



# INTEL EXPERIENCE AGILE TRANSFORMATION MODEL

Danuta Łuczak-Wieczorek, Stephan Lissel

# Introduction

- Danuta Łuczak-Wieczorek (Danka), Stephan Lissel
- Munich
- Intel Communication and Devices Group
- Agile Coaches, Quality Managers
  
- Agile Transformation in a 500+ people project in a mobile area
- Intel internal Aile coaches + Improv coaches

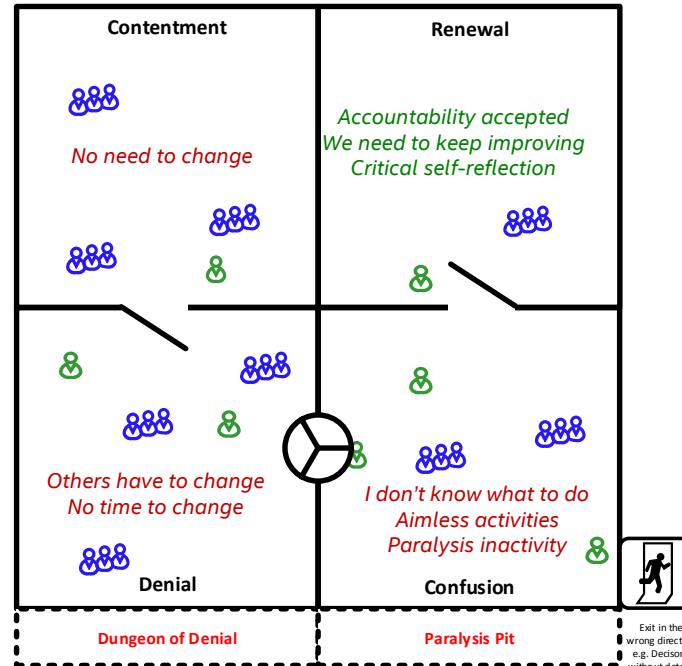
# Agile Transformation

Cynefin: Complex domain  
probe-sense-respond

House of Change <sup>1)</sup>

- Collect data and show need to change e.g. demand/capability gap, flow problems.

- Listen, ask questions & understand fears.
- Respect structure & roles.
- Allow some time to accept the need for change.
- Use Best Practice examples.



- Run effective retrospectives
- Deliver with reliable capability
- See transparency as a chance

- Have agreed that L&A helps to solve the problems.
- Call actively for coaching.
- Establish and qualify local coaches.
- Establish Agile tool box.

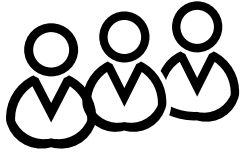
Management Support

Local Coach

1) Claes Janssen

Where to start?  
...and as we go...

