Kanban, Scrum, Waterfall

Strength and weaknesses

Kanban, vantages

- Make it easy to have a panoramic of every step and where issues arises.

- Facilitate cooperative work since team roles do not establish differences other than working around the task.
- When at issues the system makes it easy for the whole team to become a facilitator.
- In practice the whole team works on the same item making it pass to the DONE column.

Let’s point out some interesting vantages of the Kanban methodology

Kanban, when the good becomes an issue,

- The system makes no easy to forecast when a work item will get the next position on the board.

- When a item is delivered

Scrum, the goodness

- The sprint protocol helps with predictability.
- The system leads the team into a collective training for improving to delimit scope and estimations.
- Since the project is still alive evaluation from the previous sprint becomes greatly pertinent.
- The meeting system helps to forecast eventualities before they become issues.

Scrum, the not so good side…

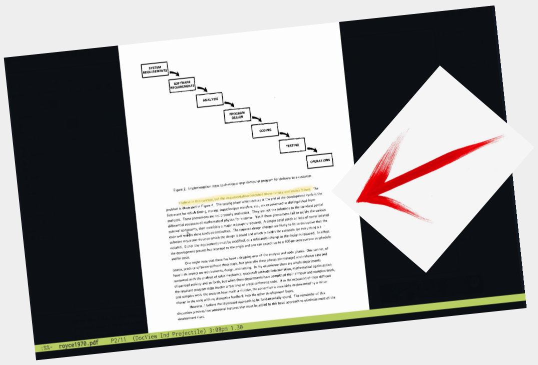
- The sprint deadline system could be tiring.
- Pretending to be a dynamic application of the interactive philosophy of Agile, the scrum system could be too punctilious about who does what and when.

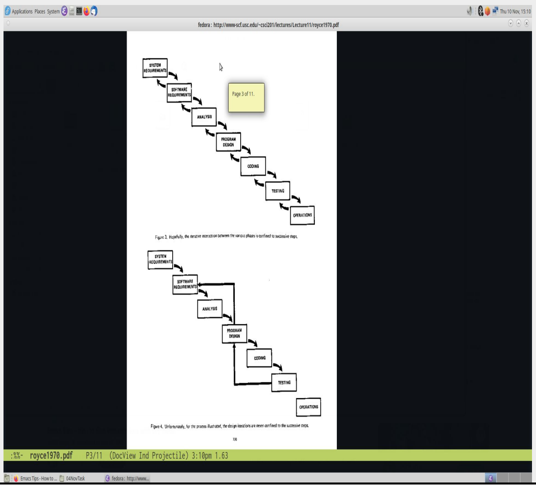
Waterfall, the system by default

- The final target of the project is known upfront.
- By design, cost estimations and margins are clearly set before starting.
- The work protocol allow members (business analyst, tester, developer, project manager) of the team to gain specialisation.
- The protocol of the model dive into a work culture based on exhausting documentation that makes it easy to transfer know-how information.

Waterfall, the very well known and documented drawbacks

- Revision and changes are not easily integrated in the process.
- User or customer are passive targets.




fedora : http://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf
SYSTEM REQUIREMENTS
SOFTWARE REQUIREMENTS
ANALYSIS
PROGRAM DESIGN
CODING
TESTING
OPERATIONS
Page 3 of 11.
Figure 3. Hopefully, the iterative interaction between the various phases is confined to successive steps.
Figure 4. Unfortunately, for the process illustrated, the design iterations are never confined to the successive steps.
royce1970.pdf P3/11 (DocView Ind Projectile) 3:10pm 1.63

