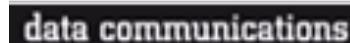
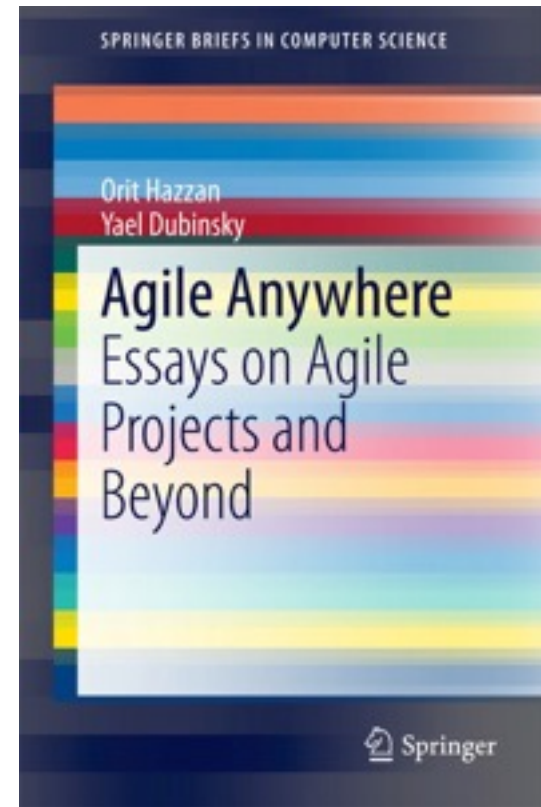
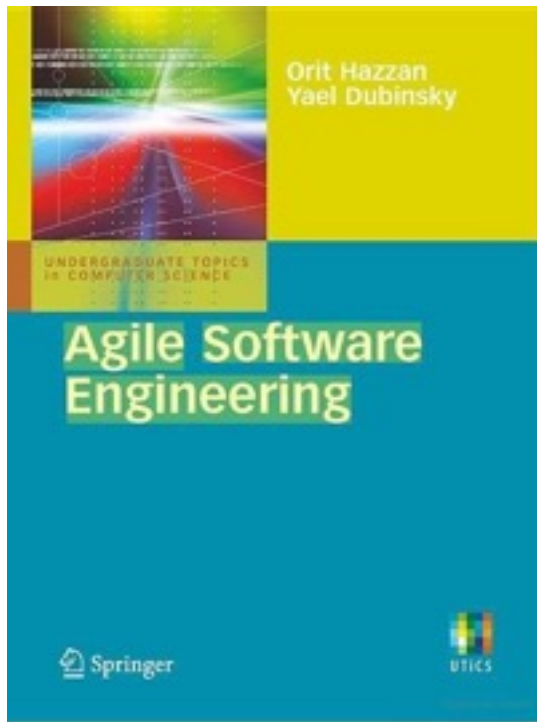
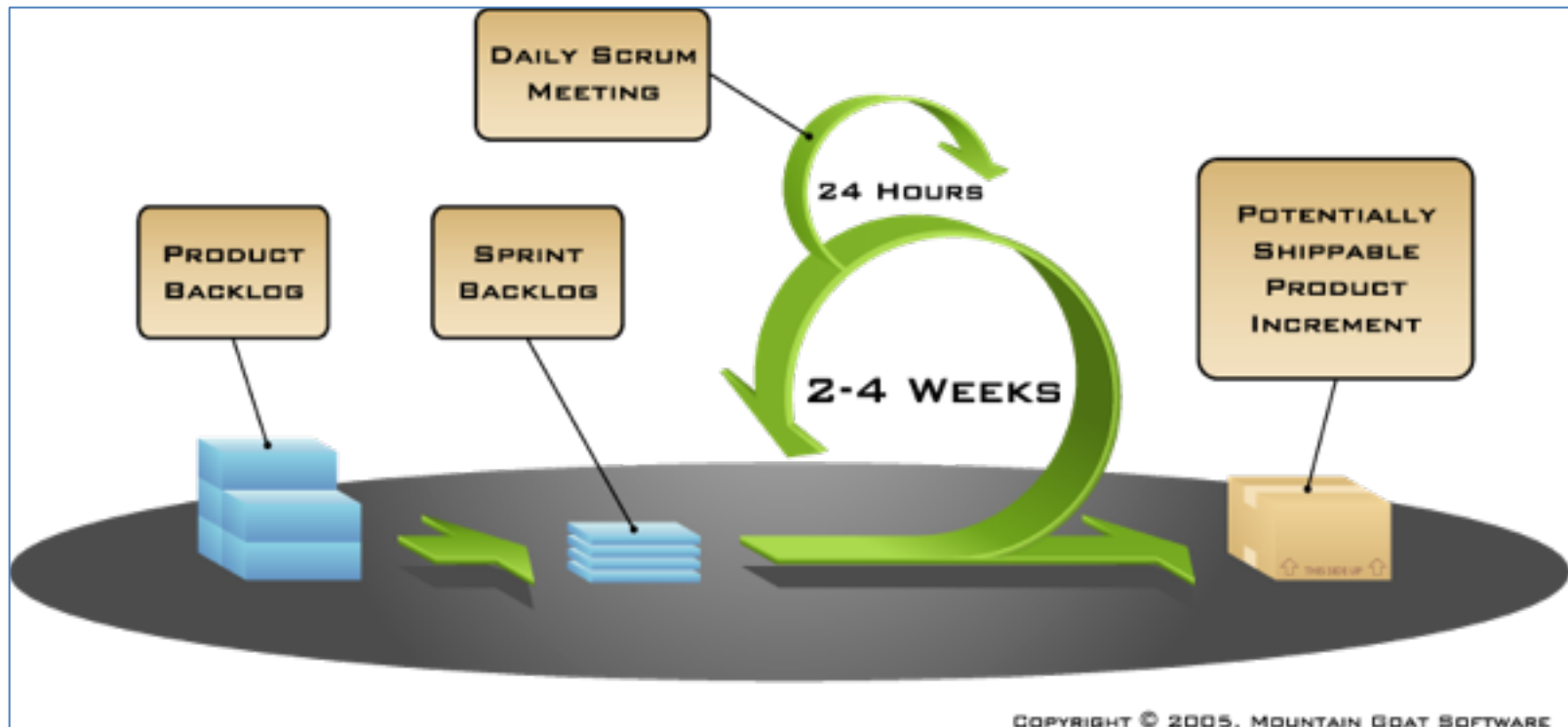


