

The Team-Knowledge-Model

Visualize team progress

MARK BREGENZER

Principal Consultant

Agile Methodology

Agile Experience:

- Agile Coach at Valtech since 2011
- Agile Coach since 2009
- Certified Scrum-Master since 2007

Software development Experience:

- Since 1997 as developer, lead developer, subproject leader...

Telecommunication and automotive



Scaling: Key for productivity and performance

Team states according to their ability to scale:

→ scaling potential

→ scaling state

→ scaling impediment

→ scaling progress

Successful scaling means continuous learning and know-how ramp up for all team members.

Know-how: Responsibility of the team

Teams often face these questions/challenges:

- How to handle learning and being productive?
- How to see who is expert, who needs help?
- How to focus in which direction a team develops?
- How to combine individual and team targets?

The Team Knowledge Model, based on flow model, can deliver some answers, but...

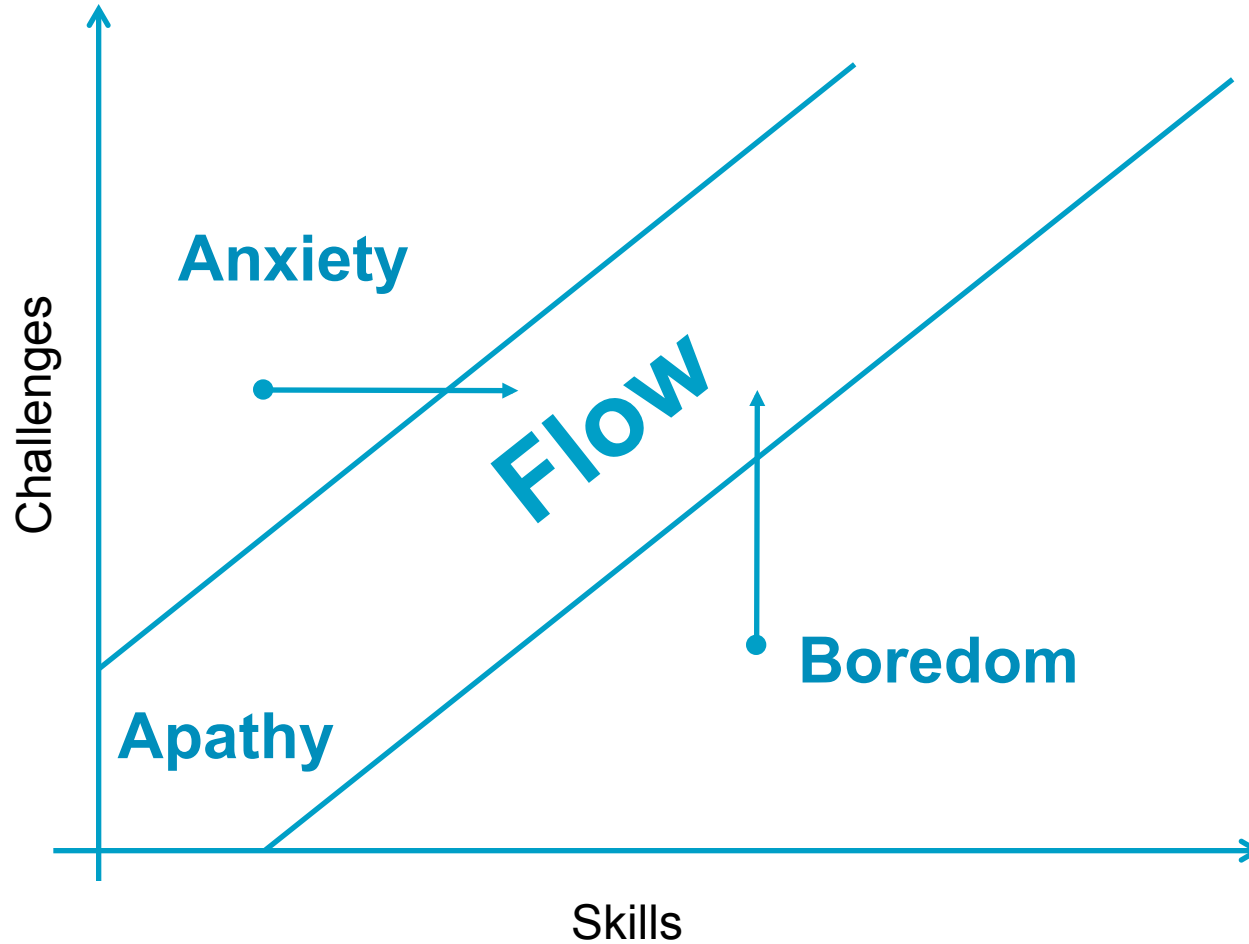
...what is the flow?

The flow in the world of working means the balance between requirements/challenges and skills.