# Management



Wants to see a return

# Teams

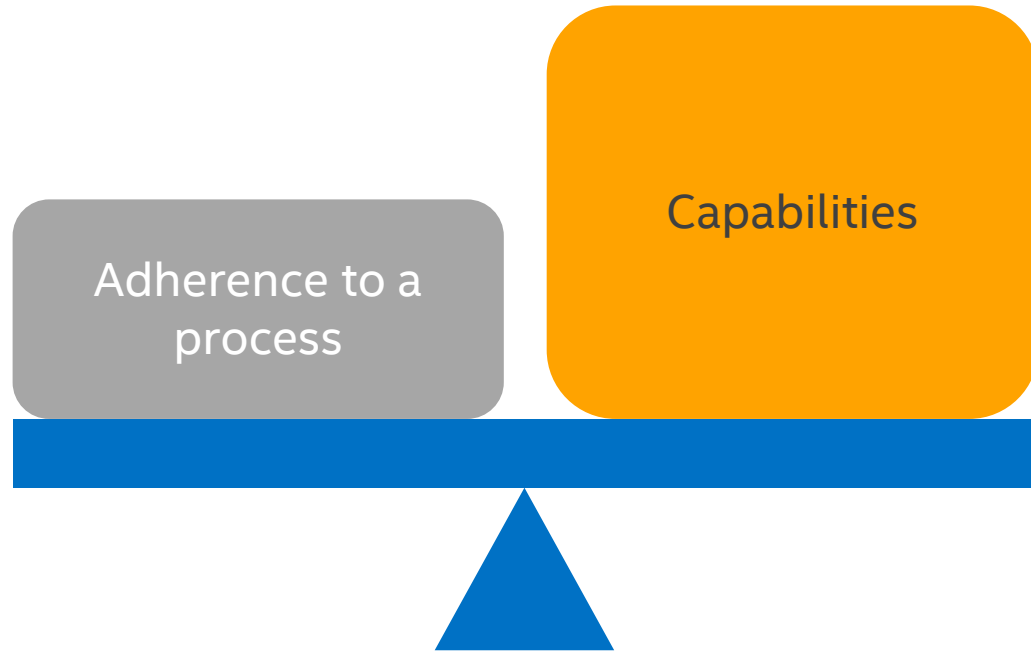


Need to assess their maturity

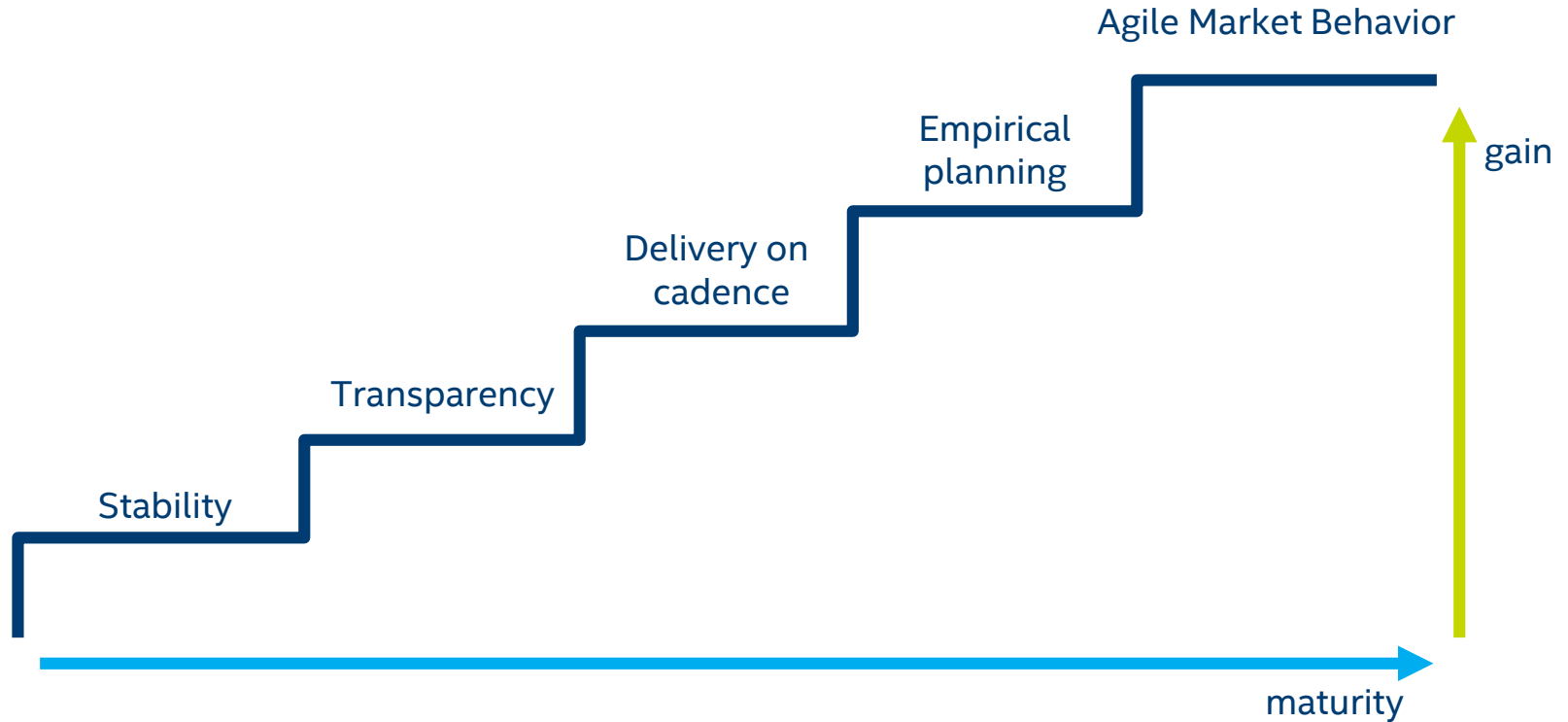
# Agile Coaches



What can be improved?  
When to let team go?



