP Overview

1. TSP Project Structure
   - Requirements phase
   - High Level Design phase
   - Implementation phase
   - Integration and test phase

2. TSP and PSP
   - Each component is developed individually using PSP

Products:
- Software Requirements Specification
- System tests and user manual outline
- System Design Specification
- Performance and integration test specs.
- The system's components designed and built using PSP. Unit and product test specs and draft documentation
- Completed product
- Final documentation

Component plan
Detailed Design (DLD)
DLD Review & Inspection
Coding
Code Review & Inspection
Compile
Unit test
Post Mortem
TSP and Software Architecture

- TSP does not give detailed guidance with respect to architectural concerns
  - Quality attributes
  - How to design the architecture
  - What is the granularity of a “component”
  - No “architect” role (the closest may be Design and Implementation Managers)
  - No concept of architectural evaluation (only HLD inspection).

Development context at Quarksoft

- Quarksoft develops custom software for customers in several different sectors
  - Insurance, Manufacture, Telecommunication, Retail, Government, Healthcare

- Some particularities
  - Typically, Quarksoft customers require the company to provide a cost and time estimate very early, before the project is approved
  - Requirements are completely specified and then become contractual
  - A core team is usually designed at the beginning of the project (leader, architect, some engineers) and then development may be performed by teams that are spread among the different sites
  - The company is currently in a growth phase
TSP at Quaksoft

**Quarksoft’s TSP Project Structure**

- **Launch**
  - Preliminary analysis
- **Relaunch**
  - Requirements phase
  - High Level Design phase
  - Implementation and integration phase
- **Relaunch**
  - Test phase

**Project Time and Cost Estimate Based on High Level Requirements**

- All requirements are specified in one or more cycles (~6 weeks on average)
- High level design is completed in one or more cycles. Ideally **ALL** components are specified. Sometimes system is re-estimated.
- The system is built incrementally over several cycles. Unit and integration tests
- System tests
- Final documentation

Software architecture at Quarksoft

- **Before this project started, a 2-month study was conducted to understand the state of the practice**

- **The study involved**
  - Reviewing process scripts, artifact templates, checklists and other process artifacts
  - Reviewing existing HLD documents
  - Observing a team in the HLD phase
Study results

The study uncovered many “common” issues related to software architecture:
- Business goals specified inappropriately (too vague)
- Quality attributes specified inappropriately (not measurable, not aligned to business goals)
- Poorly documented architecture designs (not always UML, huge diagrams, too high level, underspecified component interfaces)
- Design focused on satisfying functional requirements
- Excessive focus on technology (lack of pattern usage)

Study results (2)

Other issues were more specific to (Quarksoft’s) TSP
- Process scripts and templates did not provide guidance to help capturing and documenting quality attributes and perform design in a systematic way
- HLD inspection, which is performed by team members, took place too late in the HLD phase

Also some issues were specific to Quarksoft’s context
- Preliminary analysis constrains development time and cost
- Requirements and HLD phases are performed sequentially
- Lack of architects and available ones lack strong theoretical foundations on software architecture
Proposal

- To overcome these problems, a strategy focused on introducing architecture development methods was defined.

- The original idea was to directly introduce SEI’s methods: QAW, ADD, VaB and ATAM:
  - An initial study led us to conclude that we could not introduce them directly, the methods had to be adapted (and simplified) to the particular problem’s context.
  - Furthermore, they had to be introduced into Quarksoft’s TSP.

Method introduction overview

- Architecture development methods are introduced in the Requirements (REQ) and High Level Design (HLD) phases of TSP.

- HLD activities are divided in two:
  - Architectural design
  - Other HLD activities
Architectural requirements

The goal is to produce a list of prioritized quality attributes which are documented in the SRS document.

