



Don't be random!

Tools for structured team development,
for ScrumMasters, agile coaches,
managers and leaders

What's the problem?

Teams don't become teams by themselves. They need careful coaching and nurturing over a long period of time, by people who are close.

How do you get money and support from your boss?

How do you record what is going on in your team?

How can you describe the situation to a mentor?

Teams don't become teams by themselves

How do you get good input from a colleague?

How do you know if your work is actually helping?

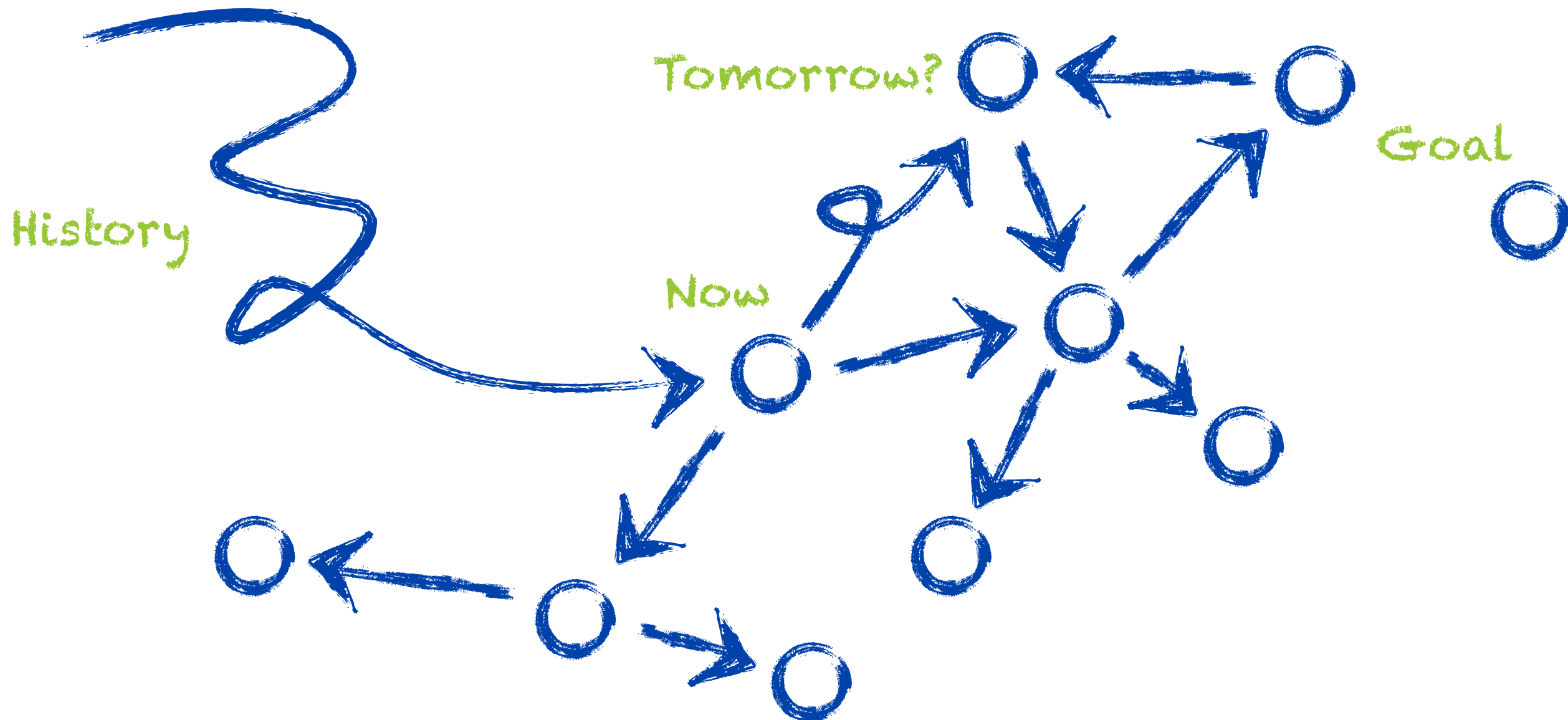
Changing the Organization



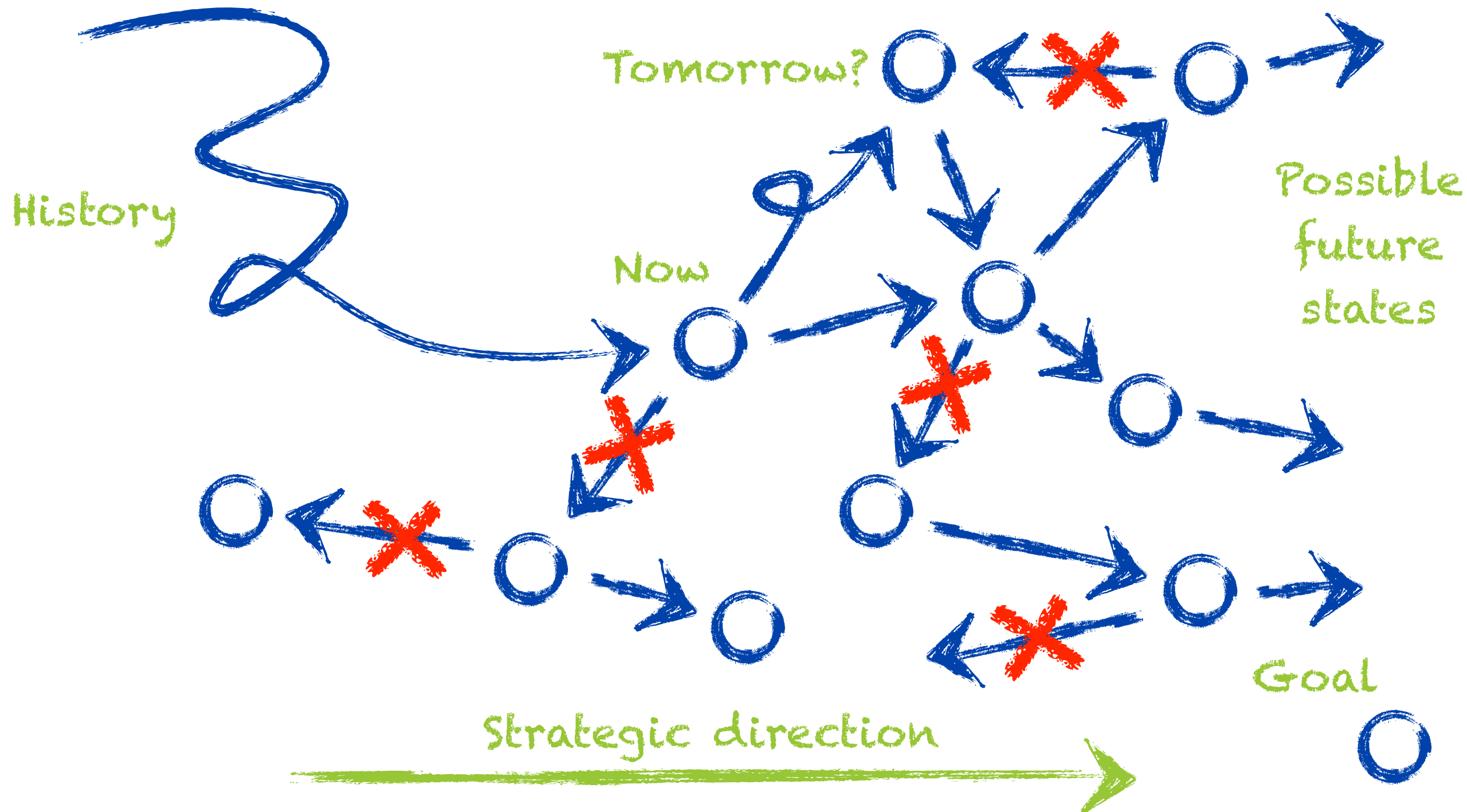
Organizing for Change



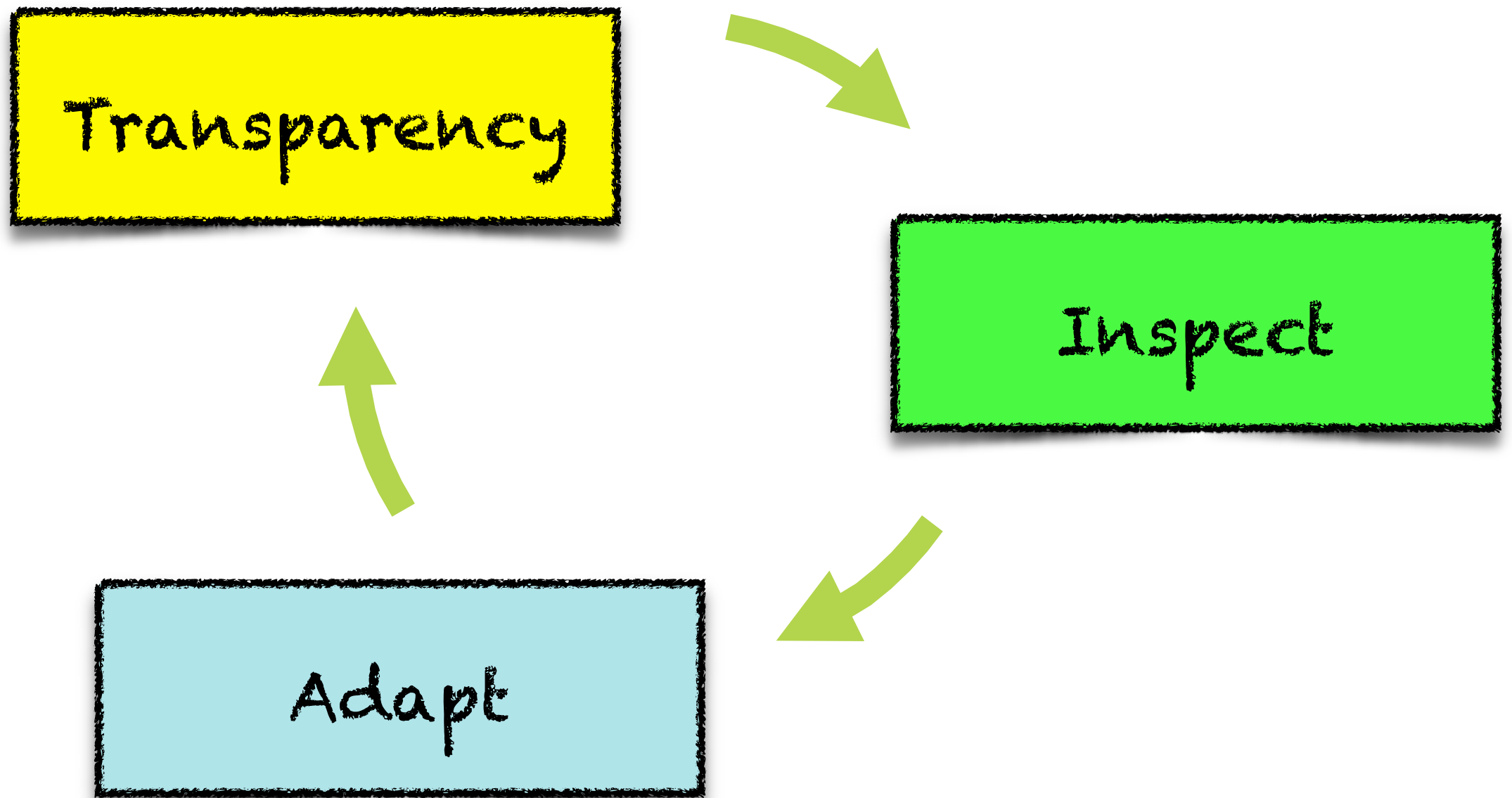
Organizing for Change



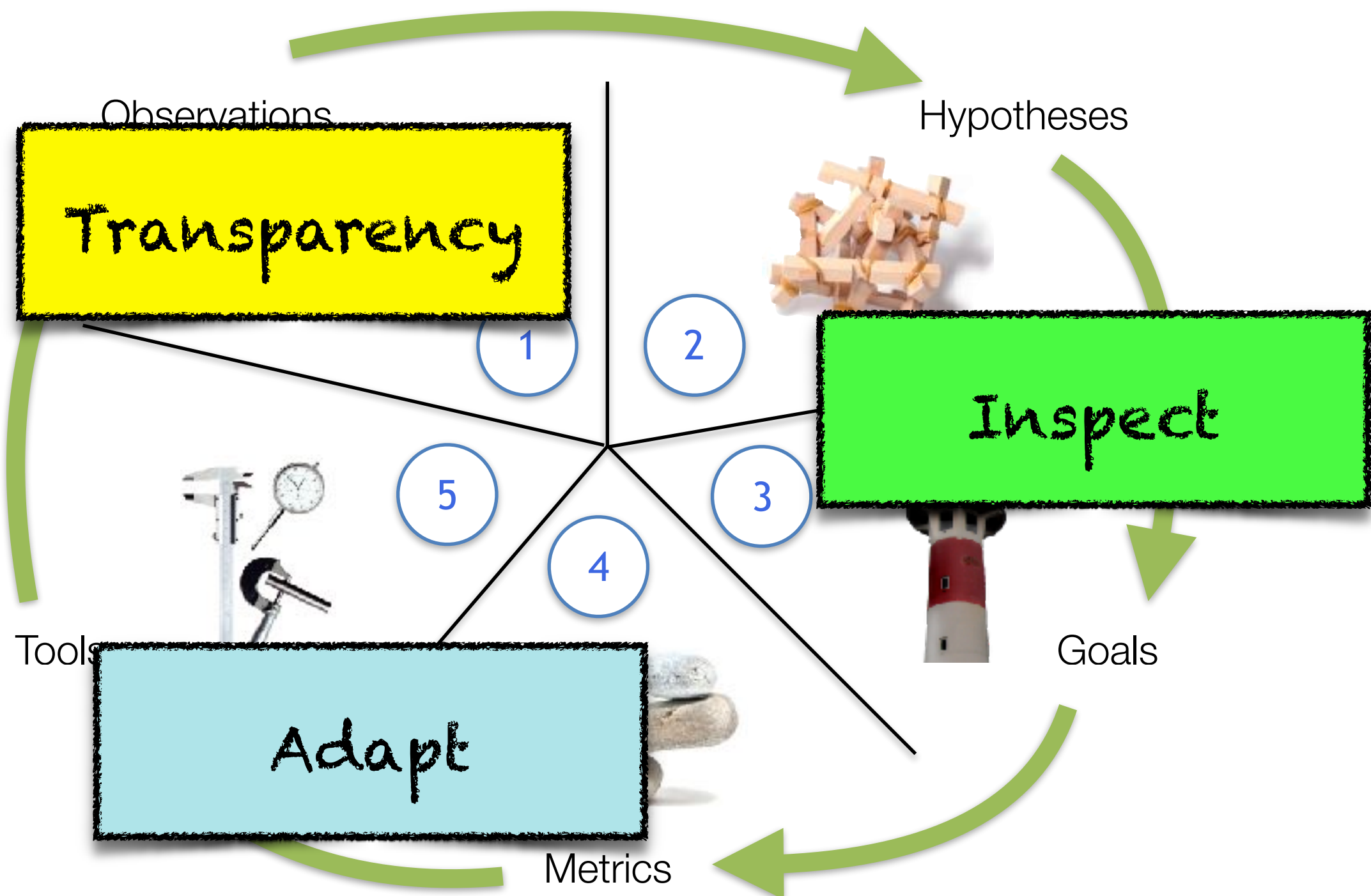
Organizing for Change



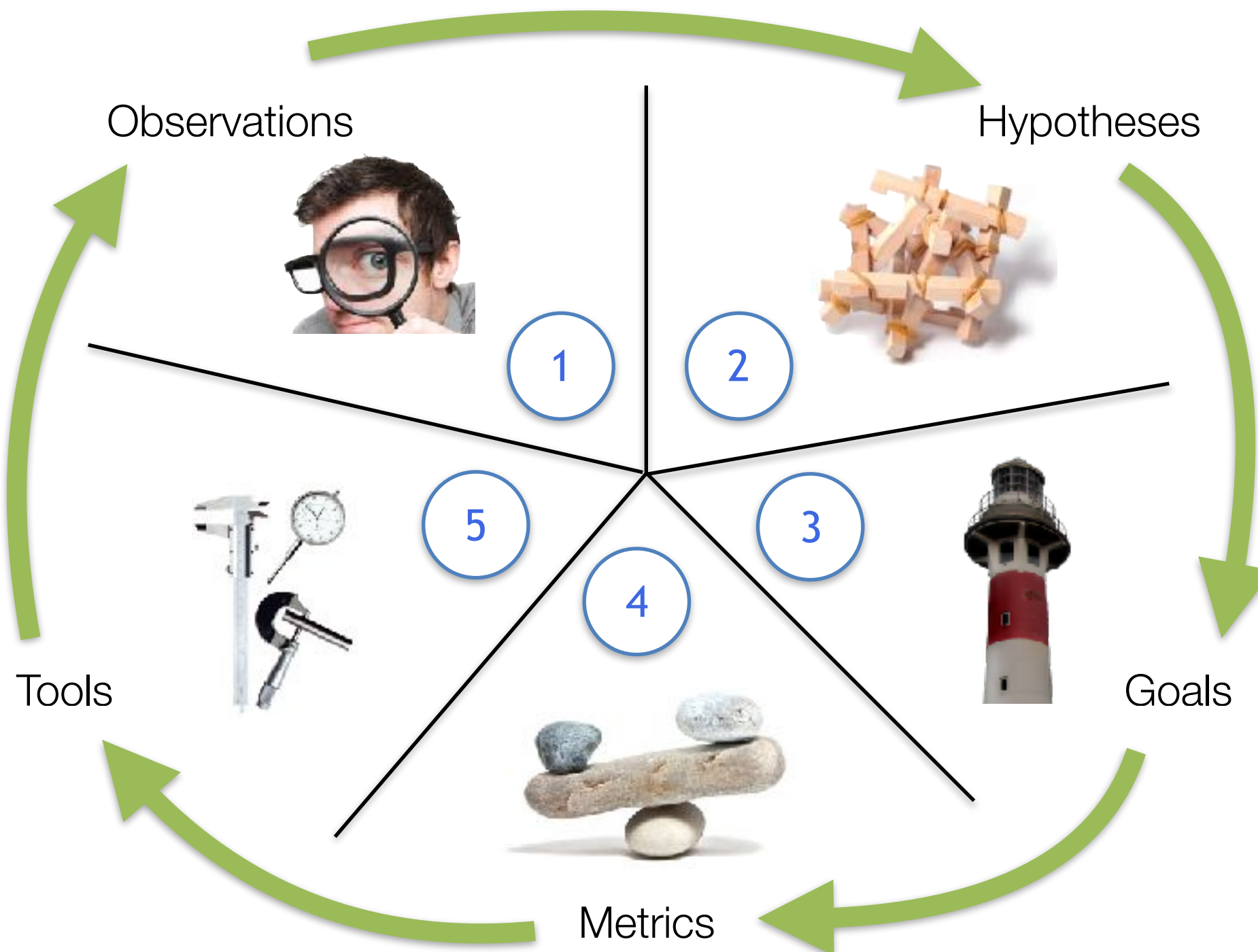
Empiricism