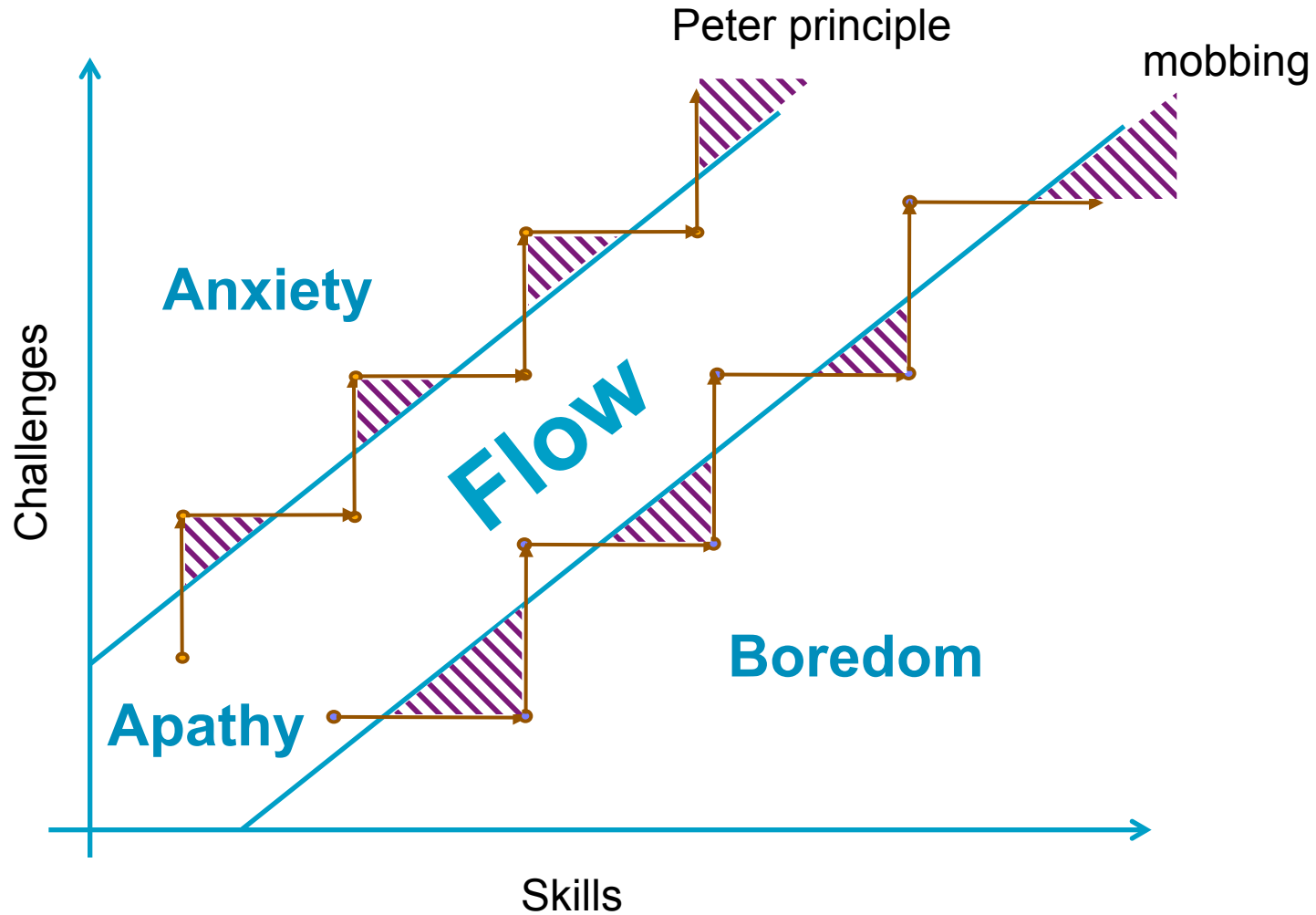
You are in the flow, if you can say:

I've got interesting work topics.
I'm working smoothly with unbroken continuity,
as in the manner characteristic of a fluid!

Flow Model for individuals (Mihály Csikszentmihalyi)



Learning at the edge of the flow (Joseph Pelrine)



Team Knowledge Model (TKM)

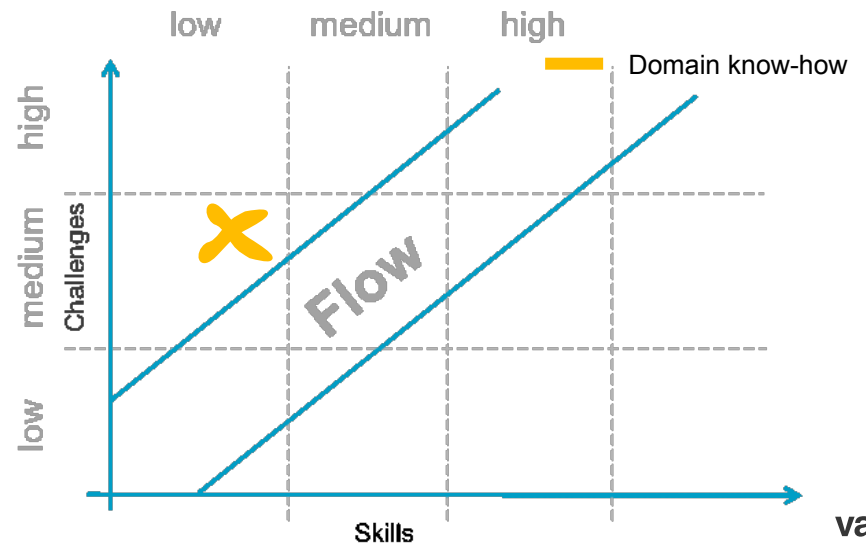
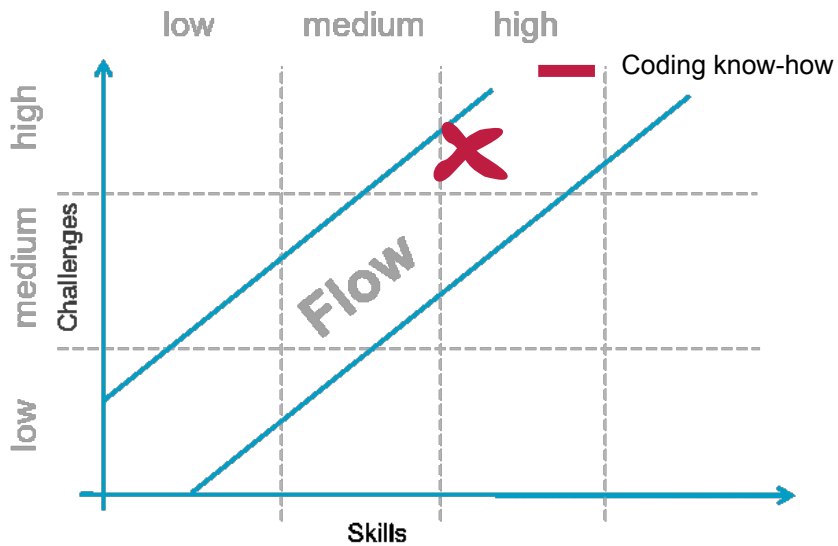
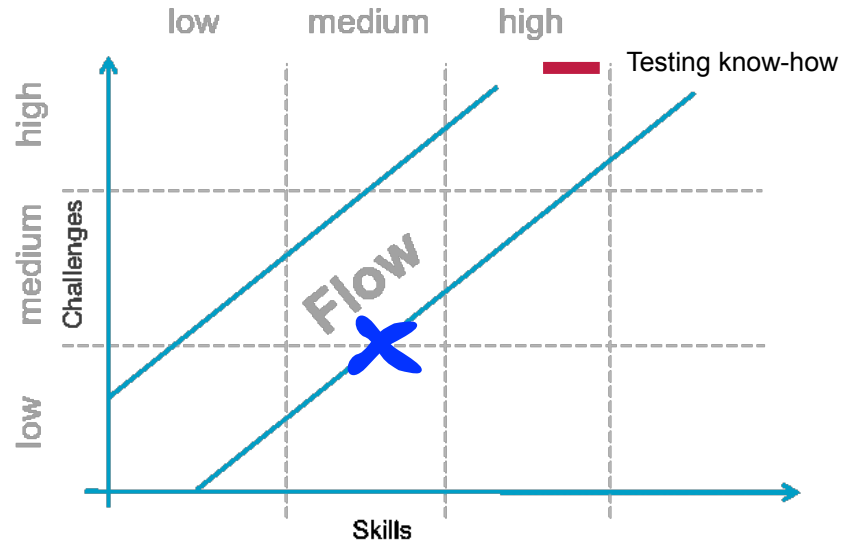
Only tracking velocity to display team improvements is not enough!

