

Requirements and Analysis: Techniques and Tools

The Requirements Discipline

▪ Requirements Drive Development: A Use Case-driven Process

As stated in previous posts and in articles like [Real World Development Practices: RUP and XP](#) , I apply much of Craig Larman's UP style and its emphasis on rightsized, "essential" use cases, which then collectively act as a lynch pin that links together the disparate disciplines of Business Modeling, Requirements, Analysis and Design, Implementation, Test and Project Management. Furthermore, achieving success with use cases is more difficult than it first appears, and many pitfalls in usage await the inexperienced practitioner. Consistent application of the techniques espoused by Alistair Cockburn's de facto standard for [specifying use cases and structuring them in relation to goals](#), which provides a repeatable, traceable discipline for use case development and maintenance.

▪ Executable Requirements: Aligning Requirements and Development

These days I particularly like the idea of 'Executable Requirements' (XR) to capture requirements. This approach has the benefit of not only enabling the Pull method described above but they also ensure that software developed matches the specifications provided. XR basically provides a mechanism where a requirement is captured in a 'pass/fail' style using an Excel or HTML table to define the requirements. The power of this approach is that it not only

moves requirements out of the fuzzy, prose style that can plague use cases (and which is why use cases have so many sections) but also allow a team to automate a series of tests that demonstrate that a requirement has been 'fulfilled'.

For those of us with a testing orientation we can immediately see the opportunity to regress through all of our tests every iteration and ensure that new changes don't break old functionality. There's a lot to this subject and something that I'll update on more in the future but there are some good reference sources for this such as the [Fitnessse wiki](#) and Ward Cunningham's [Functional Integration Testing \(FIT\) Framework](#).

- **Managing Risk and Non-functional Requirements (ATAM, EVO)**

Addressing Non-Functional or Supplementary Specifications is often a neglected component of software development. Notable references in this area are Tom Gilb's iterative [EVO](#) method, which emphasizes full and careful definition of non-functional requirements (which Gilb calls "attributes" leveraging his Planguage approach) and [SEI's ATAM](#) (Architecture Tradeoff Analysis Method) methodology. Documentation of all significant architectural decisions – a component of the ATAM approach – as a key mechanism for reasoning about and justifying choices between architectural options. This fits well with leveraging risk analysis as a major driver of iteration plans.

- **Early, Continuous Delivery of Business Value: Complementing the Risk Driver**

The agile methods complement UP by providing an important emphasis, not only on risk reduction, but also on the [early and continuous delivery of business value](#). Hence, a full iterative development discipline has two drivers: delivery of useful functionality and management of risks. The use case-

driven approach, when combined with non-functional drivers and the dispatching of work into developer tasks provides tangible evidence of progress to the business at each iteration's end. (See some of the [XP](#), [EVO](#), and [FDD](#) links for further details.)

The Analysis Discipline

▪ From Use Cases to Developer Tasks

The Larman method takes analysts and designers through a series of simple intermediate steps leading up to operation contracts on a system or service level interface. In accordance with [Agile Modeling](#) [below], intermediate artifacts need neither be formally developed nor maintained if the ceremony level of the process does not warrant it. I also believe strongly in a “pull”-driven approach to developer task definition, a key element in [Lean Programming](#).

▪ Applying Analysis Patterns to Streamline Design

I encourage analysts to leverage [Martin Fowler's Analysis Patterns](#), rather than reinvent the wheel. This emphasis provides synergy with the product line process mentioned later, and also opens the analysis up to alignment with standardized vertical models such as well defined reference models (e.g. Insurance Application Architecture). Another useful source of such patterns is [Penker and Eriksson's book](#).

Discipline by Discipline: Requirements

As many who follow my blog entries and have read my articles know, I use the Unified Process as framework to manage projects and programs. While the phases of the UP (Inception, Elaboration, Construction and Transition) are powerful ways to manage the risk and narrow the [‘cone of uncertainty’](#) of a project, I find the disciplines within the Unified Process as useful containers for ensuring roles are established and that artifacts are being developed that will support the project.

However, beyond the phases and disciplines I find most of the artifacts and activities as too abstract for effective application in most real world projects. Instead, I mix in a series of techniques that I have applied successfully and found round out the details of each of the disciplines with RUP. This first article focuses on the top of the “V” model, Requirements and Analysis.

[Read More](#)

A Unified Approach to Agility (Article)

With the increasing interest in Agile techniques such as Scrum and XP, I often come across clients and project managers assuming that these approaches alone are sufficient to ensure the success of their projects. In actuality, the [Agile Principles](#) are really a value system that help contribute to

effective behaviors on a project. None of the agile techniques recommend dispensing with the well defined practices that govern effective project implementations such as risk, scope and change management (amongst others). In fact most of the Agile techniques found in current literature are intended to work within existing frameworks and metamodels, without which your projects won't succeed.

