

Christian Sepulveda
cs@atdesigntime.com
OOPSLA 2003 Workshop
"Are Agile Methodologies Really Different?"
Position Paper

Each agile process can be characterized by one or more activity biases. For example, Extreme Programming is a coding centric process. Consider the 12 practices: Metaphor, Simple Design, Automated Testing, Refactoring, Pair Programming, Collective Ownership, Continuous Integration and Coding Standards are directly about coding. The remaining practices (Planning Game, Small Releases, Onsite Customer and 40-hour Week) are about either staying out of the programmers' way so they can code or improving communication about what they are going to code.

Scrum is project management centric. Scrum specifies the length of the iteration and the product backlog; both these elements guide the management of the project.

Comparing the XP and Scrum, note that XP doesn't specify the length of an iteration and provides loose guidance for story management. Scrum doesn't offer much guidance for coding or testing. I think the two different core perspectives mark the differences of the two methodologies, but also make the two quite compatible. (In my experience, XP developers welcome scrum as it doesn't outline many activities for them and provides protection so they can code; for a developer, the use or omission of scrum has little impact.)

Armed with a catalog of agile processes, categorized by the activities they are based on, a project team can elect the process or amalgamation of processes that are appropriate for its context and needs. For example, as the activity biases of XP and Scrum are different, they do not interfere with each other and therefore are compatible.

Certain agile processes are incompatible, as they share an overlapping set of activity biases. Feature Driven Development and XP would probably be in conflict as they both have programming biases. However, Feature Driven Development offers guidance for programming, project management and requirement identification. For teams who require structured support for all these activities, FDD may be a good choice.

Processes that have more than one activity bias are higher in ceremony than the processes that have one activity bias. It is important to identify these distinctions; otherwise a team could choose more or less process support than it needs.

I think process refactoring is a necessary and beneficial activity. A project team should consistently evaluate its progress and process needs. Based on this evaluation, the team should augment and modify its process to accommodate its current requirements. This approach encourages a "just in time" or "on demand" approach to process selection and tailoring.

Knowing the activity bias of each agile process supports process refactoring. Process refactoring requires the identification of the team's shortcomings or process "smells". It has been my experience that these "smells" are frequently complications of some activity in the development lifecycle. For example, XP doesn't provide much support for identifying a release date or the termination of project. For some teams, this a project management "smell". A scrum Burndown Chart could provide the project management that alleviates this need.