Coaching Structure



Coaching Structure



REALITY

(Karl Tomm)

1. Observations

Facts, metrics, indisputable

5. Tools

Specific, actionable, achievable

4. Metrics

Leading & lagging

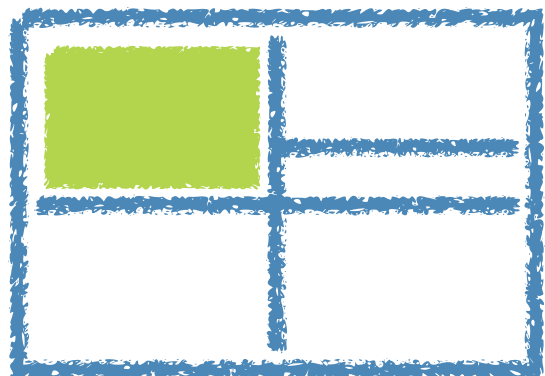
3. Goals

Challenging, lofty, relevant

2. Hypotheses

Speculation, opinions, different perspectives, hidden root causes

POSSIBILITIES**PAST****FUTURE**



Observing your team

- Hypotheses

- Curiosity

- Powerful questions



- Humbleness

- Empathy

- Patience



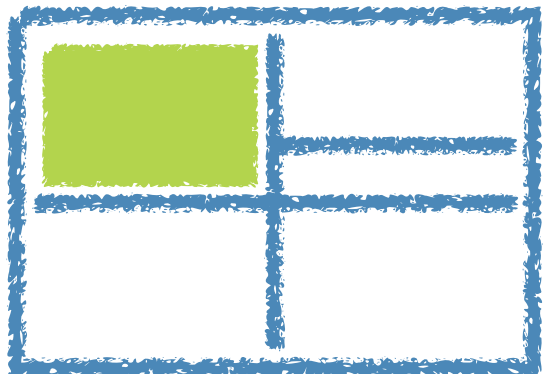
Observing your team

Events:
Planned stuff

Incidents:
Unplanned stuff

People:
Actions & interactions





Observing your team

Facts are facts.

It either happened or
it didn't happen.

Facts don't care about
what you think.

Hypotheses,
interpretations and
opinions are not facts.

They don't become facts
even if you say so.