The Team Knowledge Model visualizes...

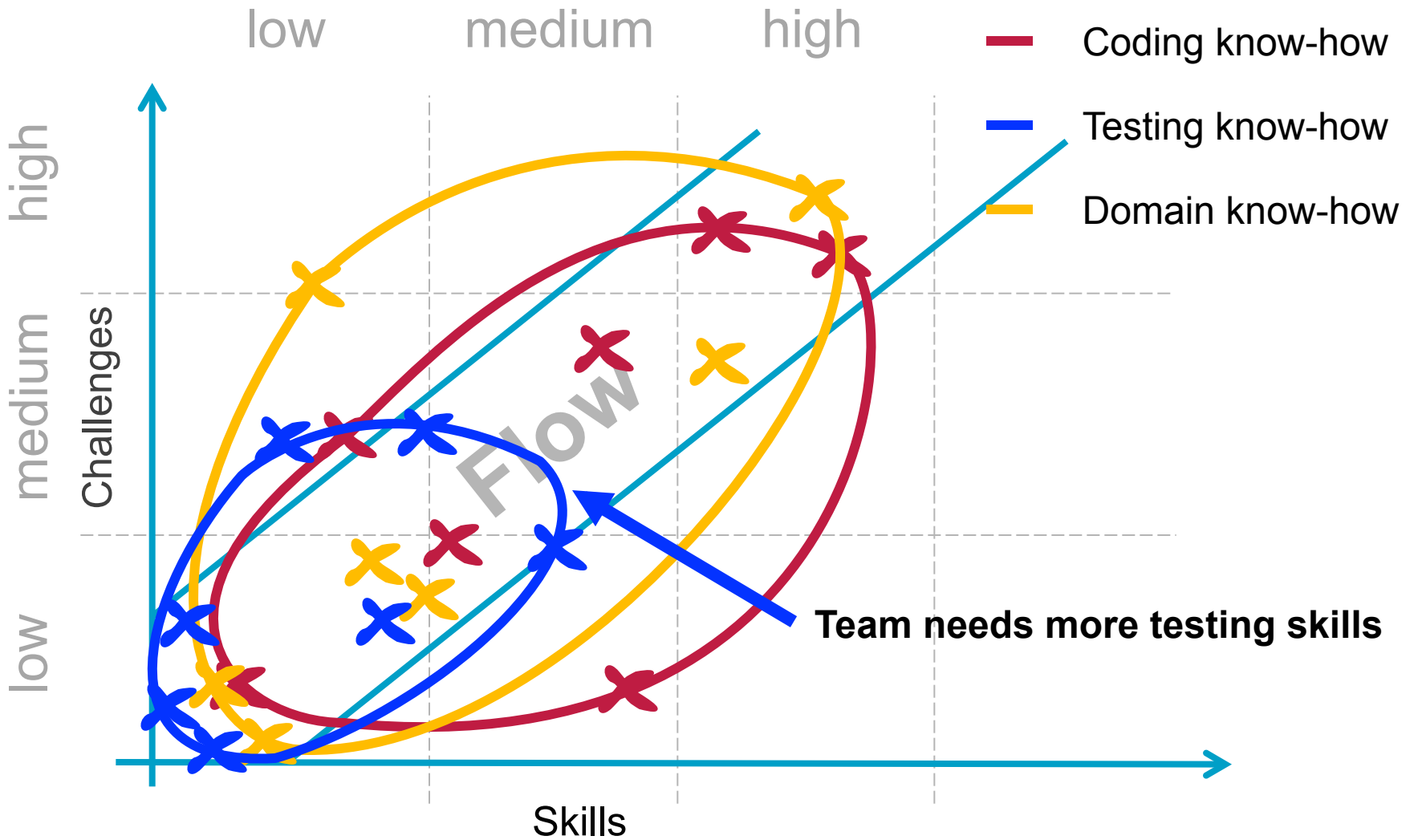
- the knowledge distribution within the team (scaling potential)
- balance of challenges and skills (scaling state)
- lack of knowledge in the team (scaling impediment)
- knowledge development over time (scaling progress)

