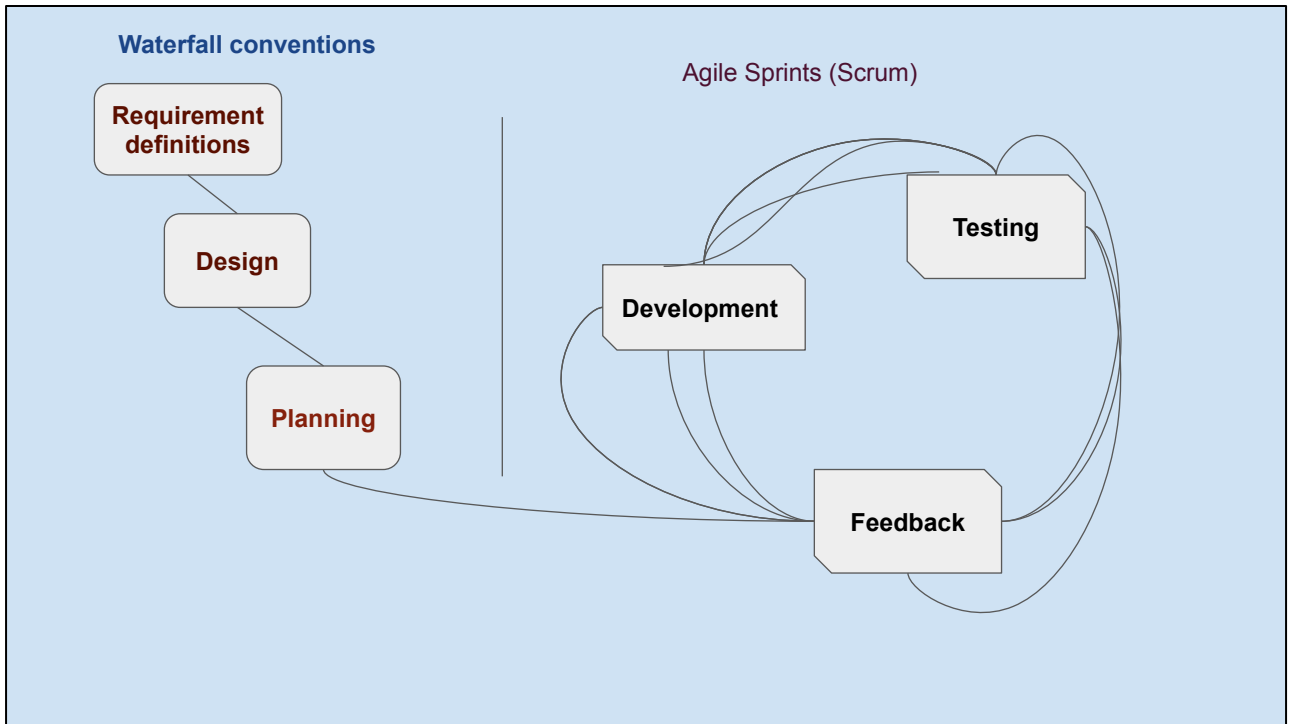


Kanban, Scrum, Waterfall

When one or the other becomes the best suited

Kanban	Scrum	Waterfall
<ul style="list-style-type: none"> • When interaction with user is crucial for the project. • When changes in requirements are highly expected. 	<ul style="list-style-type: none"> • It is reported that it works well in complex projects when crossing platform is involved. • When the project depends on parts component which are themselves other projects on their own. • When the user requirements are essential, welcomed, and encouraged. 	<ul style="list-style-type: none"> • When the user should adapt himself to the functionalities of the project rather than otherwise, think of gov, bank, interfaces and other public domain resources. • The project follows e.g. strictly and or discretionary regulations.

Since Kanban and Scrum are both applications of the Agile approach, it is probably more difficult to tell when which is better suited. However, there are some how-to-it-goes that change things. Both are made with the user involvement in mind, though Kanban goes better when this interactivity is likely to be more continuous, such as ticket resolutions, maintenance, and applications support. When interactions are based on increments of functionalities, new features, enhancing features, new sub-products or products themselves; or interactivity is likely better evaluated by units of work to be delivered (as said, plugins, enhancement of features, or new products) (whose broken-down list of items to work on, queued as a product backlog) is susceptible of management in sets of time (sprints) then Scrum system seems better suited.



Despite the scenery above quickly described I have found reports on using Agile on projects under strict regulation. Actually, when we argue against the Waterfall approach we refer to the wrong idea of the Waterfall model if we zoom in on the concept as presented in the foundational paper. So it should not be unthinkable to combine both, the question is how-to. The most extended formulas I have found go like this, dealing with the input material from stakeholders or customers and performing the requirement definition, design, and planning stages under the Waterfall method, then jumping into an Agile pattern for development, testing and feedback under Agile conventions in cycles of sprints.

The Zen of learning best suited in...

Scrum	Kanban
One thing no else until that thing and no else	Adapt changes to changes, now

About speed, the change cycle and the learning curve. Projects that should deliver a return as soon the requirement is at the work front line use Kanban. Under the Scrum system the cycle is given by the sprint timeline, no changes are injected in the middle of a sprint.