Formulating hypotheses

- Beware observer bias
- Validate your hypotheses
 - Be open-minded
- Ask for advice
 - Don't commit early





Formulating hypotheses

Agile Values

Agile Principles

Experience

Force Fields

5 Why's

Fishbone





Defining a goal

Absolving = Ignoring the issue

Resolving = Any reasonable fix

Solving = The optimal fix

Dissolving = Redesigning to remove issue





Defining a goal



~~Absolving = Ignoring the issue~~

Resolving = Any reasonable fix

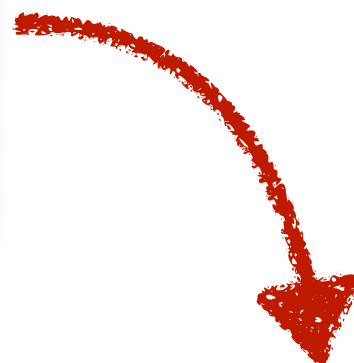
~~Solving = The optimal fix~~

Dissolving = Redesigning to remove issue



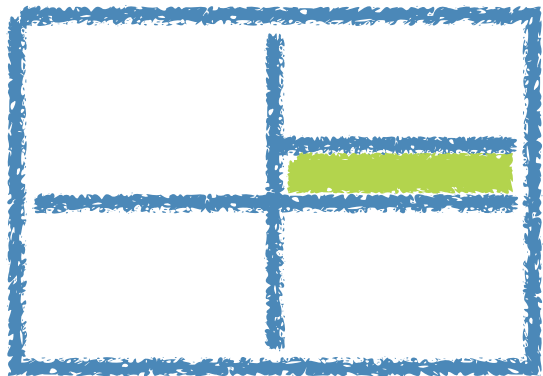
Defining a goal

Problem-solving goal:
"We want to have less
boring daily standups."



Transforming goal:
"We want to have valuable and
enjoyable daily standups."





Define metrics



Leading – To understand if we are getting closer to the goal

Lagging – To understand if we have reached the goal



Define metrics



"Miracle question": Assuming that the problem was fixed overnight, how would you notice?

- 1) Value produced
- 2) Efficiency
- 3) Qualities

- Observe people:
- What are they doing?
 - What are they talking about?
 - Who are they talking to?



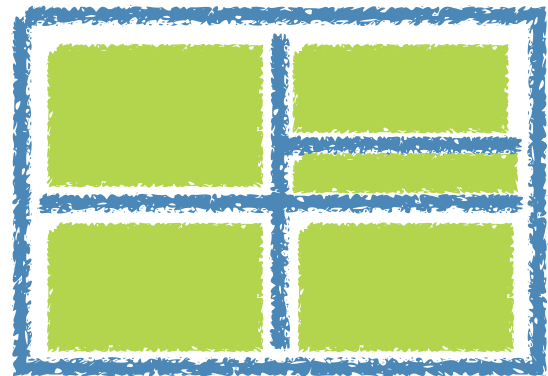
Pick coaching tools

Coaching Tool:
An intervention that
permanently changes
the habits of the team.



Some coaching tools:

