

# Integrating Scrum with the Process Framework at Yahoo! Europe

Karl Scotland  
Yahoo! Europe  
kjscotland@yahoo.co.uk

Alexandre Boutin  
Yahoo! International  
alexandre.boutin@yahoo-inc.com

## Abstract

*Large enterprise organisations usually have a Project Management Office (PMO) whose responsibility it is to oversee a standardisation of project management practices across a portfolio. This can often be in conflict with the Agile principles of self-organisation and inspect and adapt. We describe how one of Yahoo! Europe's teams helped implement a process which was compatible both with Agile values and principles, and European PMO standards, and also how it has influenced International Process Framework evolution. As a result, we demonstrate that a PMO process can be compatible with an agile process, and that a PMO can add value to agile teams.*

## 1. Introduction

Yahoo! International Engineering is an organization of over 1500 people distributed across three regions (Europe, Asia and Emerging Markets), and over 15 countries (excluding USA).



Figure 1 - Yahoo! Global Markets

In 2004-2005, Yahoo! International management structured the organization (3 regions), standardized roles and accountabilities, and started defining a strategy for each role. These initiatives provided improvements and consistency across the organization,

and by the time they were completed, were considered by US executives as Yahoo! International “Forces”.

At the end of 2005, Yahoo! International management considered it was time to address 2 top problems:

- Reliability of projects. Teams produced inconsistent results and often depended on a Hero culture for success.
- Quality of deliveries. Projects often accrued a high level of technical debt, resulting in instability and high maintenance costs.

By early 2006, the International Process Strategy Group was created to address these problems and decided to define a common “International Process Framework” (Terminology standardization, Teams education, Umbrella for other strategies) and to establish “Local Process Groups” in each region.

### 1.1. International Process Framework

The International Process Framework elaboration was evolved through a number of iterations with the following general advances:

- 2006: Full waterfall with Local Process Groups in a controlling role. The International Process Group had a background in waterfall, with no agile experience in any of the process groups.
- 2007: An open framework, independent from methodologies and with less control. This was due to positive results from Scrum pilots in London and positive feedback on Scrum from Yahoo! US teams. In addition, the waterfall-based process had proved to be too slow, with a lack of responsibility.
- 2008: A Scrum based implementation of the framework, with Local Process Groups as Coaches. The Engineering management team had made the decision to move to an Agile approach, and teams needed on the ground coaching in implementing Scrum.

The second section of this paper gives more information about how the London Audience team

helped inspire and implement this International Process Framework.

## 1.2. Local Process Groups

During 2005, a Process Group was established in Europe and provided benefits to the company, so this concept was naturally extended to other regions – even if it was sometimes challenging to get regional management agreement to invest in Process resources.

The overall strategy was to establish local Process Groups in each country, rather than having a centralised Process Group in one location, and is best illustrated with the motto:

“Nothing is better than speaking in native language for teaching and getting peoples’ buy-in”

Each Local Process Group works in the same mode and Yahoo! call it: The **DICE** model

- Definition of local process
- Implementation by education and training
- Coaching by participating to projects
- Evaluation of local practices, learning and continuous improvement

All local Process Groups are allowed to define and practice their own agility, but they frequently synchronize with the International Process Group to ensure their local processes correctly implement the International Process Framework, and to be able to share and learn practices from other regions or countries.

## 1.3. Challenges to move to Agile

Implementing Agile in different teams in different countries requires taking into account cultural specificities, local market needs, team maturity, and local organization and relationship with the business organization. Each country is moving to Agile at a different speed and the International Process Group has, at the same time, to support groups starting with Agile and to lead or contribute to new or advanced initiatives e.g. a kanban approach being used in London.

Some Yahoo! International “forces” could be considered as blockers for moving to Agile. The concept of “shared resources” (role based) is contradictory with the “dedicated resources” concept of Agile. The distribution of specialized resources (the testing group for Europe are based in France) prevents having a whole Scrum team co-located. Some specific functional strategies could not fit in Scrum (Design Review required by Architecture Strategy Group).