TKM: Individual first

- Individual
- Independent
- Self estimated



TKM: Starting point

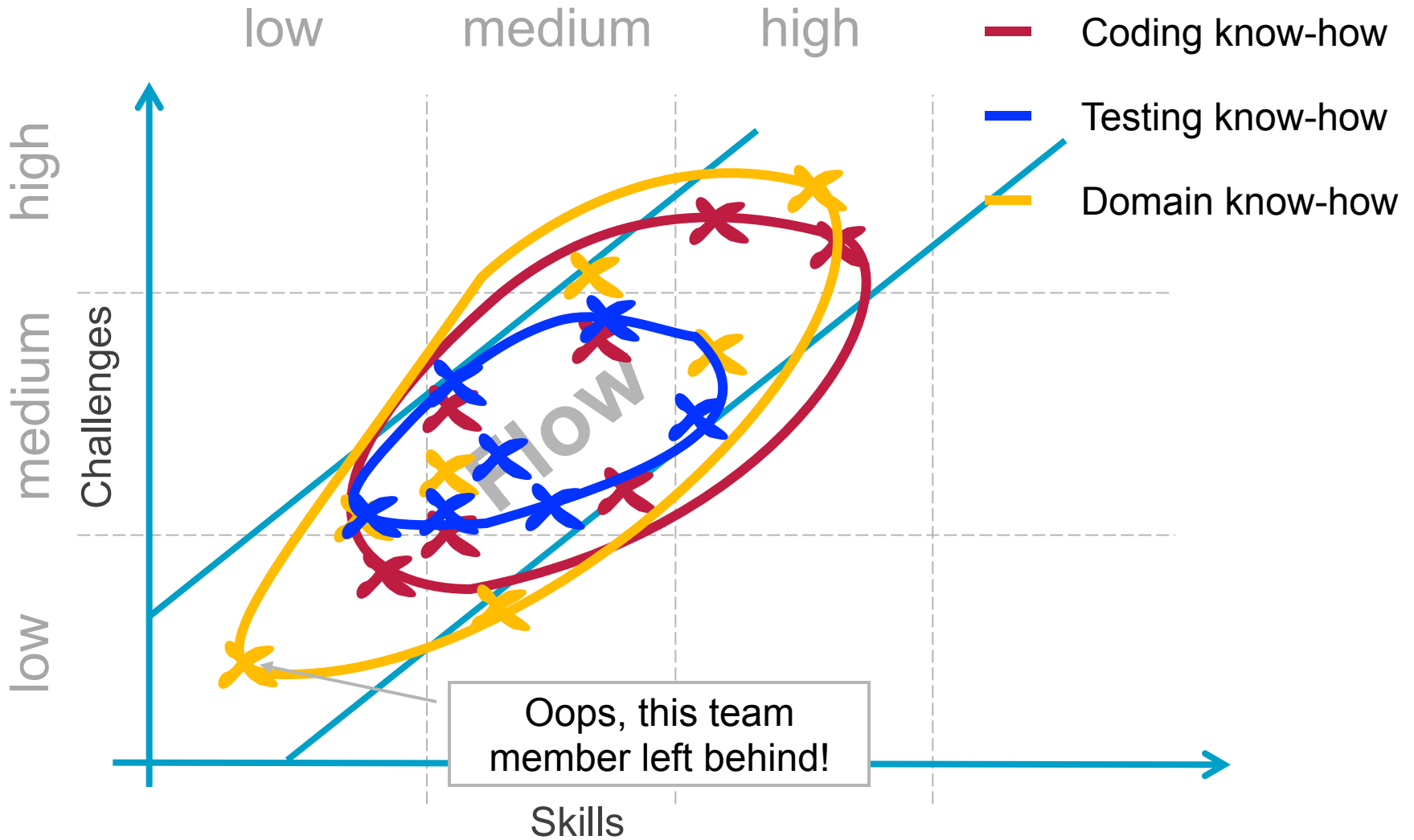


TKM: Working on results

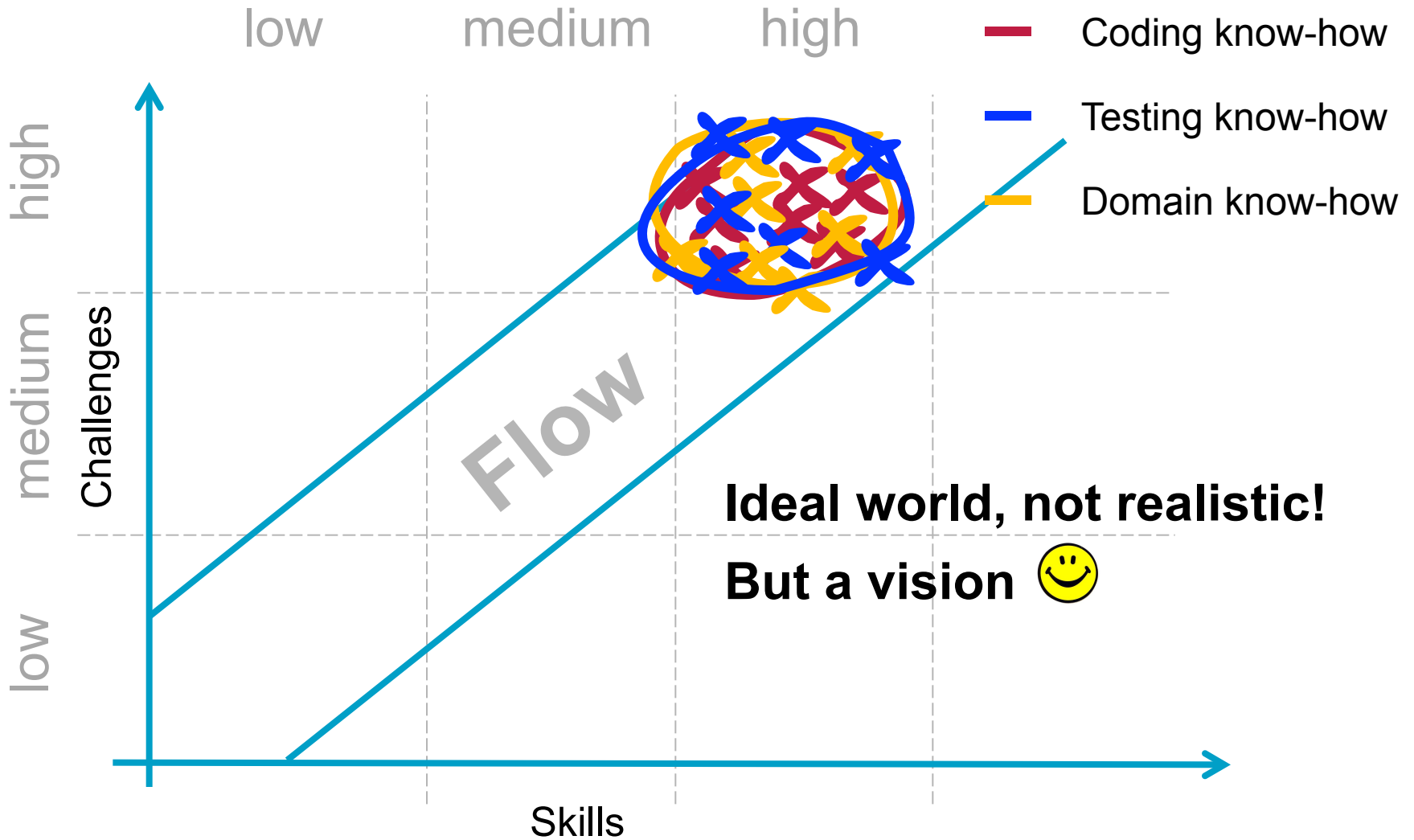
Improvement of team knowledge is a team target

