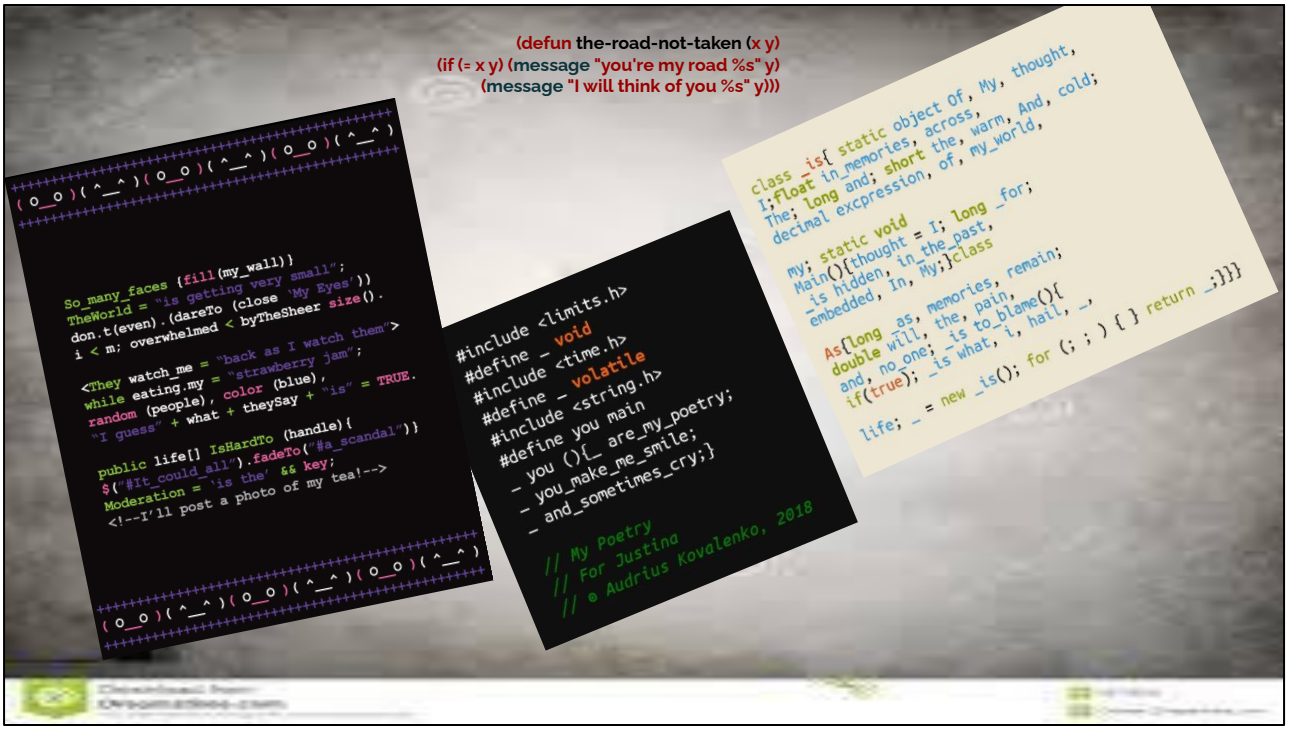
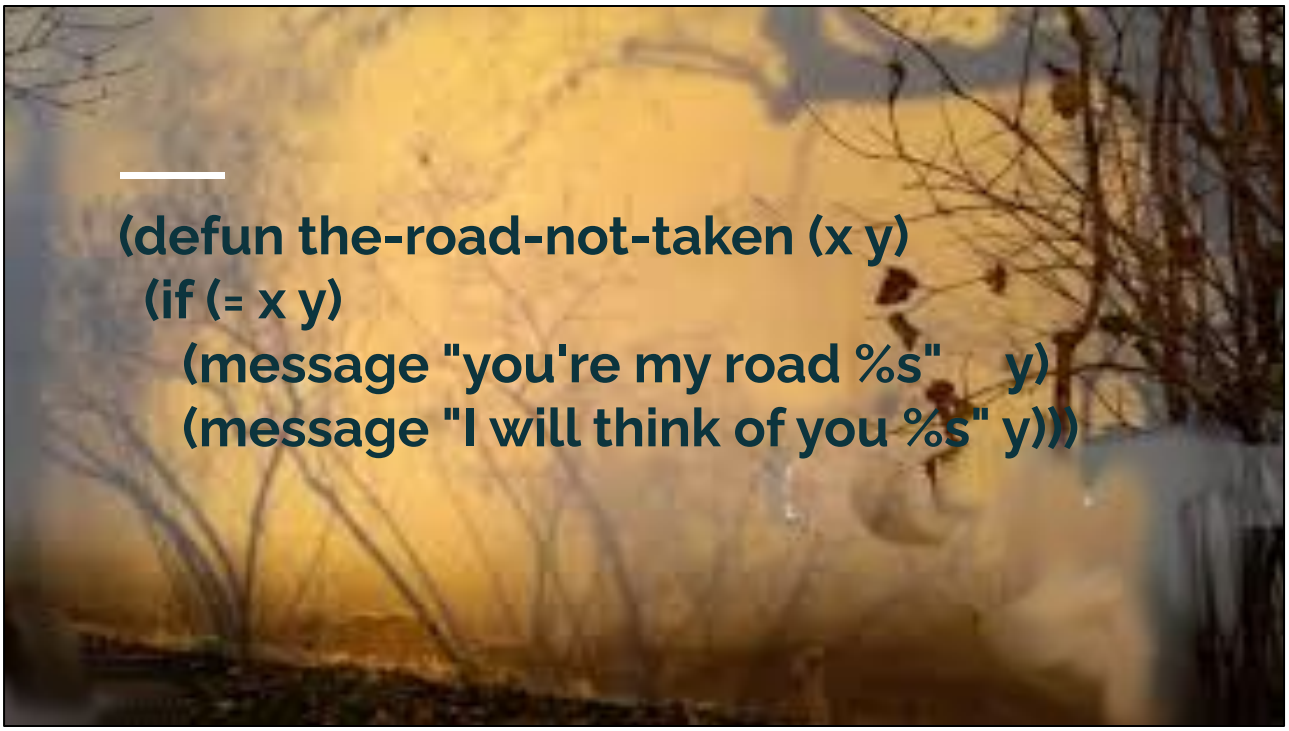
Kanban,

