

I have to	Scrum	<input type="checkbox"/> Be an Example <input type="checkbox"/> Establish Scrum <input type="checkbox"/> Enable a potentially releasable product after Sprint <input type="checkbox"/> Enable time-box
	Agile	<input type="checkbox"/> Facilitate meetings <input type="checkbox"/> Continue learning Agile & Scrum for Dev. Team and myself <input type="checkbox"/> Continuously increase productivity and quality <input type="checkbox"/> Continuously improve developing, engineering practices <input type="checkbox"/> Be sure that impediments will be eliminated
	Coach	<input type="checkbox"/> Lead by serving <input type="checkbox"/> Support to understand the Agile mindset <input type="checkbox"/> Ask indirect questions, influence but no authority <input type="checkbox"/> Train and coach for each new and old team member, this is a continuous process (technique, framework, learn to learn) <input type="checkbox"/> Make progress through small safe-to-fail experiment/accept failure as learning <input type="checkbox"/> Mediate through conflicts <input type="checkbox"/> Enable that everybody will gather impediments & solve them
	PO	<input type="checkbox"/> Support and guide my PO with his responsibility of his backlog <input type="checkbox"/> Get developing only driven by PO <input type="checkbox"/> Remind PO about maximizing ROI
	Dev. Team	<input type="checkbox"/> Get the Dev. team self-organizing <input type="checkbox"/> Get the Dev. team to decide <input type="checkbox"/> Get the Dev. team to responsible and motivated with passion <input type="checkbox"/> Get the Dev. team to work cross-functionally
Organisation	<input type="checkbox"/> Make sure everybody work full productive <input type="checkbox"/> Make sure Dev. Team & PO work together, enable cooperation across all roles/functions <input type="checkbox"/> Make sure teamwork is good <input type="checkbox"/> Serve my organisation	

**I'm not a Team Leader**  
**I'm not a Manager**  
**I have to be Master of Scrum**  
**I'm a servant Leader**



## Pillars

- 1** Transparency
- 2** Inspection
- 3** Adaptation

Ask the team		<ul style="list-style-type: none"><li>■ Is this useful?</li><li>■ What should I do? / What can we do?</li><li>■ Do you think we should do?</li><li>■ What is your decision now?</li><li>■ I noticed <b>&lt;situation&gt;</b> what shall we do?</li><li>■ I feel <b>&lt;feeling&gt;</b> do you share that ?</li></ul>			
Impediment		<ul style="list-style-type: none"><li>■ Remove redundant steps</li><li>■ Get everything away that can reduce the productivity</li><li>■ Don't accept status quo, ...</li></ul>			
Retrospective	Quick	<ul style="list-style-type: none"><li>■ What could you see? (x answer)</li><li>■ What could you feel? (x answer)</li><li>■ What could you learn? (3 answers)</li><li>■ What should be changed? (1 sentence + plan) ▶</li></ul>			Use the <b>5-Whys</b> Rule for finding the root cause
	Gather	<ul style="list-style-type: none"><li>■ Quality, Feedback, Communication, Simplicity</li><li>■ Satisfaction, Team work (e.g mark 1-10)</li></ul>			
Planning 1: WHAT	Planning 2: HOW	Understand Customer needs <ul style="list-style-type: none"><li>■ Design</li><li>■ MoSCoW</li><li>■ Risk</li><li>■ Customer Value</li></ul>		Physical	<ul style="list-style-type: none"><li>■ User story</li><li>■ Pen</li><li>■ Sticky note</li><li>■ Teamboard</li><li>■ Camera</li><li>■ Index cards</li></ul> ▶ K.I.S.S
Artifacts		Backlog ▶ <ul style="list-style-type: none"><li>■ Product</li><li>■ Sprint</li><li>■ Impediment</li></ul> Transparency Definition of "Done"			
Team Decision		Vote by hand Use sticks, dots Poker cards		Developer	<ul style="list-style-type: none"><li>■ TDD</li><li>■ BDD</li><li>■ XP</li><li>■ DOJO</li></ul>
				Management	<ul style="list-style-type: none"><li>■ Kanomodel</li><li>■ Team success</li><li>■ Management 3.0</li><li>■ PDCA- Cycle</li><li>■ Kanban</li><li>■ Burndown</li></ul>