IBM Jazz Collaborative Lifecycle Management: Agile Anywhere

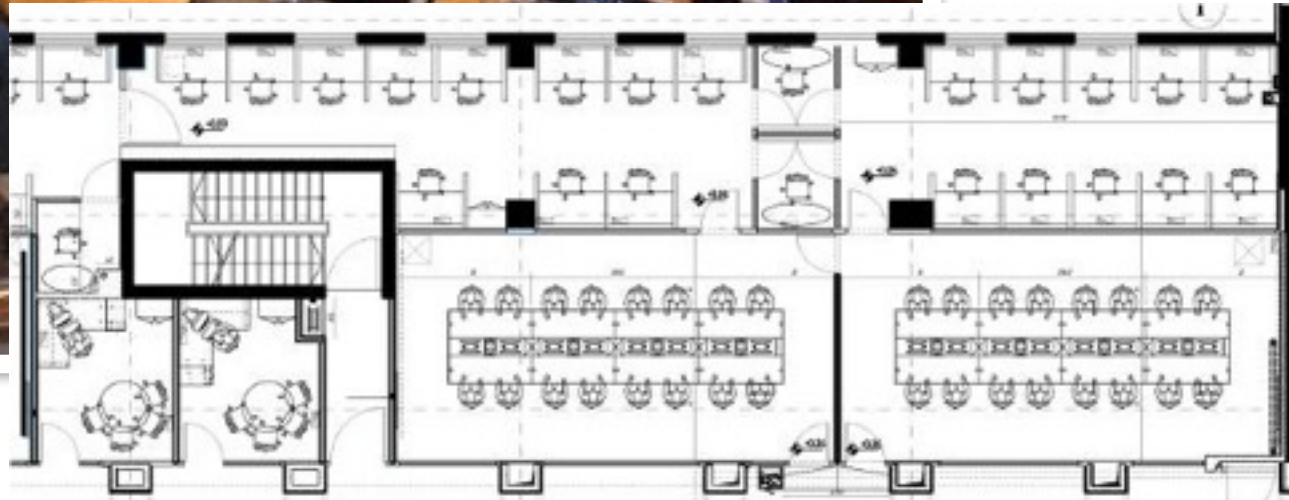
Dr. Yael Dubinsky
IBM Research - Haifa
Nov 2015