Requirements method

- Standard QAW was not chosen primarily because of the perceived difficulty of involving customers in scenario related activities
  - The essence of QAW which involves identifying quality attribute scenarios aligned to business goals is maintained
- Quality attribute related activities were integrated inside standard requirements activities of the existing process

Obtaining quality attribute categories from interviews
Deriving quality attribute categories from business goals
Identifying metrics
Identifying “raw” scenarios
Studying rationale and impact
Specification of scenarios
Revision using checklist
Prioritization according to: importance to customer and difficulty of implementation
Requirements method and TSP

- Software architects already participate in project requirement activities as other project members
  - The idea was to maintain the architects participation but to focus their activities on quality attributes

- Process elements created to support the method
  - Quality attribute process script
  - Quality attribute template
  - Quality attribute checklist

- Changes in existing requirements script

Architectural design

The goal is to produce a documented architectural design which has been evaluated by other architects. This design must both satisfy quality attributes and serve as a guide during implementation.
The design method that was introduced is ADD

- Iterative design method, starting with domain model
- Not only “conceptual” design (based on patterns and tactics), but also technological choices are made during design iterations

Integration with TSP

- One iteration is specifically focused on defining the list of components that will be developed independently using PSP in the implementation phase (work assignment)
- Design time has to be planned at the beginning of the HLD phase
- Process elements: Design script, changes in HLD script
Documentation method

Documentation method and TSP

- Documentation is based on the VaB templates, but limited to a number of views to ease migration from 4+1 and to help in planning activities
  - Logical
  - Physical
  - Runtime
  - Work assignment

- Process elements: Documentation Script, View Template, View Checklist
Evaluation method

Evaluation method and TSP

- A scenario-based evaluation method based on ACDM was introduced
  - Short evaluation (1/2 to 1 day)
  - No driver discovery (as opposed to ATAM), use of an “evaluation package” composed of drivers + views
  - Evaluation committee is composed by other architects from the company

- Integration with TSP
  - Defects identified during evaluation are collected
  - Risks can be used in next re-launch
  - Evaluation script
One of the goals is to define an external specification for all the work-assignment components that will be developed independently during implementation using PSP.
Introduction strategy

- The proposed introduction strategy considers starting with changes in requirements
  - Start with new projects
  - “Just in time” training: before REQ and before architectural design

- Hopefully, the introduction of changes in HLD will be smoother for these projects
  - They start with clarity with respect to drivers

Evaluating the results

- Only one pilot project so far...
  - The collected data does not allow conclusions to be made yet but the project artifacts show significant differences with respect to what was observed in the initial study

- Metrics that we will be studying
  - Defect data from evaluation will be a very valuable source of information
    - Quantify the benefits of the approach
    - It can help focus training activities
  - Time data is also important
    - Greater time in architectural design should show reduction in integration and (system) test time
Lessons learnt

● Requirements
  ▪ Business goals must be correctly specified
  ▪ Metrics to specify quality attributes may be hard to identify

● Design
  ▪ The architect must really have clarity with respect to architectural drivers before starting design
  ▪ A bridge must be made between “conceptual” (pattern-based) design and frameworks
  ▪ A work assignment structure is fundamental to guide development and also very helpful for re-estimation
  ▪ Best ways to use case tools to support design must be identified
  ▪ Estimating design time is not straightforward

Lessons learnt

● Documentation
  ▪ Documentation activities take a long time so moving design from CASE tools to documents must be straightforward

● Evaluation
  ▪ Doing evaluations in a short time is difficult but longer evaluation can be a logistics challenge
  ▪ It can be hard for the architect to effectively communicate drivers and design decisions in a short time
  ▪ Architects are not automatically good evaluators
  ▪ Defect data gathered from evaluation is extremely valuable
Lessons learnt

- The introduction of architecture development methods into Quarksoft’s TSP has required considerable time and work
  - There is an impact on several process elements: Scripts, Templates, Checklists (mainly from REQ and HLD)
  - The introduction of development methods must also consider training and technology issues
    - A complete course covering the methods has been created
    - Software engineers must also receive some training as they participate in related activities

- Other aspects must be considered
  - Integration with CMMi for example: Decision Analysis and Resolution (DAR)

Lessons learnt

- Preliminary analysis imposes many constraints on software architecture

- A subset of architecture tasks need to be performed during this phase to improve estimates

- The amount of information and limited time make this difficult
Conclusion

• Architecture development methods can be integrated into TSP without requiring significant changes to the process

• However, the biggest challenges are at the organization level
  ▪ Process elements changes, training development, technology…
  ▪ A gradual introduction strategy may be undertaken

• The data collection framework of TSP should provide us with data that will help to understand the benefits of the approach in a measurable way

Thank you

• Contact:

  Humberto Cervantes
  hcm@xanum.uam.mx