

How to let Agile thrive in a Virtual Environment

One of the principles of the Agile Manifesto emphasizes the importance of co-located teams collaborating actively. Collaboration is still as valuable as ever but 2020 is forcing us to revisit how to achieve the benefits of co-location without the co-location. The good news is that much has changed since the Manifesto was written which has created some answers to how to enable the benefits of being Agile. Here's what you can do to allow Agile to Thrive in a Virtual Environment.

How to Allow Agile to Thrive in a Virtual Environment

How to make Agile work in a distribute and virtual environment

Remote First: Creating Resilient Organizations

One by-product of an increasingly global and interconnected world is the exponential increase speed of change and the feeling of constantly being in a state of crisis. How can companies thrive during in an environment of continual

crisis? The answer is developing resilience and one important step is developing a Remote First organization structure. Learn more about this idea by reading: Remote First: Creating Resilient Organizations with my co-author Bhavik Modi.

Remote First: Creating Resilient Organizations

Creating a Remote First organization unlocks powerful competitive advantages for companies that rapidly adopt this approach. Learn how Remote First can help you organization thrive in a time of continuous change.

Which is the Best Agile Certification for you?

In this age of rapid business acceleration, Agile is an answer for organizations struggling with how their traditional methods of Project Management are responding to change. Since, many organizations are looking for agile knowledge and experience to help them adapt their traditional methods, Agile certifications are an important way professionals can enhance their career opportunities so they can be part of this rapidly growing area. Agile certifications provide career advancement, salary and promotion opportunities so they're a great way to invest in your future.

There are a lot of Agile certifications available and it's important to pick the right one so you can get the most for your time and money. Here is a short overview of the top 3 Agile certifications so you can pick the one that will help you the most.

[Read Article](#)

Requirements and Analysis: Techniques and Tools (article)

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 [Fittes 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.