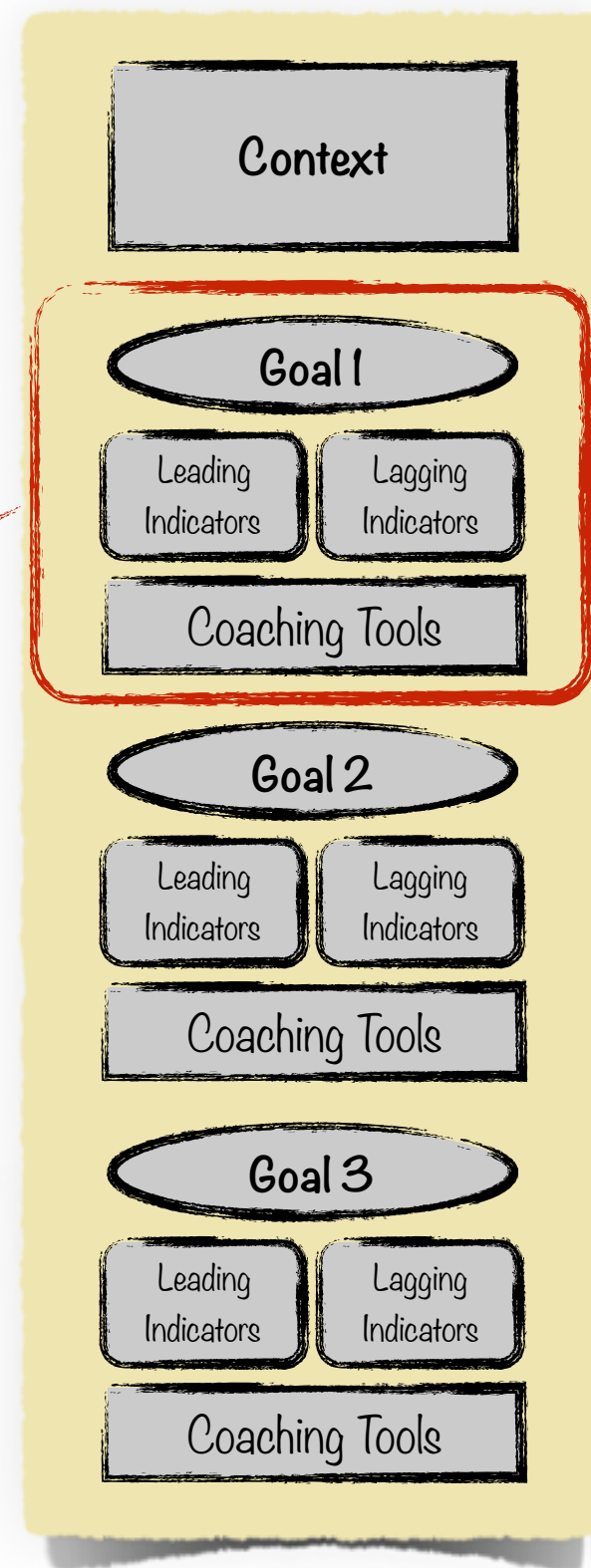
Surveys <ul style="list-style-type: none"> +15 TEAM agile assessment questionnaire "check on the team" (questionnaire + spider diagram) 	Documents/Cheatsheets <ul style="list-style-type: none"> scrummaster self-help worksheet scrummaster cheatsheet 	Metrics <ul style="list-style-type: none"> team ladder CFD PO/Release/Portfolio metrics/dashboards team metrics/dashboards control charts/histograms 	<ul style="list-style-type: none"> talking stick (allow self-organizing conversation, focus on our attention, full attention/respect) talk one-on-one with scrummaster, team members, PO parking lot look/listen for feedback agile wiki scrum of scrums stating observations stand behind talking person stand-up meeting correct way of shifting blame (failure: line manager or scrummaster accepts; success- all of team is regarded also for individual efforts) 	<ul style="list-style-type: none"> create stories in team everyone write a story (no gender no name, - send to facilitate - hang-up - guess who is who) team building in old fashioned sense (bowling etc) market of skills team name team rewards do food (beer) maximum pain (let them fail) team space (sofas, table, board, flipcharts) speed dating and other techniques for self-forming of teams (requires 3+ teams) growing teams exercise hudson bay start 	<ul style="list-style-type: none"> value stream mapping kanban board (visualize the flow) kanban system (adding policies to control the flow and measure it)
Flow <ul style="list-style-type: none"> limit WIP (kanban system) small batches pomodoro technique timebox meeting timer prioritized backlog (list of TO DOs) name game 	Guiding <ul style="list-style-type: none"> team coaching framework (PPT) kanban method mentoring (internal & external) sit with team / observe behavior agile strategy map gold card coaching structure 	Questions <ul style="list-style-type: none"> miracle questions Karl Tomm questions scaling questions powerful questions bridging questions 			
Care <ul style="list-style-type: none"> Teach them positivity and respect appreciation (promote good behavior) sharing positive examples or behaviors (sharing in a meeting) dancing and hugging be silly scrum dance 	Training / Education <ul style="list-style-type: none"> 4C's marshmallow challenge kata dojo kanban pizza game get kanban game 	Decision-making <ul style="list-style-type: none"> involve teams in decisions fist to five (quick voting, used to get impressions on the going) thumb voting roman vote dot voting (coaching tool used for rapid prioritization and decision making) traffic lights (to check working agreements and DoD) decision patterns for common tasks and questions (e.g. what kind of documentation is needed for this study) 	Estimation <ul style="list-style-type: none"> planning poker (facilitation tool structure conversation about product development) relative estimation (bubble sort) business value game (facilitation to allow stakeholders alignment with business goals) #noestimates :) 	Information Radiator <ul style="list-style-type: none"> happy/sad board portfolio board missing to ten spider chart (team dimensions) burndown chart burnup chart laugh-o-meter 	Polices <ul style="list-style-type: none"> working agreements definition of done release definition of done definition of ready pull policy
Community of Practice <ul style="list-style-type: none"> camps (e.g. PO Camp, coach camp) lean coffee hot topics brown bag sessions book circles 	Backlog <ul style="list-style-type: none"> portfolio budgeting release mapping release / portfolio planning meeting backlog grooming story splitting impact mapping user story mapping (focus on large business process view and break it down in smaller stories) user story (focus on user value and allow for conversation vs. written communication) 	Communication <ul style="list-style-type: none"> active listening ask the team skills matrix silence 	Team building <ul style="list-style-type: none"> ABIDE (attraction, barrier, identity, diversity, environment) make the team learn personal stuff 	Visualizing <ul style="list-style-type: none"> visual board visualize workflow team board avatars on team board task board (visualize the flow of work) draw your process (kind of value stream) PO board 	Problem solving <ul style="list-style-type: none"> pairing brainwriting pair programming evaporating cloud 4D model Fearless journey 5 why's A3
Validated Learning <ul style="list-style-type: none"> idea cards (lean startup, validated learning) minimum viable ... (release, product, test) lean canvas 	Vision <ul style="list-style-type: none"> elevator pitch workshop 				Sustainable pace <ul style="list-style-type: none"> personal kanban core protocols improvement backlog demand analysis
					Retrospective <ul style="list-style-type: none"> active learning cycle starfish pre-mortem safety check ROTI (return of time invest at the end of the meetings) reflecting team kaizen (continuous improvement practices) speed boat (gather data exercise/game)