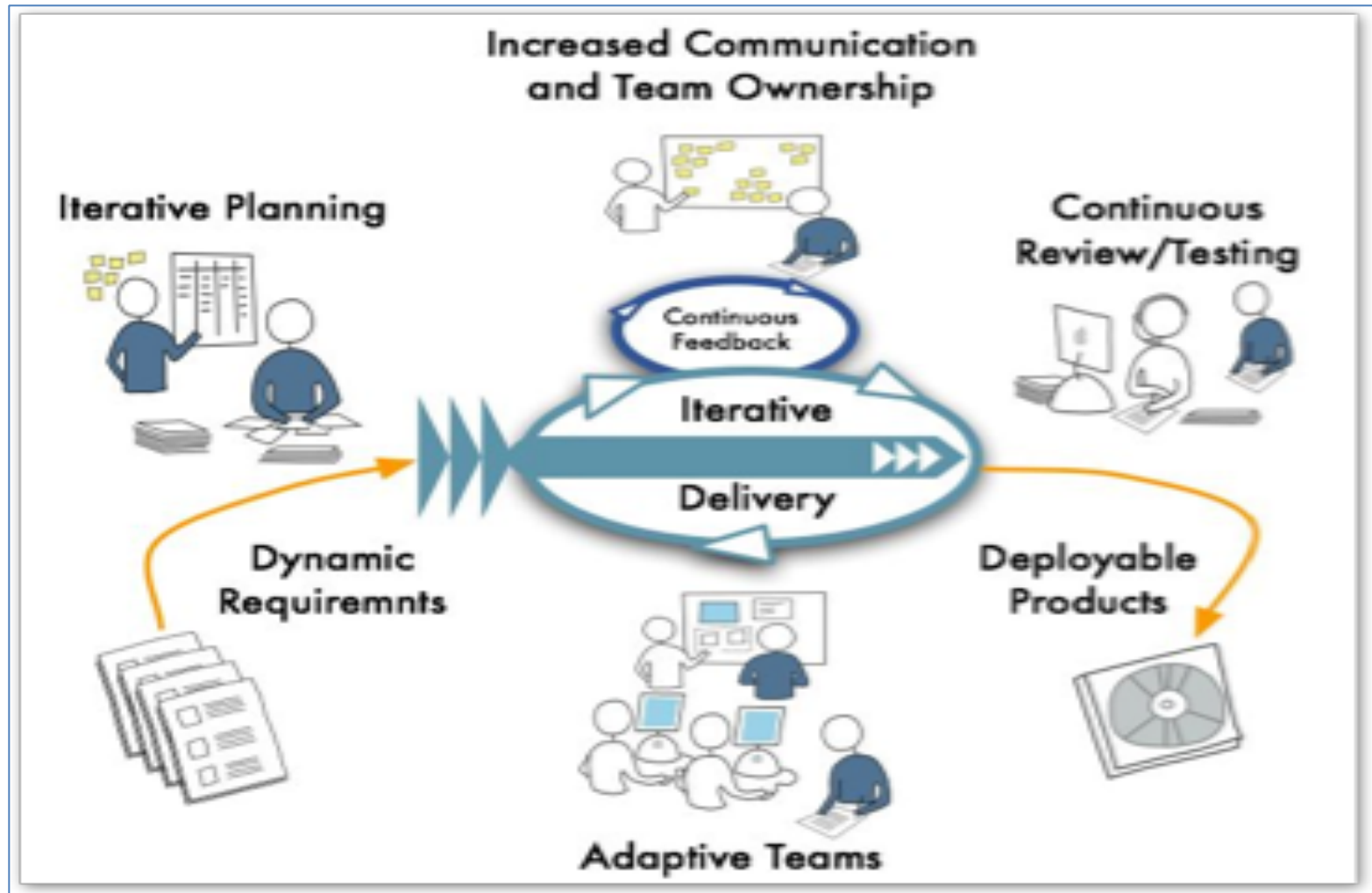
Time



Space



People



Lean

The Agile: Scrum Framework at a glance

Inputs from Executives,
Team, Stakeholders,
Customers, Users