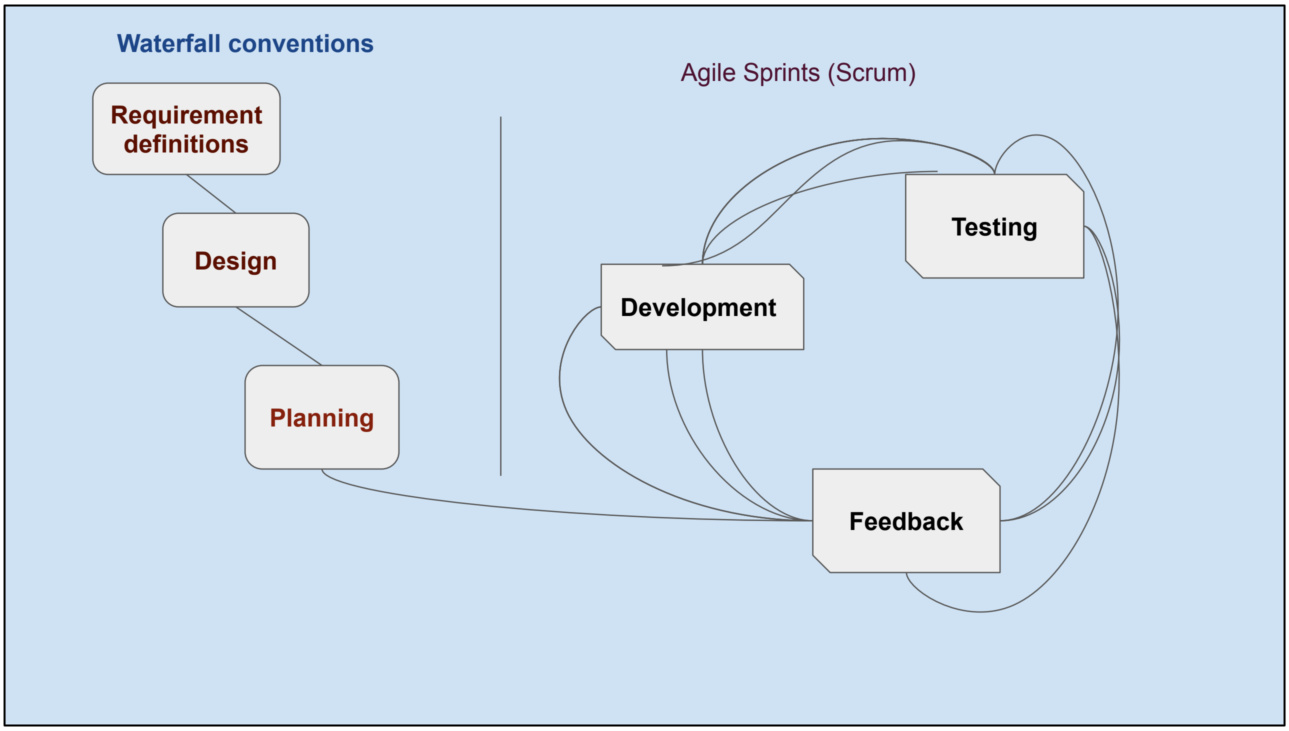
Something really interesting about the drawbacks of the Waterfall methodology is the fact that its so-called deficiencies were foreseen from the very moment of its proposal in 1970 and by its first author, W.W. Royce, and considering corrections suggested in the very foundational document, WW Royce known as the creator of the Waterfall method, could rather be considered not only its first critic but also the first author of Agile.

Kanban, Scrum, Waterfall

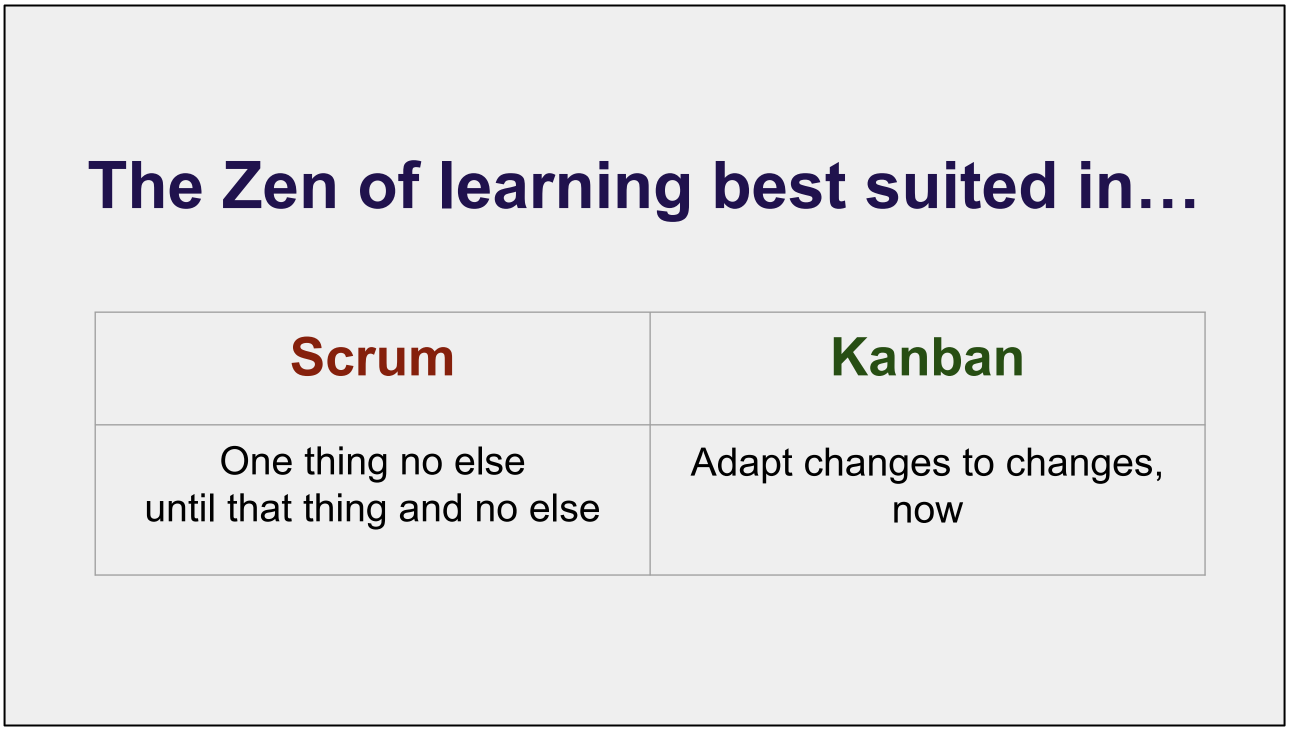
When one or the other becomes the best suited

Kanban	Scrum	Waterfall
● When interaction with user is crucial for the project. ● When changes in requirements are highly expected.	● 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.	● 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.