Workflows to optimize value to optimize value by optimizing flow.

Attending to the brief requirement for this task I am going to summarize the Kanban methodology though my choice is still a mystery I cannot explain entirely, yet.



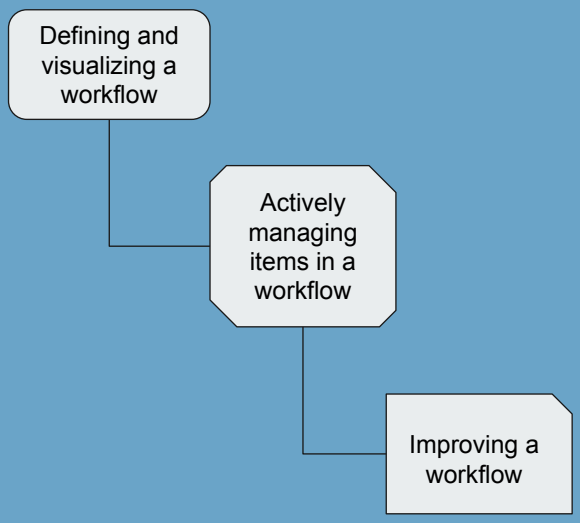
I should say that I got a bias on reading poetry, interest that swapped to coding



when once learned a little about programming I used to find myself quite long whiles awestruck on the beautiful simplicity of some scripts.