Sprint end date and team deliverable do not change



Are We Agile?

The customer role?

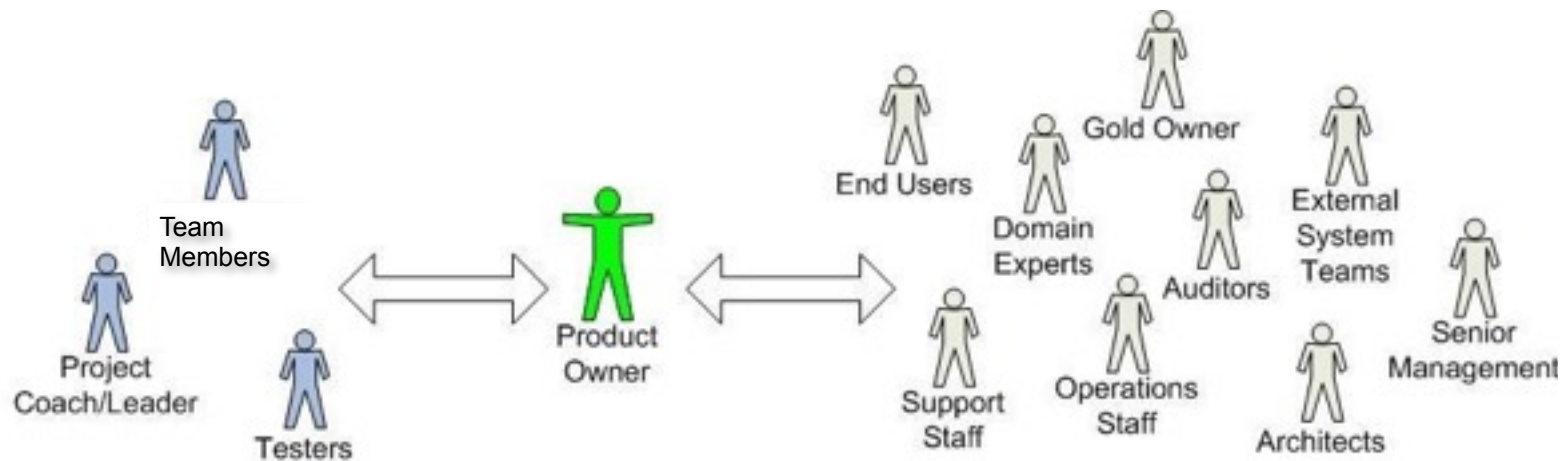
Iteration management?

Continuous high-quality delivery?

Learning and retrospective?