- Find concrete measures
- Identify pairs for learning
- Agree on schedule for the update of the TKM

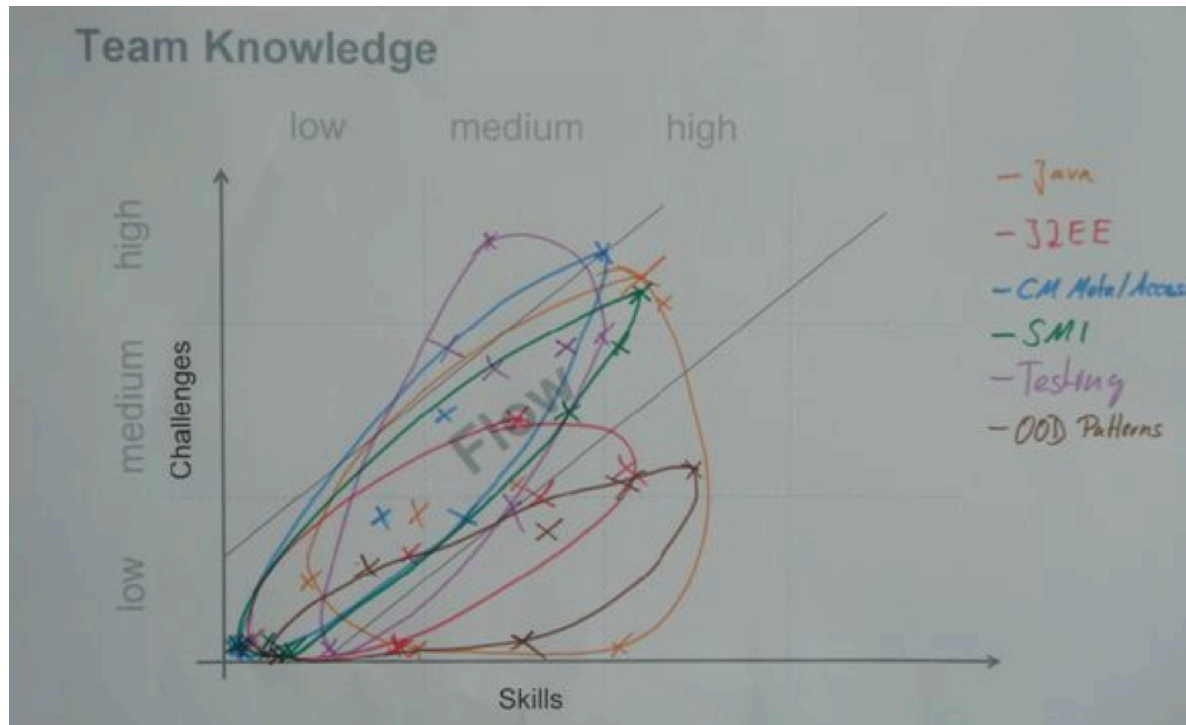
TKM: Progress



TKM: Ideal world (goal)



TKM: Real example



- This chart was created during the first team building workshop.
- Team setup was done with new and experienced employees.