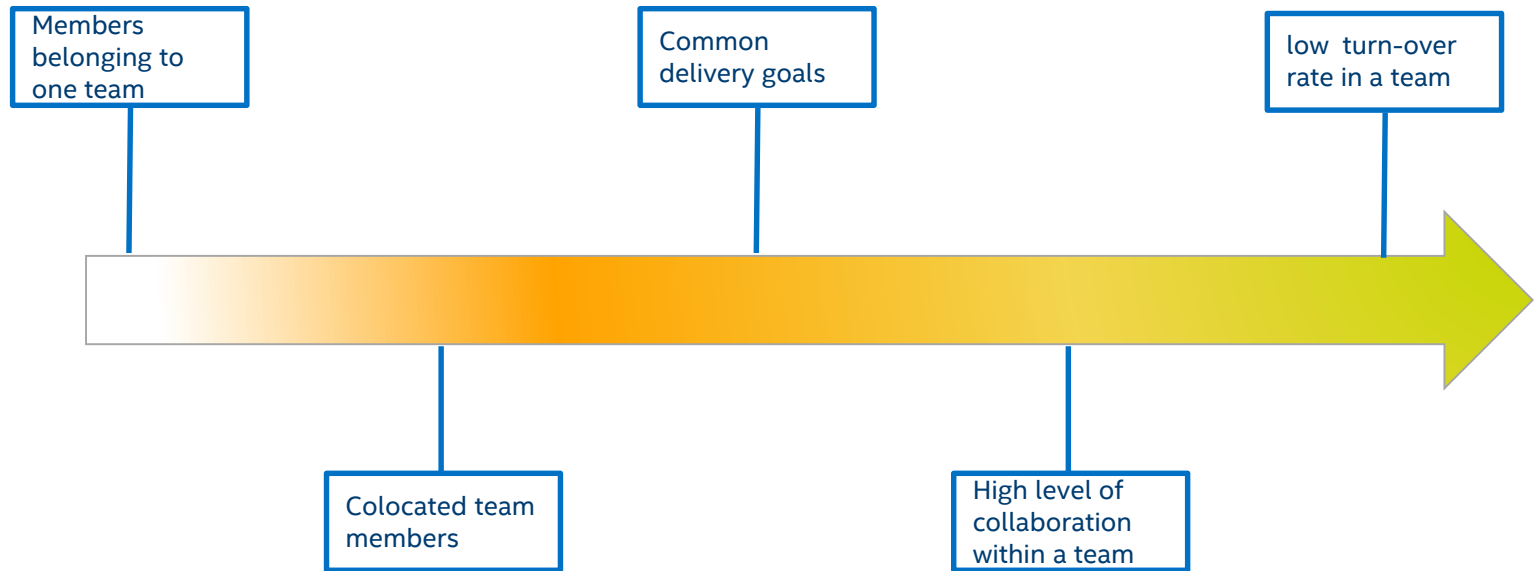
# Capability Model



# Stable Teams

Team learning isn't possible with short-lived teams.

A team (engineering as well as management) which stays intact for a longer period of time.