Customer in the 2-Week Loop

- The customer role
 - Accountable for requirements prioritization
 - Approves plan
 - Defines acceptance criteria
 - Gives feedback
 - Share vision



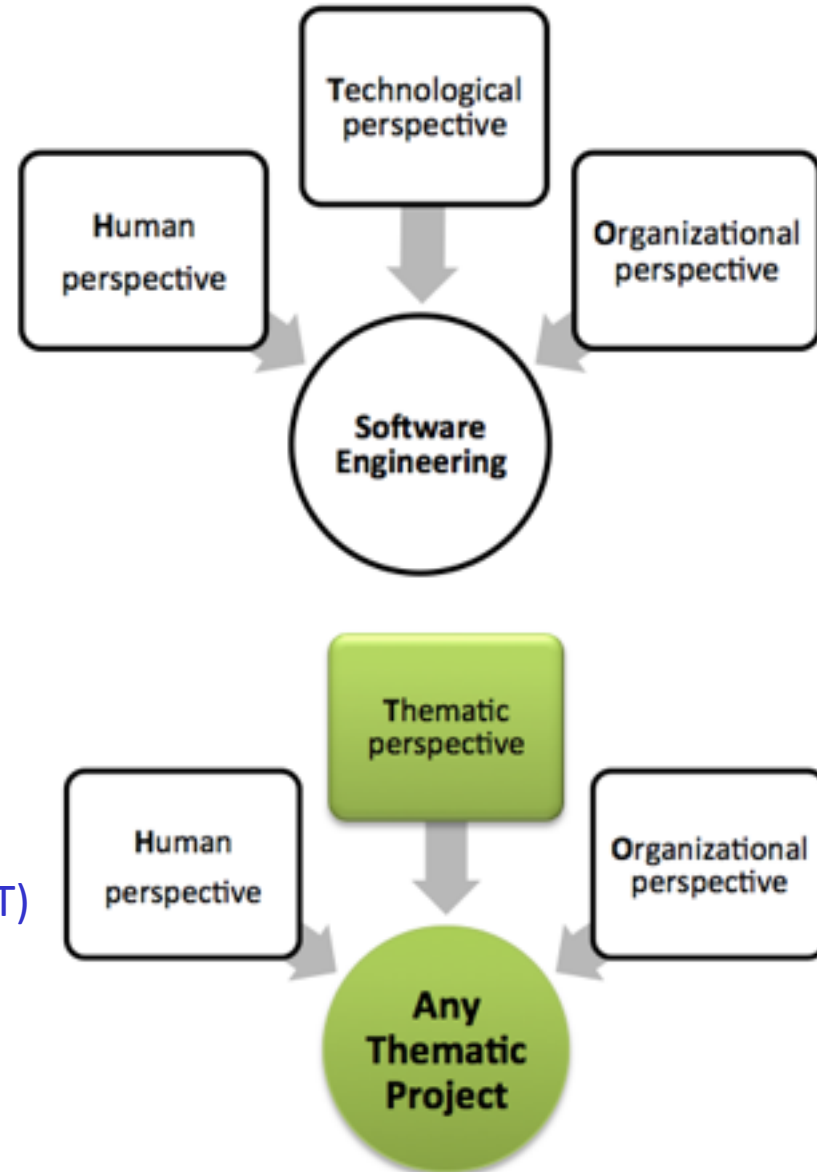
The Agile Development Approach

What to focus on?

- The value system
 - Communication and feedback
 - Courage and respect
 - Simplicity
- The practice
 - Iteration management
 - The customer role
 - Continuous high-quality delivery
- The conceptual change
 - Learning and retrospective

Agile Anywhere

- The value system
 - Communication and feedback
 - Courage and respect
 - Simplicity
- The practice
 - Iteration management (H)
 - The customer role (O)
 - Continuous high-quality delivery (T)
- The conceptual change
 - Learning and retrospective



Continuous High-Quality Delivery

- Automated testing to support integration
- The *Done* procedure

Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

<http://agilemanifesto.org/>

Agile Software Development: Organizational Perspective

- Short releases
 - business day, coping with changes in requirements, time estimations
- Whole team
 - development environment, daily stand-up meetings, roles
- Customer involvement
- Metrics