When I develop project plans and teach project management approaches, I frequently turn to the Metamodel offered by the [Unified Process](#). What I like specifically about the Unified Process is that it breaks a project into four phases (**Inception, Elaboration, Construction and Transition**) that have clear entry and exit criteria that are easy to manage against. In addition the phases are well defined and relatively intuitive to most people (**Inception** involves scoping and structuring the project, **Elaboration** focuses on de-risking the project and developing an Architecture, **Construction** emphasizes the rapid development phase of the project and **Transition** focuses on readying the application for deployment).

The UP also contains a number of useful '*disciplines*' which reflect major workstreams in a project lifecycle. **Business Modeling, Requirements, Analysis and Design, Implementation, Test and Deployment** ebb and flow across the project lifecycle while **Project Management, Configuration/Change Management and the Environment** disciplines are focused on supporting the lifecycle in its entirety (these latter are found in the IBM version of the Unified Process called the [Rational Unified Process](#)).

The popularity of the Unified Process is reflected in its evolution into a number of forms including the [Agile Unified Process](#), [Enterprise Unified Process](#) and even the [Oracle Unified Process](#). IBM recently released an open source version called [OpenUP](#) which is based on its popular Eclipse [Process Framework](#).

Over time I've come across a number of agile/lean techniques that support the disciplines I mentioned above and enhance these disciplines to make them more effective. Over the next few postings I'll offer a walk through on a discipline by discipline basis on each of these techniques.

Discipline by Discipline

[Requirements and Analysis](#)

Design

Development

Testing

Project Management

A Unified Approach to Agility

With the increasing interest in Agile techniques such as Scrum and XP, I often come across clients and project managers assuming that these approaches alone are sufficient to ensure the success of their projects. In actuality, the [Agile Principles](#) are really a value system that help contribute to effective behaviors on a project. None of the agile techniques recommend dispensing with the well defined practices that govern effective project implementations such as risk, scope and change management (amongst others). In fact most of the Agile techniques found in current literature are intended to work within existing frameworks and metamodels, without which your projects won't succeed.

[Read the article](#)

Project Iteration Routemap

The Iteration Route Map is a tool that identifies what functionality will be delivered in each iteration of a project. As the name suggests it acts as a map that project stakeholders can reference in order to anticipate how the application will develop throughout its lifecycle. The Iteration Route Map is primarily used by the **architect** and **development team** to manage *analysis and design* and *implementation* activities throughout the project but every team member and stakeholder will find value in reviewing it once completed. The template comes with instructions and examples.

Crowdsourcing your PMO Governance Model

Crowdsourcing is a relatively new term that describes the act of taking a task traditionally performed by an employee and outsourcing it to an undefined, generally large group of people or community. In this presentation, that was part of the **2009 PMO Symposium**, an innovative approach to developing an organization's PMO Governance model was offered that leveraged crowdsourcing and Wikis to create a self-sustaining governance model that was easily scalable and adapted readily to changing organizational needs.

Enterprise Implementation of a Software-As-A-Service Solution: An Experience Report

*Software-as-a-Service or SaaS is a “new” approach to providing software solutions based on a Multi-tenant solution, focusing on configuration over customization and moving companies from software licenses/maintenance to software subscriptions. SaaS helps reduce the costs of managing software and provides faster implementation opportunities but it also requires a different perspective for large enterprise IT organizations to manage. This presentation provides an experience report of several SaaS implementations at a large enterprise IT organization. Presented at the **2010 Project Management Institute Conference (Houston)**.*

Iterative Development Testing Approaches

Over the past five years, there has been increasing interest in agile development approaches to software development (such as eXtreme Programming), however, integrating these into a unified testing approach can be challenging given their rapid delivery model. Applying a testing framework to an agile development approach provides a greater opportunity to ensure

a robust and high quality application. This paper reviews a web-services software development project completed at the end of the year 2000 for a large Fortune 500 company. The project used an object-oriented design and blended the more formal Rational Unified Process (RUP) with the low ceremony approached advocated by eXtreme Programming (XP). The application testing approach applied the unit test framework of XP with a formal testing methodology required by the client for this high profile project. The paper describes the process followed and key learnings discovered throughout the project lifecycle.

Accelerating Enterprise Agile Adoption

As organizations embark on increasing their agile effectiveness understanding the agile maturity of your organizations becomes critical for scaling agile adoption. This presentation and associated reference material provides a framework to track agile maturity and provides role playing scenarios to determine where your team members are on their agile journey. This presentation was part of the **Agile 2011 Conference, Houston Tech Fest and the Agile Alliance Summit.**

Agile Applied

Presented at the **Central Mississippi PMI Chapter Annual Conference**, this is the second part of a two part series

introducing Agile concepts and aligning them with the PMI-ACP certification. This presentation takes a detailed look at an Agile project and provides examples and techniques for managing an agile project.