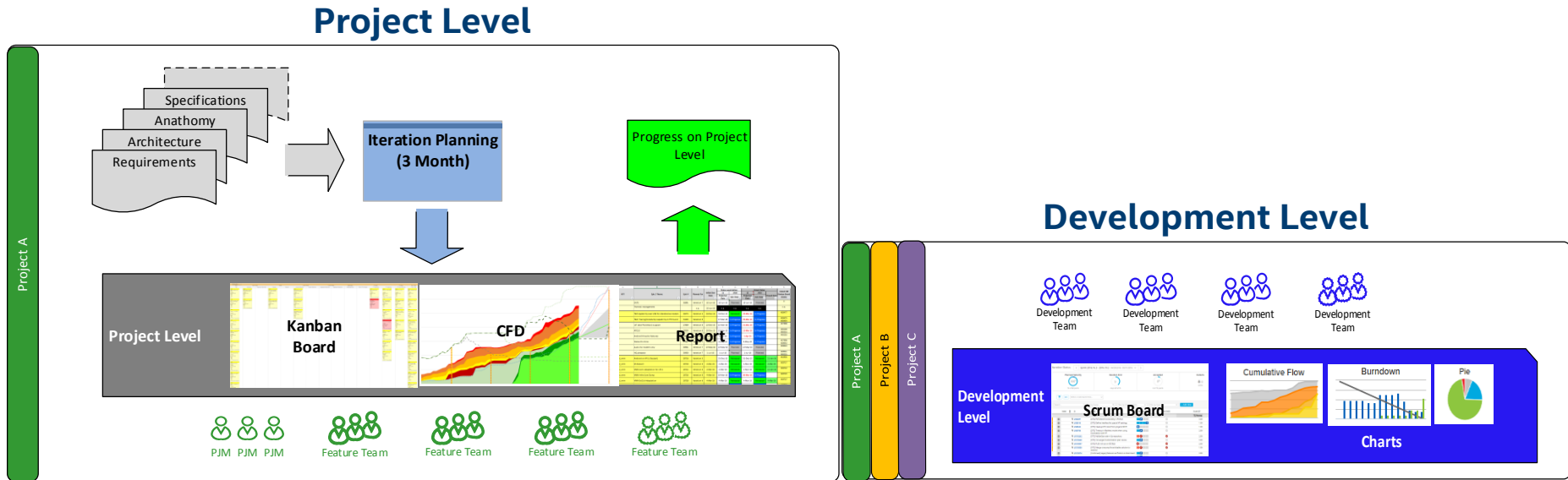




# Transparency

Project Level: Capability and Delivery of Feature Development not transparent.  
Development Level: Overload due to unknown team's capability.

- Known and transparent capabilities, impediments and successes.



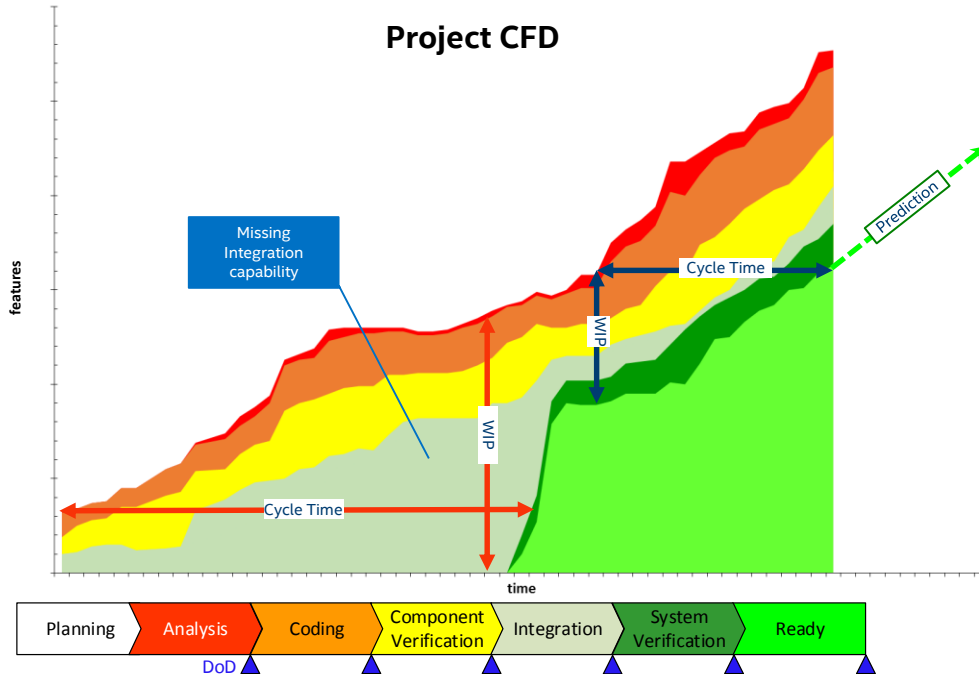
# Delivery on Cadence

Batch size development leads to substantial “cost of delay”

- Deliver value on short and fixed cadence in a sustainable mode.

## Project Level

- Visualize mid-to long term flow.
- Information on throughput, cycle time & expected finalization date
- Show up bottlenecks
- Feedback on effect of changes
- Established for management reporting.



## Feature Level

- CFD for each Feature Team.
- Used for regular alignment meeting.
- Feature Team Lead responsible.