Agile Software Development: Technical Perspective

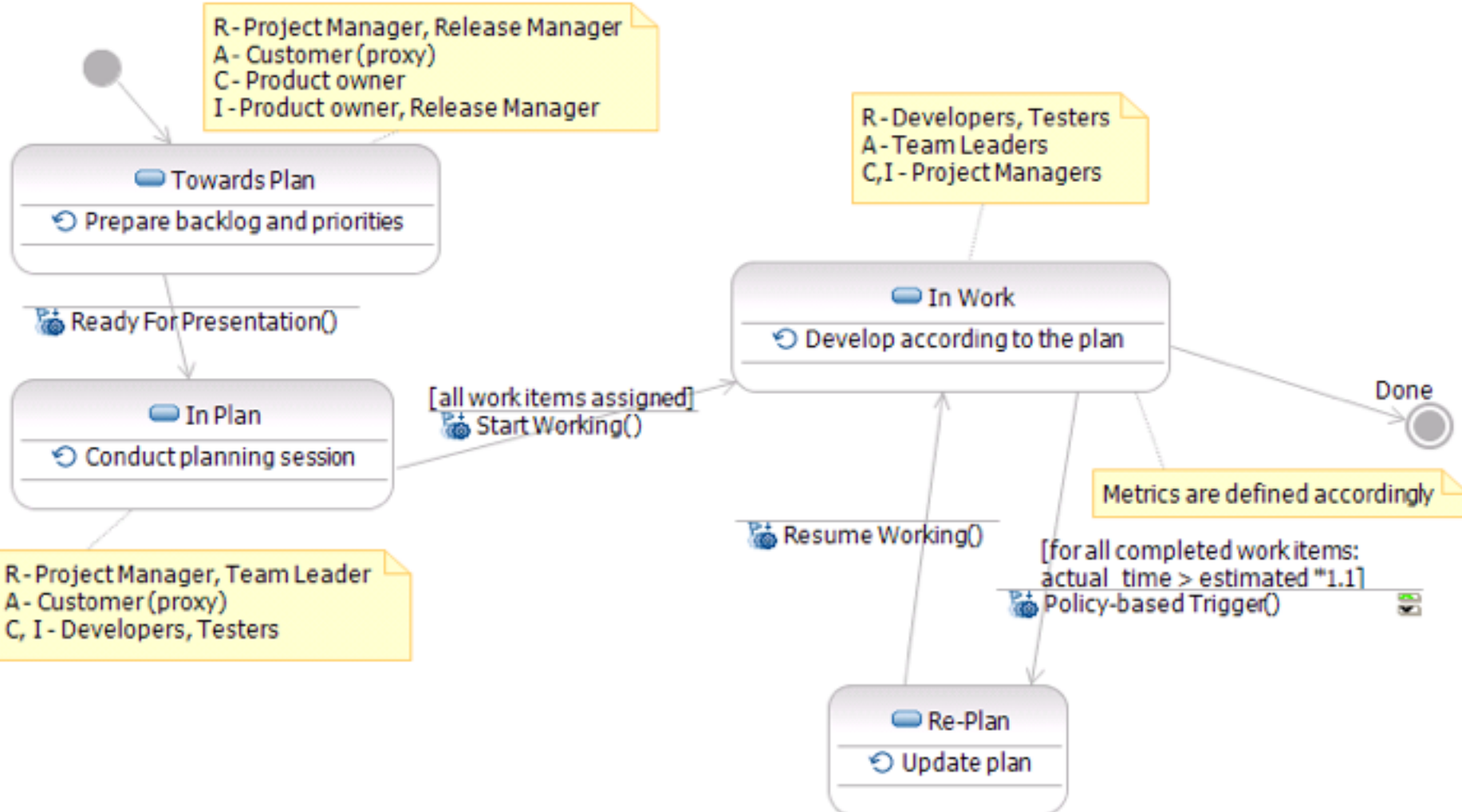
- Automated testing
 - part of the development process
 - unit test, test driven development
 - acceptance tests
- Coding
 - refactoring
 - pair programming
 - continuous integration