The same appealing reaction I got after learning that the keystone of the Kanban methodology was not that visually centred workflow tracking system everybody uses to talk about to define Kanban

As most workflows exist to optimize value, the strategy of Kanban is to optimize value by optimizing flow



this was just a myth among others, but the insight you might get by focusing on the quest of the method which reads as it follows "from workflow to optimise value, to optimise value by optimising flow".

The Basic
(No) tandem
(but)
Triplet

- **DoW**
- **Actively Managing Workflow**
- **Improving Workflow**

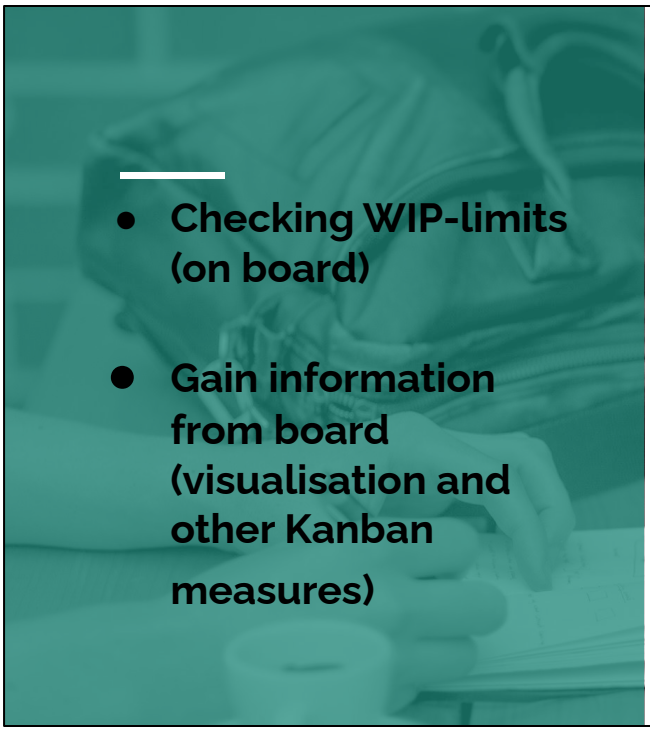
Being this complexity crowded with so many concerns in action, namely customer requirements, team skills and conditions the first quest is get proper definitions, do it properly and measure it. The Kanban handbook call it the basic Tandem:

- **Define When started & Finish**
- **WIP (Work in Progress) one or more states the work item flows from start to finish.**
- **Define how-to control WIP**
- **Explicit policies about how items can flow through each state**
- **a SLE (Service level expectation) which is a forecast of how long it should take a WI to flow from started to finish**

And here is the mysterious connection. You probably will got new insights by stopping and unblocking blocked tasks

- **Controlling WIP** (Work-Items in a workflow or slots/tokens on a Kanban board).
- **Avoiding W-Items piling up any part of the flow**
- **Ensuring work not age** (tool: SLE)
- **Unblocking blocked Work**

(it seems obvious but check out what is actually blocking and clean it) because solving tasks by coding is really of the same nature than crafting expressions with words, and as poetry, it depends so much on inspiration. Here creativity is a strategy.