Build a coaching structure



Coaching Card



Coaching Structure

	BACKGROUND 	HYPOTHESIS 	GOAL 	INDICATORS 	TOOLS 	REPORT 
PRO	<ul style="list-style-type: none"> - IS SHORT - CAPTIVATED - AN IMPROVED VERSION OF 	<ul style="list-style-type: none"> - A LOT OF CLARIFICATION - CLARITY IN "WHY WHY" 	<ul style="list-style-type: none"> - FOCUS ON WHAT WE'RE A TEAMWORK 	<ul style="list-style-type: none"> - INDICATORS - INDICATORS - INDICATORS - INDICATORS - INDICATORS 	<ul style="list-style-type: none"> - TOOLS - TOOLS - TOOLS - TOOLS - TOOLS 	<ul style="list-style-type: none"> - REPORT - REPORT - REPORT - REPORT - REPORT
CONCERN	<ul style="list-style-type: none"> - What is the team - Is it relevant - How many times has it been created? - How many times has it been created? - How many times has it been created? - How many times has it been created? 	<ul style="list-style-type: none"> - What is the team - Is it relevant - How many times has it been created? - How many times has it been created? - How many times has it been created? - How many times has it been created? 	<ul style="list-style-type: none"> - What is the team - Is it relevant - How many times has it been created? - How many times has it been created? - How many times has it been created? - How many times has it been created? 	<ul style="list-style-type: none"> - What is the team - Is it relevant - How many times has it been created? - How many times has it been created? - How many times has it been created? - How many times has it been created? 	<ul style="list-style-type: none"> - What is the team - Is it relevant - How many times has it been created? - How many times has it been created? - How many times has it been created? - How many times has it been created? 	<ul style="list-style-type: none"> - What is the team - Is it relevant - How many times has it been created? - How many times has it been created? - How many times has it been created? - How many times has it been created?

OBSERVATIONS

Two tasks on the task board have been in the same positions for at least a week. It seems like those two stickies were more or less abandoned as they got stuck, and the team members are instead focusing on other tasks that proceed at the usual speed. Also neither the Scrum Master nor the local team coach are reacting to this. If this keeps up, some stories will not be delivered.

TOOLS

- Hold training about "flow" using e.g. the Ball Point Game or one of the Lean batch size/WIP simulations
- Create a pull policy with the team
- Help Scrum Master set up an impediment board

METRICS

- The team reviews stuck tasks in every daily standup
- The team has a plan for an increasing number of stuck tasks
- Number of "stuck task days" trends towards zero

HYPOTHESIS

People do not understand or appreciate the concept of "flow" within the sprint, and how it relates to risk.

GOAL

Work flows within the sprint and problems are addressed immediately.

Team KDT Coaching Structure

1. Context

- [1.1 Collaboration is very weak and ad hoc](#)
- [1.2 Controlling the work in the sprint](#)
- [1.3 Meetings are unstructured](#)

2. Hypothesis

- [2.1 Team not collaborating on solving stories but working as individuals](#)
- [2.2 Lack of understanding of responsibility and risk management](#)
- [2.3 Unstructured meetings](#)

3. Goals

- [3.1 Team understands and accept a common Sprint Goal and commits to it](#)
- [3.2 Team is aware of the risk and focus actively in delivering at a regular and sustainable pace](#)
- [3.3 Effective and structured meetings](#)

4. Metrics

- [4.1 Improving collaboration and commitment to a common goal](#)
 - [Leading](#)
 - [Lagging](#)
- [4.2 Team is managing the risk through the whole Sprint](#)
 - [Leading](#)
 - [Lagging](#)
- [4.3 Meetings](#)
 - [Leading](#)
 - [Lagging](#)

5. Tools

- [5.1 Improving team collaboration](#)
- [5.2 Improve risk management during the Sprint](#)
- [5.3 Structured meetings](#)
 - [Books](#)

1. Context

1.1 Collaboration is very weak and ad hoc

The team doesn't seem to work as a team, mostly individual focus in developing or testing. There are very experienced people at play, and they would rather say the last word. This leads to a lot of unfocused conversations (in every meeting) and to a very low ability to come to conclusions and make team decisions. Despite this, the Daily Scrum look pretty good, and the team seems to have an overview on what is going on, since the training they are using a talking stick, which facilitates the conversation (before they used to have multiple conversation at a time).

Controlling the work in the sprint

At managing the risk inside a Sprint. They open multiple stories in parallel, not completed by the end of the Sprint. There are a lot of dependencies with the team and the team is not protecting itself from variation on that level. They are out of the framework in mid Sprint causing quite some troubles. The size of the stories, resulting in long lead times, and causing overburden for the tester the next day. Big stories means also less stories, and this means that the team is in high delivery risk, as a 13 points story is more than half of their current capacity.

Unstructured meetings

The meetings we have observed are extremely unstructured. In the retro and only a meeting room had been reserved. The time box for the meeting was not clear, no agenda, and the initial discussion expanded to fill the available time. When they had only five minutes left for the actual planning, they created a meeting agenda.

Team not collaborating on solving stories but working as individuals

When working on the sprint, the team members are not collaborating to reach the sprint goal. They work on parallel Backlog items and have problems finishing them by the end of the sprint. There is no collective responsibility feeling and team members are not tending to withdraw requests because they are busy with their own tasks. This lack of collaboration, the size of the stories, generates the behaviour that individuals feel they need to finish first in completing their "tasks" and then in helping the others. Most of the time, they get a story done by the end of the Sprint. Furthermore, as there is no one responsible for a story, as they get dragged along till they are eventually done. This lack of understanding the meaning of *commitment* and is not understanding their own responsibility, therefore they fail to manage the risk and to control the process.

Lack of understanding of responsibility and risk management

It is clear that there is a very high focus from the individual perspective on completing a story first. It seems there is no collective will of completing a story, either by the end of the sprint or that the story is the minimal unit entailing value, or it is not important to complete it... given the pressure Siemens is putting on delivering the second option. Focusing on completing the stories, according to the team, something the team is not doing, and they also appreciate the fact that

are not tested. In this way the tester is not fighting constantly with the tests, but works with "dedicated" developers. This lack of testing is a big dysfunction and is a spiral that might lead to

and lack of focus. It also sends the message that the team members feel that the Scrum meetings are boring, this could be a problem in the first place.

structured meeting.

