

# Overview of Agile

Naresh Jain

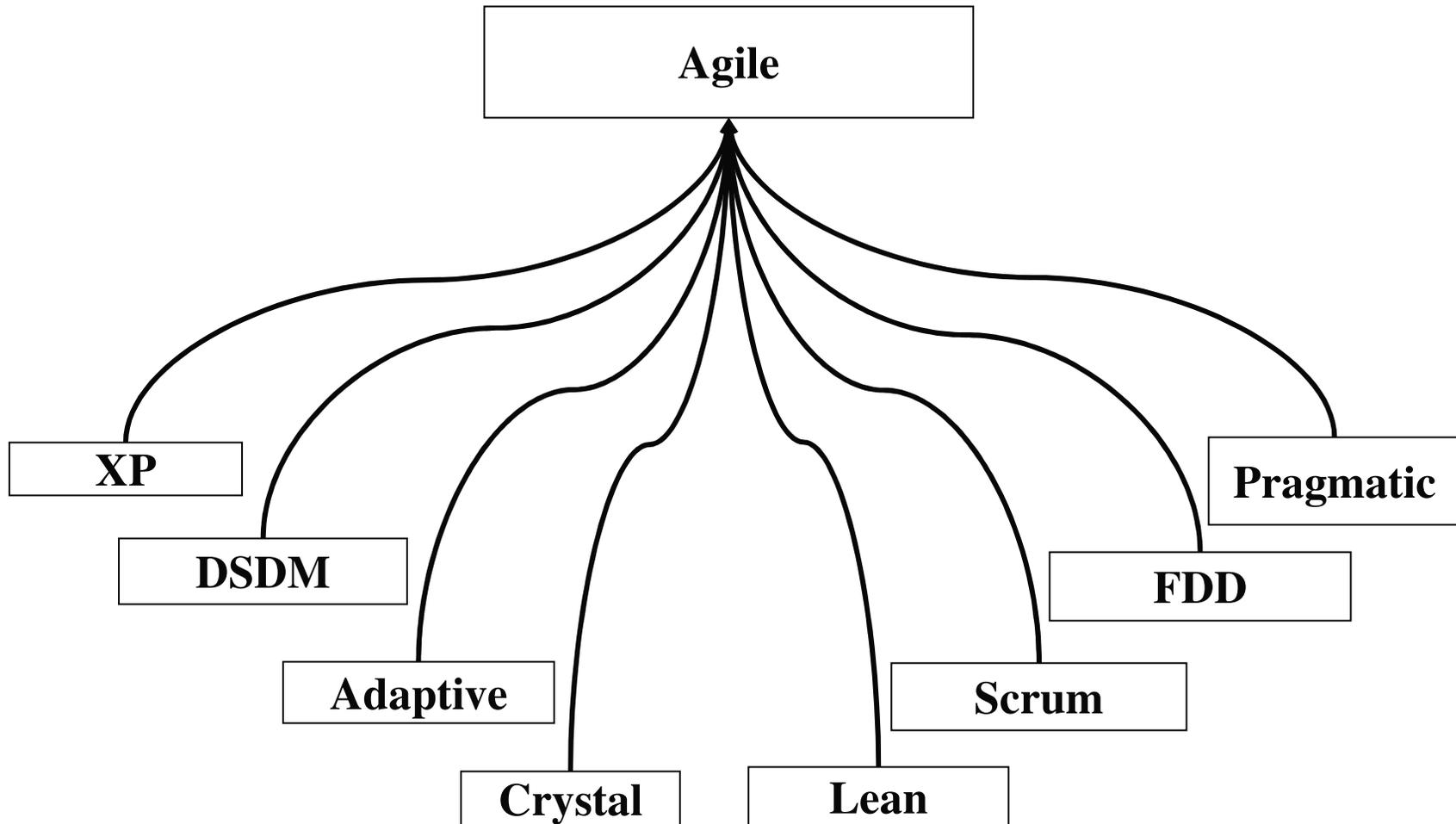
ThoughtWorks

Agile Software Community of India

Change is the only constant!

Stay Aware. Adapt. Change.

- Accelerate realization of business benefits
- Accommodate change
- Reduce risk
- Increase visibility (risks, schedule, etc.)
- Improve quality
  - Address business needs
  - Reduce missed and false features
  - Fewer defects
  - Increase maintainability



**Individuals and interactions** **OVER** Processes and tools

**Working software** **OVER** comprehensive documentation

**Customer collaboration** **OVER** contract negotiation

**Responding to change** **OVER** following a plan

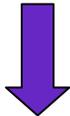
It is a system philosophy which views the *system* as a *chain*. The *weakest link is the constraint* that keeps the system from doing any better at achieving its goal.

The TOC is essentially about CHANGE

- What to change? (Where is the constrain)
- What to change to? (What should we do with the constraint?)
- How to cause the change? (How do we implement the change?)