Agile Software Development: Human Perspective

- Value system
 - Communication, Feedback , Courage, Simplicity, Respect
- Learning process
 - Gradual learning, coping with changes, abstraction
- Reflection / retrospective
 - Reflective practitioner

The Iteration Plan Lifecycle

Iteration Plan



Iteration Plan: Decision Rights and Policies

Activities	Project Manager	Release Manager	(proxy) Customer	Product Owner	Developers	Testers	Team Leaders
Prepare backlog and priorities	R	RI	A	CI			
Ready for presentation()	R	RI	A	CI			
Conduct planning session	R		A		CI	CI	R
Develop according to plan	CI				R	R	A
Update plan	R		A		CI	CI	R

Policies and Metrics	Type	Condition	Comments
Start working ()	Auto (guard)	All Work Items are assigned	
Actual time versus Estimations	Manual		Team members fill in data
Re-Plan	Auto (trigger)	For all completed Work Items: Actual > Estimated * 1.1	

Agile Retrospective Technique

- Only one specific problem is discussed at each retrospective meeting.
- The discussed problem should relate to the development process, or the developed product.
- The subject is chosen in advance by the moderator (after informal consultation with other team members), and presented at the beginning of the retrospective meeting.
- The retrospective session does not exceed one hour.
- The whole team is required to attend the retrospective .
- The retrospective may be an open discussion, or use some structured problem solving technique.
- Everyone is proactively encouraged to speak, but is not required to do so.
- Team members are encouraged to speak their own opinions, as honestly as they see fit.
- The moderator records important insights and proposed action items that surface during the meeting.
- The moderator summarizes the meeting by reading to the team the decided action items.
- The decided action items are effective immediately. They are actual changes in day-to-day team operations that should reduce the debated problem.
- The moderator publishes the main insights and action items to the team soon after the retrospective .

Retrospectives along the Project Timeline

#	Topic	Brief Description
1	Time management	Using Covey's concept of <i>First Things First</i> [31], the team decides to redefine activities that were identified as 'urgent and important' to be 'not urgent and important' by planning in advance.
2	Knowledge management I	The team raised the problem of lack of knowledge in various subjects of their project. The team decides to map their expertise areas and the team member responsible for each area.
3	Estimate vs. actual	The team members shared their feelings regarding the gap between their estimations and actual performance. They decided to have a one hour meeting every week to share feelings and talk about team issues.
4	Iteration planning I	The team raised problems that relate to the way tasks are broken down and planned. They decided that the team member who is in charge of a task will break it down and estimate its development time, and that only the customer can approve a non-planned task to be committed to the iteration plan.
5	Iteration planning II	The team worked to improve its understanding of the essence of context switches that occur during their work and ways to overcome these. No decision was made.
6	Connections to other disciplines	The team members shared their concerns that the other disciplines of the project are not committed to the process changes that the software team has adopted. Following a guided group activity, the project management changes the format of the project morning meetings.
7	Knowledge management II	Team members claimed they do not understand some of the information presented in the daily stand up meetings. Team members prepared and presented their knowledge on main topics.

8	Stakeholders	Team members identified the project stakeholders. No decision was made.
9	Automated testing	Team members discussed ways to adopt the practice of automated testing. No decision was made.
10	Definition of task completion	The team discussed the definition of 'done' with respect to a task. Following a guided group activity, the team leader agreed to continue refining the definition.
11	The lab I	The team discussed issues related to the use and physical setup of the lab. They suggested deploying a tool to schedule lab stations to avoid idle time.
12	The lab II	The team worked on defining the requirements of a tool to schedule lab stations in advance.
13	Release delivery	The team suggested discussing the pressure that exists before a release and ways to cope with and improve the situation.
14	Knowledge management III	The team reflected on what topics should be learned and how to arrange the learning process. It was decided to focus on the system design in the next few retrospective sessions.
15	Process changes	The team discussed the process changes they went through since the beginning of 2009.

THANKS

DUBINSKY@IL.IBM.COM