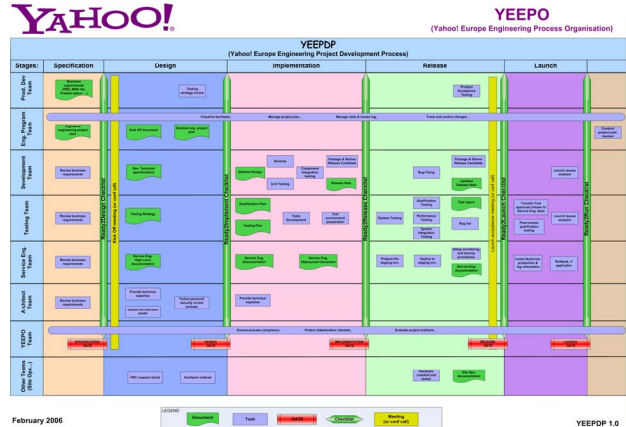
Some Yahoo! products are developed centrally (most often in the US), sometimes heavily adapted locally, and integrated and tested in the regions that will launch it. When the core product development methodology is Waterfall, it’s difficult to implement an agile approach because it creates long waiting times for International agile teams due to high dependencies and lack of visibility on the real delivery dates.

## 2. The Journey

In early 2006, Yahoo! teams in the US had openly embraced Scrum and an Agile team existed to help teams implement the new processes and practices. In Europe, there was awareness of Agile, and what US teams were doing with Scrum, but little experience or understanding to do the same.

Around the same time, a number of new employees joined the company (including one of the authors) who had the missing knowledge, and were keen to begin initiating an agile adoption with the support of their peers as part of the Audience London group. This group consisted of a number of teams working on a variety of product, the best known of which is the Front Page (e.g. <http://www.yahoo.co.uk>).

### 2.1. YEEDDP 1.0



At this time, the European Process Group had announced the “Yahoo! Europe Engineering Project Development Process” (YEEDDP 1.0).

As Figure 2 demonstrates, the YEEDDP 1.0 was very detailed in its breakdown of the process - the result being a very heavyweight and high ceremony way of working. The following elements were identified:

- Stages i.e. Specification, Design, Implementation, Release, Launch

- Roles i.e. Product Development Teams, Engineering Program Teams, Development Teams, Testing Teams, Service Engineering Teams, Architecture Teams, YEEPO Team and Other Teams
- Documents i.e. Specifications, Designs, Strategies, Plans, Notes, Reports, Others
- Tasks i.e. Reviews, Development, Unit Testing, System Testing, Live Environment Preparation, Project Coordination, Process Compliance, etc.
- Gates i.e. Specification, Design, Implementation, Release, Launch
- Checklists i.e. Ready2Design, Ready2Implement, Ready2Release, Ready2Launch
- Meetings i.e. Kick Off, Release

While all these elements describe some aspect of good practice, the overall complexity is not Agile due to the additional operating expense, and controlling element. An implication is that individuals are not expected to think about the process, but simply follow the steps and complete the artefacts.

The Audience London teams had three options when presented with YEEPDP 1.0. They could ignore it, comply with it fully, or engage with the Process Group in order to educate and inform the process in order to evolve it into one which enabled the development process, rather than one which added overhead. This last option was selected, with the teams effectively volunteering to be a pilot for using YEEPDP with Scrum.

## 2.2. EDP4EU 1.0

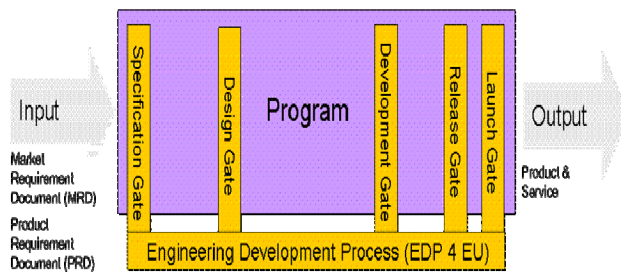


Figure 3 - EDP4EU 1.0 – Q4 2006

One of the main pieces of feedback from the Audience London group was that Agile teams valued collaborative, iterative and incremental development of working software as a measure of status and progress. Thus, while many of the roles and artefacts in YEEPDP 1.0 were well intended, the same desired outcomes could be achieved through frequent delivery with inspection and adaptation.

As a result, a new simplified process was designed, called the “Engineering Development Process for

Europe” (EDP4EU), which was intended to clarify the scope and goals of the initiative. The EDP4EU only covered the actual delivery phase of a project, with the inputs being a Marketing Requirements Document (MRD) and Product Requirements Document (PRD), and the outputs being the released code which is deployed into production. In addition, the gates remained as a means of validating the quality of the various aspects of the process.