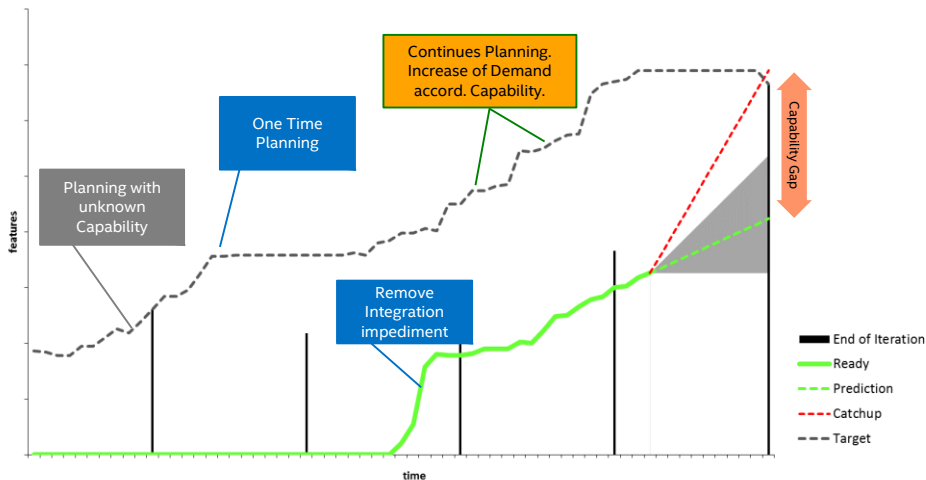
# Empirical Planning

Frequent re-planning due to unknown delivery capability.

- Realistic planning decisions based on measured delivery capabilities.

## Demand

Demand is defined in terms of features, priorities and deadlines.



## Capability

Capability is defined in terms of throughput of features delivered.

## Corrective Actions

If the Capability is not sufficient to satisfy the demand, decrease the demand or increase the capabilities.

# Key Learnings for evolutionary Agile Transition

- Simply rolling out Scrum is not enough
- It's evolutionary, not revolutionary
- Be aware of where you are
- Probe, sense, respond
- Define the desired capabilities (not processes), and focus on enabling them

Backup

# Key Learnings for evolutionary Agile Transition

- Simply rolling out Scrum is not enough
- Establish a model like our “Capability Model” to measure progress and demonstrate value added
- Embed Agile Transition in a Change Model
- Team collaboration is critical for success
- Build Guiding Coalition to understand current state, limitations and opportunities of the Agile transition
- Support and encourage line management in adopting appropriate improvements
- Collaborate with other initiatives to align and strengthen Agile transition
- Build Agile network with Core Team and Local Agile Coaches
- Actively promotes success stories and lessons learned

# Stability

1	Are you part of multiple teams?	
2	On how many locations is your team based?	
3	Does your team have common delivery goals?	
4	How much do you collaborate in the team to fulfil your tasks?	
5	How many members has your team changed over the past 6 months?	
6	Did these changes impact the efficiency of your team?	

# Transparency

1	Is it visible to all team members what everybody is working on?	
2	Is progress/status information about your team easily available to all team members and easy to understand? (also to external people)	
3	Is progress/status information about your team easily available to external people?	
4	Are impediments/risks visible and are these actively being worked on?	
5	Are dependencies (also to other teams) visible and easily accessible?	



# Sustainable delivery

1	Do you have common Definition of Done?	
2	Are you familiar with the Acceptance Criteria of the receiving end?	
3	Do you have a backlog with prioritized work items?	
4	Do you break down backlog items so they can be delivered in short iterations (normally less than 4 weeks)?	
5	Do you have automated tests to make sure you find and fix most bugs before full system integration and delivery to Mainline?	
6	Do you involve other teams (SE, Dev, VP, SIT) to ensure that you deliver to Mainline in a sustainable pace?	

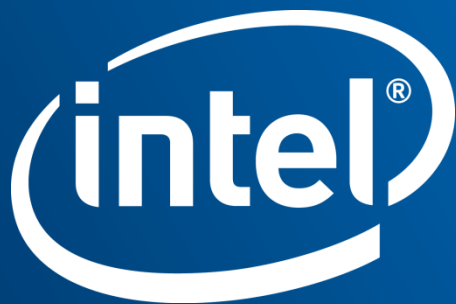
# Empirical Planning

1	Are you able to deal with unplanned work without blowing up your plan?	
2	Do you gather data about your delivery capability?	
3	Are you able to forecast deliveries into the future with a high level of certainty?	
4	Do you adjust your plans based on historical data that shows your delivery capability?	

# Agile Market Behavior

1	Do you as a team learn continuously from your mistakes?	Orange
2	Do you as a team recover easily from failure?	Orange
3	Do you think that you're adding the right value for the customer?*	Yellow
4	Do you think that you're always delivering the most important features first?*	Light Green
5	Are you able to deliver new features faster than the competition?*	Light Green

*\*This question is applicable to a team on system level rather than component/sub-component level*



## Intel Communication and Devices Group