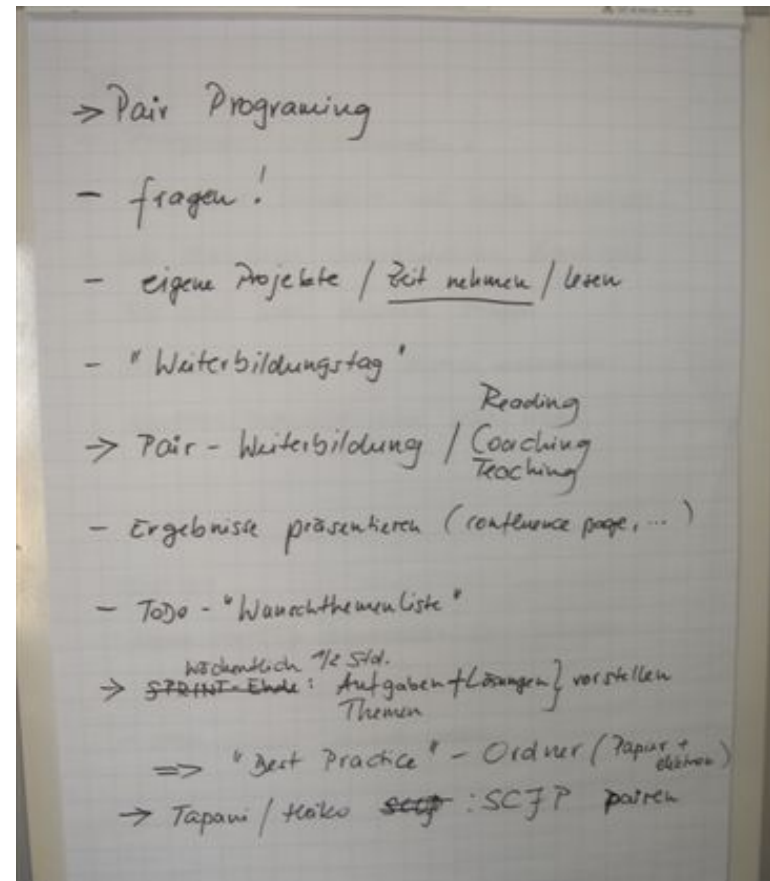
TKM: Working on results

→ Discuss results

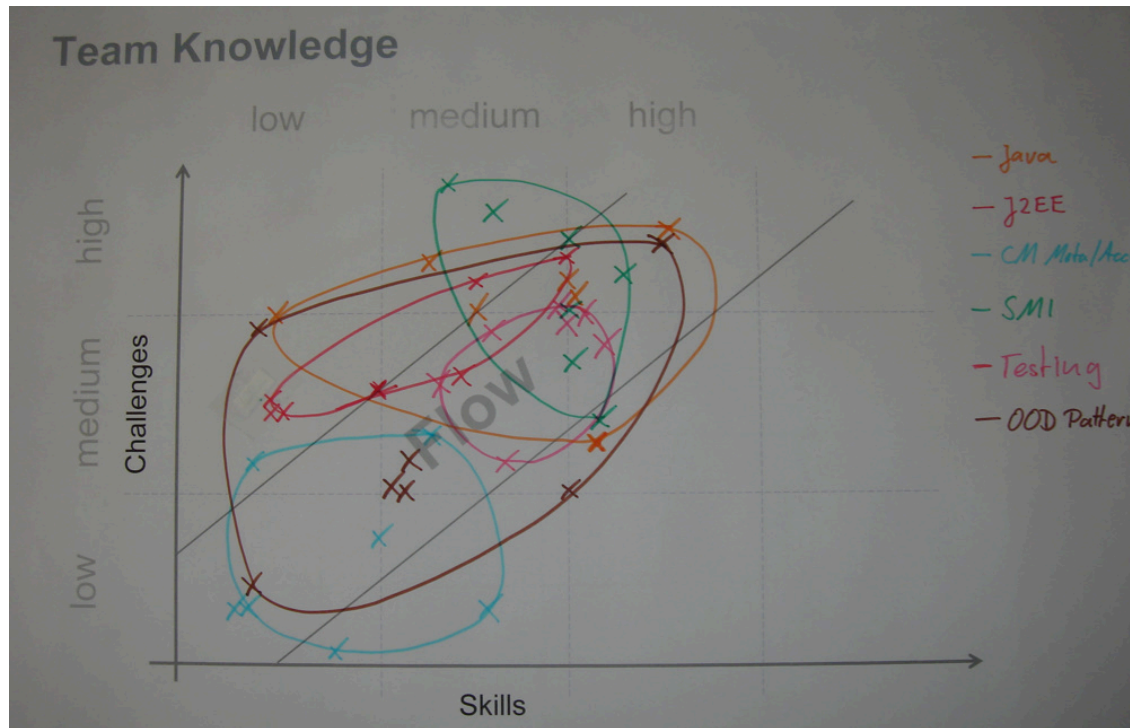
- Expert needed?
- Hugh circle?
- Smal circle?

→ Find measures to improve

- Pair-Programming
- Pair-Learning
- Questioning
- Learning day
- Wish list

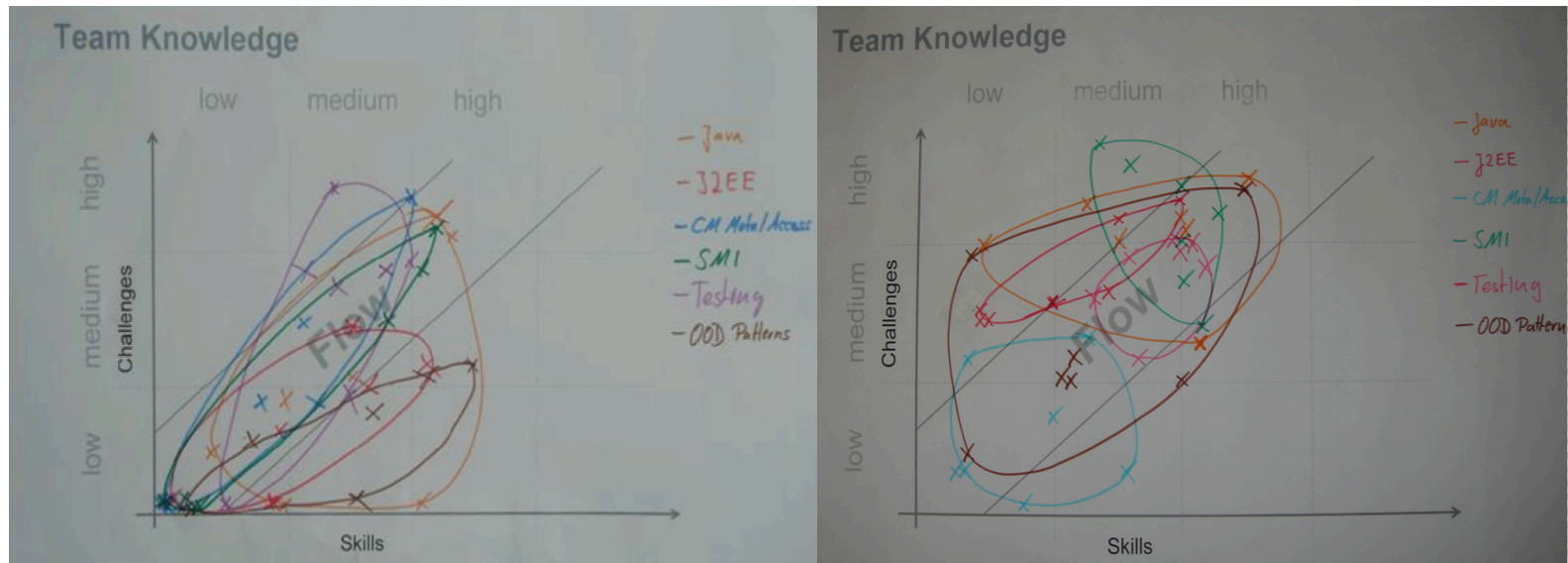


TKM: Real team progress



→ This chart was created after one year team working.