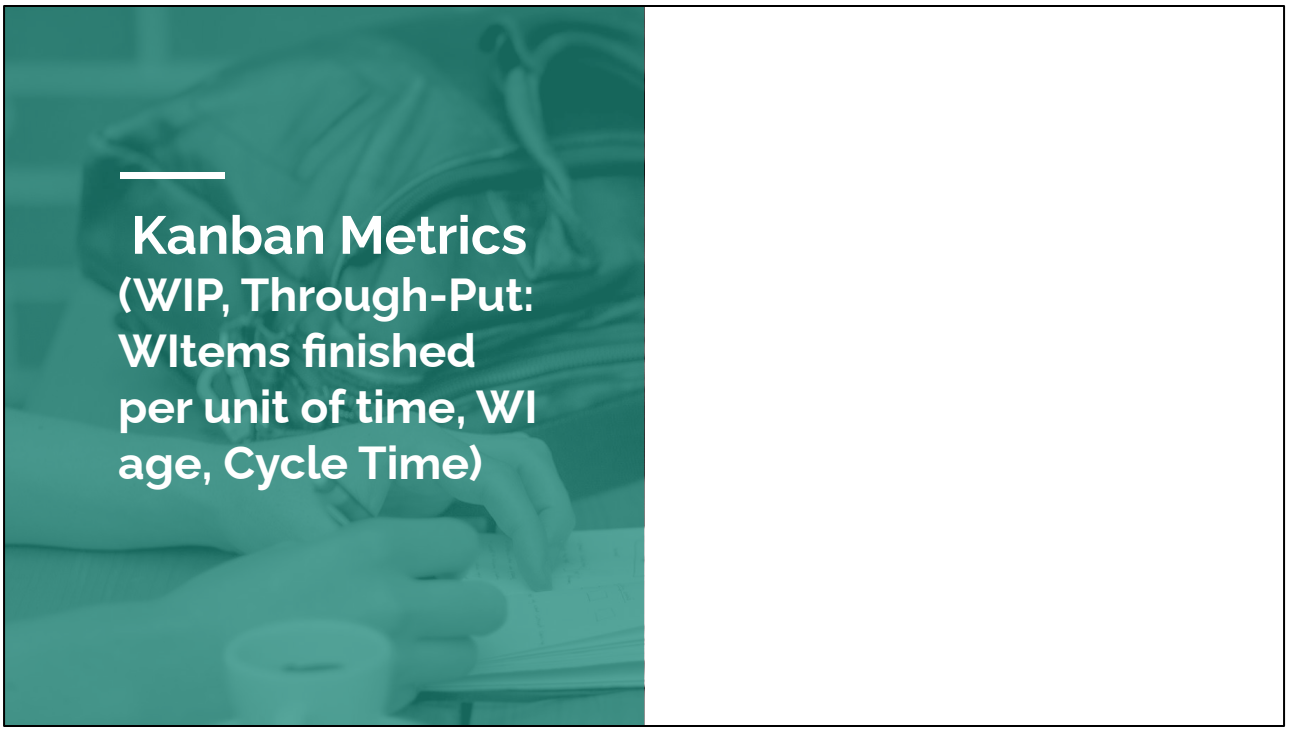


- **Checking WIP-limits (on board)**
- **Gain information from board (visualisation and other Kanban measures)**

State the problem you are solving in one or two sentences.

Make sure to explain why it is a real problem.

Tools are self explanatory about how-to deal with issues…



…even regarding how-to gain experience and improving as a team, at this point, making use of some smart metrics.

Describing Similarities and Differences

Differences

	Kanban	Scrum	Waterfall
Roles & Duties	Captain, Assistant (also found SRM and SDM) Service Request Manager Service Delivery Manager	Product Owner, Scrum Master, Team member	Developer, Tester, Business analyst, Project Manager

Delegation & Priority-Setting	Pull system, usually new tasks only, once the previous task is completed. (SRM) Set the next work item from the product backlog. - Checking indicators WIP, WIP-limit, DoW, SLE	Pull system though an entire batch is pulled for each iteration. P.Owner break down complex backlog items and make the team refine further into items suitable in one sprint.	The development sequence itself settles priorities and dependencies in advance
Productivity, measurement	by cycle time (time it takes to complete an project from beginning to end)	Capacity (team availability) & & Velocity (how-many user stories completed in previous sprints) (checking KPI, Key Performance Indicators)	As entirely sequential each phase must be completed fully before move on with the next

Similarities

	Kanban	Scrum	Waterfall
	Agile application focused on facilitate the workflow	structured Agile by breaking down task into sprints	noticed to be risky and invite failure from the beginning unless corrections involved feedback between stages

Considering the context, all methodologies share more than diverge.