While this still appears to be very waterfall-based process, it was agreed that the gates did not have to be performed in a linear manner. Instead, an agile team using iterative and incremental development could make the design and development gates optional. This model allowed agile teams to use a Product Backlog as the PRD, Sprint Planning as the Requirements Gates, and the Sprint Review and Retrospective as the Release/Launch Gate.

With EDP4EU, the London Audience teams could continue to use Scrum, while still conforming to the Process Groups standards.

## 2.3. Mapping Scrum to EDP4EU

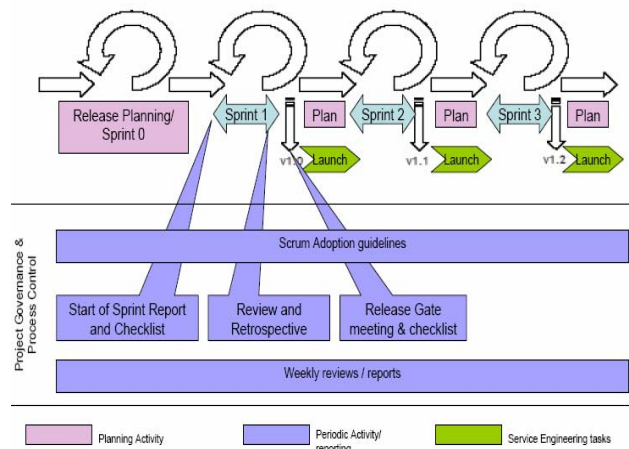


Figure 4 - Scrum and EDP4EU – Q1 2007

One of the side-effects of teams using Scrum within EDP4EU was that the Process Group lost some visibility of how teams were working because of the reduced ceremony and documentation. In order to counter this, the practice of checklists and reports was formalised, with Scrum specific checklists for agile teams, as shown in Figure 4.

A specialized “Start of Sprint” checklist was introduced as a lightweight “Specification Gate” as well as using the Sprint Review and Retrospective as the “Release Gate” with a “Release Checklist”. In addition, Sprint metrics, such as Product Backlog items

in progress and the Burndown, were included into weekly reviews and reporting.

This was the first significant step towards the Process Group recognising the different nature of agile processes, and enabled the Audience London teams to begin using a common language with them.

## 2.4. International EDP 3

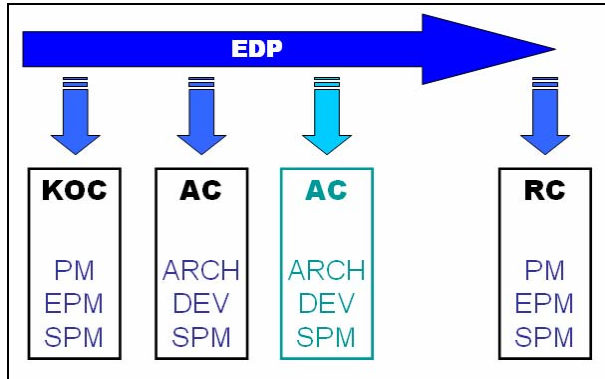


Figure 5 - Intl EDP 3.1 – Q1 2007

At around the same time, The International Process Group announced a new “International Engineering Development Process” (EDP3). This framework was another significant advance, because it provided a major simplification to a set of checkpoints which were based on organisational requirements, rather than process implementation. As Figure 5 shows, they are:

- Kick-Off Checkpoint (KOC)
- Architecture Checkpoint (AC)
- Release Checkpoint (RC)

The Kick-Off Checkpoint is to determine whether the organization knows what is going to be done. This checkpoint maps onto the agile concept of project chartering, when the vision, goals and objectives are agreed.

The Architecture Checkpoint is to give the organisation confidence that the team knows how it is going to deliver the project. There can be more than one AC, so they can be repeated a number of times, allowing the architecture to iterate and increment.

The Release Checkpoint is to demonstrate to the organisation that the team delivered the right thing. This checkpoint ensures the quality of what is delivered into production, both from a business perspective, and from an engineering perspective. For an agile team which has been working as a whole team, with continuous collaboration between the various business and engineering stakeholders, this checkpoint should produce no surprises.