TKM: Visible improvement



Hot spots:

- Nearly all areas improved
- Team works mainly on SMI Interface, which is clearly visible
- Testing knowledge increases in the whole team quite well



Hints



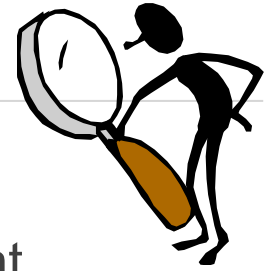
- Self estimation process is not an accurate science, but a good indicator for self confidence and mood
- Recommended frequency: 3,4 or 6 month
- Use maximum of six improvement areas
- Do not write names to the x

Problems



- Some people have problems with offering their self estimation to others
- TKM can be misused by management to judge people
- Finding right measures is more difficult, if it's done anonymously

Potentials



- Use the TKM in multi team projects to setup the right team structure
- Give target TKMs to the teams to coordinate know-how development in multi team projects
- Use Shu-Ha-Ri instead of low, medium and high



Thank you

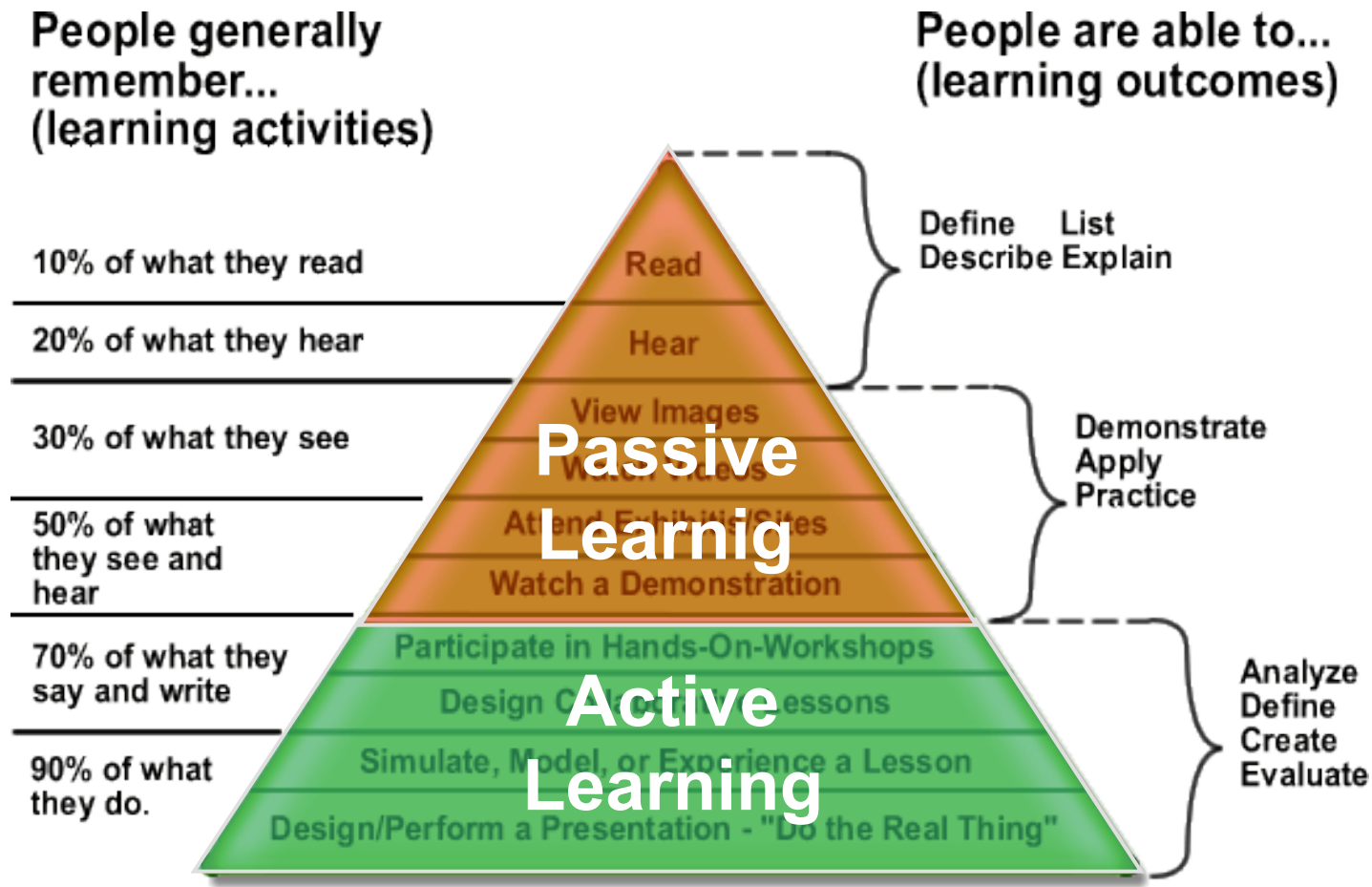
MARK BREGENZER
Principal Consultant

+49 172 2 855 788

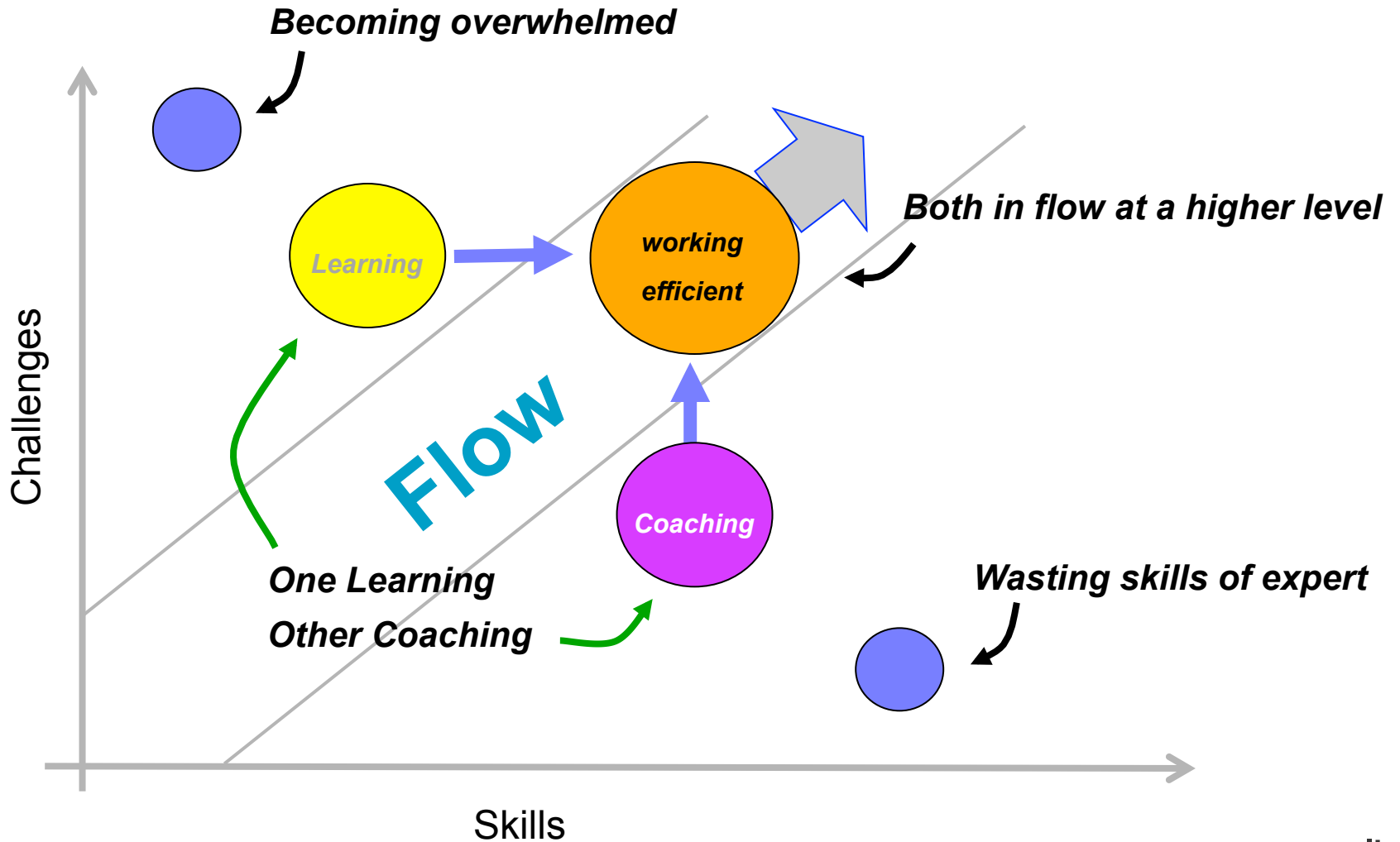
mark.bregenzer@valtech.de

→ valtech.de

How do we remember what we learned?



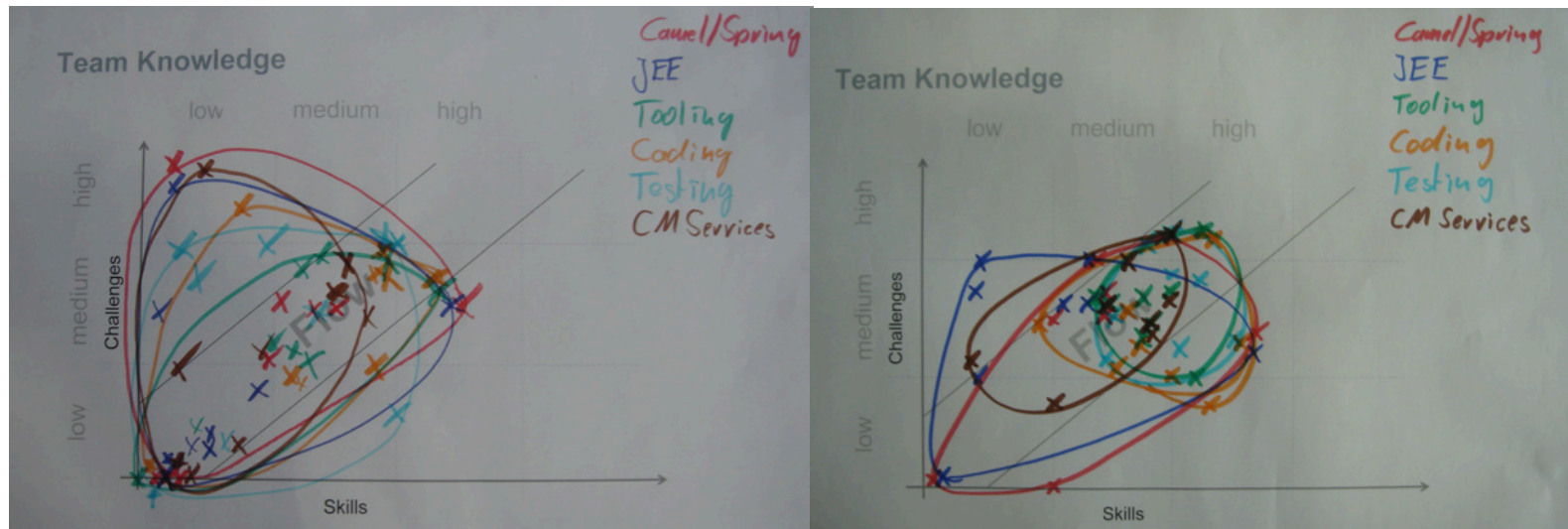
Use Pair Programming to improve individuals/teams



Some helpful learning practices

- Pair-Programming
- Coding-Dojos
- Community of Practice (cross-team-learning)
- Pair-Learning, Pair-Reading
- Reserve time for individual learning in each sprint
- Forums (e.g. an Agile-Design-Forum and introduce the S.O.L.I.D. design principals)
- Introduce ATDD and TDD (will lead to better domain, test and coding understanding)

Another example of a team knowledge ramp up



Hot spots:

- Team develops quite well at Testing, Coding and Tooling.
- Less people feel overwhelmed
- Scrum-Master and Interface Architect causes the three Xs at the bottom.