It would be relatively easy to fix and quickly lead to positive results, as the risk is quite high.

Team understands and accept a common Sprint Goal and commits to it

The importance of having a shared Sprint Goal as a clear target, even before selecting stories. The choice of the right stories will be subordinate to the Goal negotiated for the sprint. The team is able to evaluate on which stories to focus on to reach the Goal. In this context everybody will collaborate on every story, in order to get it done as fast as possible, ensuring the efficiency that the team learned to deal with the awareness about uncertainty. Helping each other to achieve the goal of the team.

Team is aware of the risk and focus actively in delivering at a regular and sustainable pace

The team is working at a sustainable pace, and focuses on completing the Sprint. Starting from the Sprint Planning, the team evaluates the options to achieve the Sprint Goal, and they work towards the Goal. At the same time, the team will pay attention to the work of the individuals and doesn't create dependencies. The team actively seeks to commit to at least 6 stories in the sprint, frequently, in order to avoid larger stories, which would work towards the end of the Sprint. The team is aware of the dependencies (Syngo, Fraunhofer Mevis) and finally the team embraces empirical control on managing dependencies between those stories at the Daily Scrum.

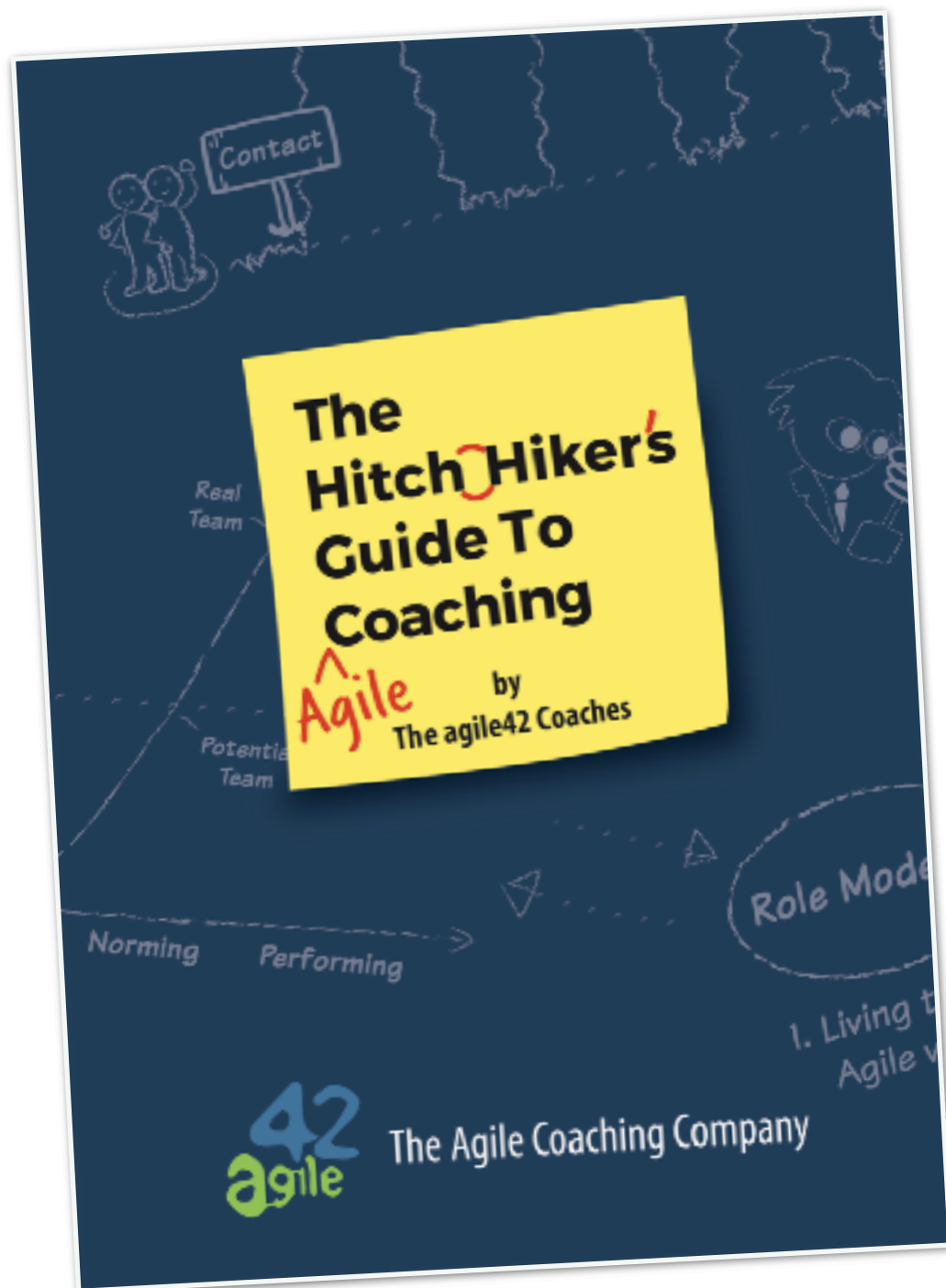
Self-explanatory,
self-contained

Forces you to "do
your homework"

Late commitment!

Enables
collaboration

Enables mentoring



<http://promo.agile42.com/guide-agile-coaching-ebook/>