Figure 5 also shows the stakeholders involved in agreeing each checkpoint. They are:

- Product Manager (PM)
- Engineering Programme Manager (EPM)
- Service Programme Manager (SPM)
- Architect (ARCH)

The Product Manager is equivalent to the Product Owner. The Engineering Program Manager coordinates the delivery, and is equivalent to the ScrumMaster. The Service Programme Manager coordinates both the deployment and support of the released software. The Architect ensures that the delivered solution reaches a sufficient technical quality.

One of the main consequences of the EPD3, was that there was a change of emphasis from controlling to coaching. Rather than trying to define how processes should be run, it defined the organisational goals that the process was trying to achieve.

In EDP3, our approach changed from “doing the right things” (applying a strict process and controlling it) to “doing things right” (focus on results and support the team to achieve it). So one of the missions of the European process group became to coach the team by providing advice, sharing good practices from other teams, and help the teams to figure out specific solutions. Controlling was not a mission of this group anymore.

## 2.5. European Agile EDP3

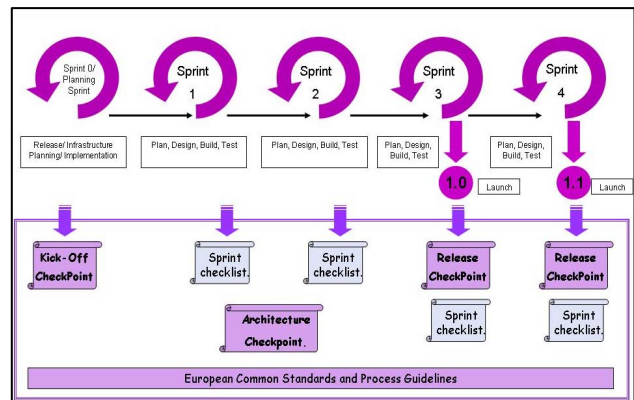


Figure 6 - European Agile EDP 3 – Q4 2007

As a result of EDP 3, the European Process Group updated its Scrum guidelines, as shown in Figure 6. This version introduced the concept of the Sprint checklist, which is a more process-centric checklist similar to the earlier Start of Sprint checklist. The aim was to provide more support for the Scrum specific practices and give some visibility of Scrum implementations to enable coaching. It was felt that Scrum teams doing iterative and incremental development with small cycles could end up under the

PMO radar because they never needed any major kick-off or release points.

## 2.6. The Future

The delivery teams are continuing to collaborate with the process group in order to inspect and adapt the process so that it can both enable the teams, and give visibility to the business.

A recent informal survey of programme managers showed that some found the checklists to be extremely useful as a reminder of tasks and actions they should be performing for the major checkpoints, while some found them an extra overhead which added little value. The strengths and weaknesses of checklists versus basic checkpoints is still being explored, with a view to seeing how the Process Groups ensure solid guidance on process and promote transparency of successes and failures, without becoming a policing entity.

Different ways of implementing the checkpoints and checklists which minimise the overhead they add are still being explored. An online checklist tool is currently being trialled to see whether it reduces the effort required, and increases the resultant effectiveness compared to the original spreadsheet based format.

Another idea that is being tried is to plan checkpoints on a quarterly basis, using them to improve visibility to the wider business of agile projects which are more iterative and incremental. Thus agile teams could have a kick-off checkpoint at the start of a quarter to set the roadmap, and a release checkpoint at the end of a quarter to review progress against the roadmap. Architecture checkpoints would occur as necessary dependant on the nature of the development work, with the architect ideally being closely involved in the project on a regular basis, rather than solely at checkpoints. One team is holding fortnightly meetings with the architect to discuss any relevant topics.

## 3. Summary

Our experience has shown that a PMO, and specifically a Process Group, can add value to agile teams if set up appropriately, as opposed to being something usually associated with waterfall and command and control cultures.

Close collaboration between delivery teams and the PMO is essential in order for each party to understand the needs of the other and to create standards and frameworks which can satisfy everyone.

The implementation of a standard framework should follow its own inspect and adapt cycle. It is unlikely that the ideal process will be invented in one

go, so trying ideas out and observing how well they work, while iterating over the cycle will work better.

As with agile software development itself, collaboration and iteration creates a relationship built on trust, enabling much more effective progress to be made.

Finally, organisational-based, rather process-based checkpoints allow delivery teams to have more flexibility in implementing practices which work for them, but still meet the business needs. This allows a Process Group to be a coaching rather than a controlling organisation. The goal should be to share and foster proven practices, as opposed to trying to define and enforce any silver bullets.