- **Throughput** is the rate at which the entire system generates money through sales. Throughput is all the money coming into the system.
- **Inventory** is all the money the system invests in things it intends to sell, or all the money tied up within the system, e.g. raw materials, unfinished goods, purchased parts, investments in equipment and facilities.
- **Operating Expense** is all the money the system spends turning Inventory into Throughput. Operating Expense is all the money going out of the system, e.g. direct labor, utilities, consumable supplies, and depreciation of assets.

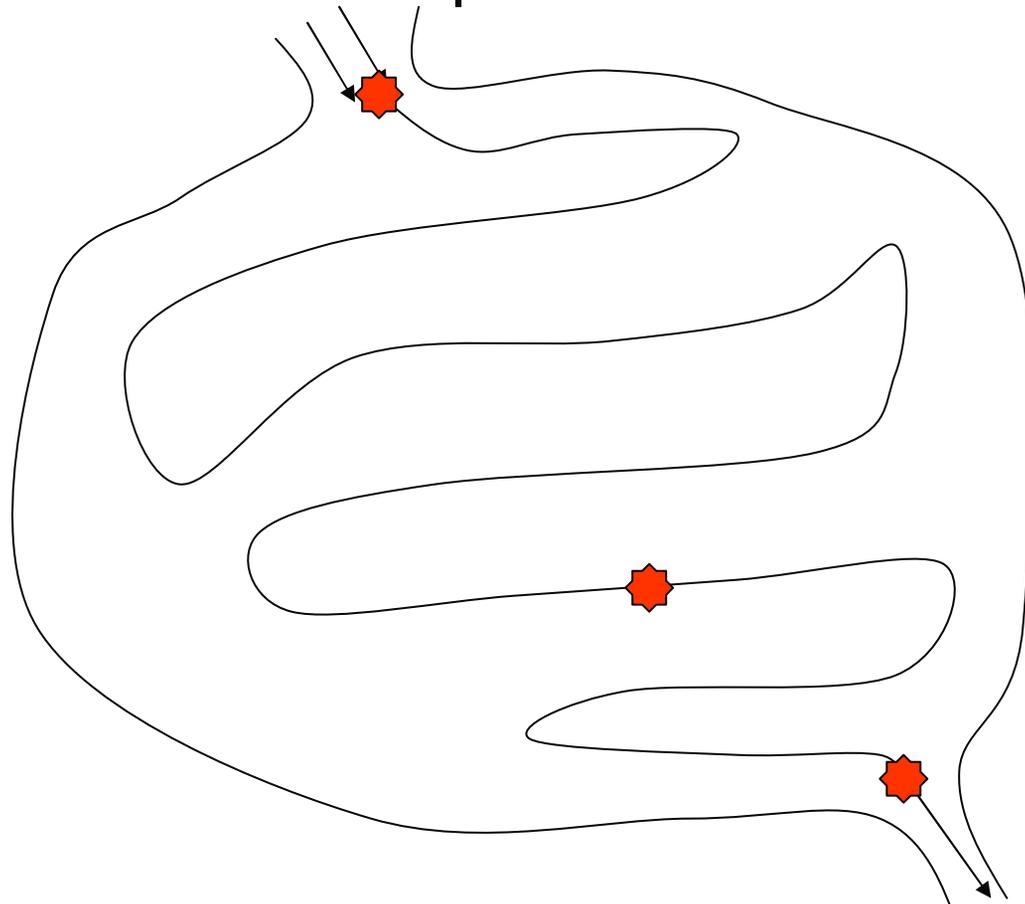
Will this

- Throughput 
- Inventory 
- Operating expenses 

How does your system look like?

ThoughtWorks®

Feature Request



Tested Working Features

- We rely on the *creativity and self-organization* of the team
  - Product owner and ScrumMaster handle external constraints
  - Team handles internal constraints
- It engages the people with the *profound knowledge of the system*
- Agile teams are typically *diverse teams of generalizing specialists*

- From “The Toyota Production System” by Taichii Ohno
  - Eliminate Waste
  - Expose problems – shorter feedback cycles
- From “Lean Software Development” by Mary and Tom Poppendieck
  - Engage work’s intelligence
  - Optimize the whole

- The Theory of Constraints is compatible with the Agile Movement and with Lean Thinking.
- It gives us thinking tools to identify and eliminate constraints.

“It is not necessary to change;  
survival is not mandatory.”

W. Edwards Deming

- **Business novels by Eliyahu Goldratt:**
  - *The Goal*, 2nd Edition, North River Press, 2004.
  - *It's Not Luck*, North River Press, 1994.
  - *Critical Chain*, North River Press, 1997.
  - *Necessary But Not Sufficient*, North River Press, 2000.
- **Theory books by William Dettmer:**
  - *Goldratt's Theory of Constraints*, ASQ Quality Press, 1997.
  - *Breaking the Constraints to World-Class Performance*, Mc Graw-Hill Education, 1998.
- **David Anderson: *Agile Management for Software Engineering*, Prentice Hall, 2003.**
- <http://www.agilealliance.org/>
- <http://www.xprogramming.com/>
- <http://www.controlchaos.com/>
- <http://martinfowler.com/>
- <http://c2.com/cgi/wiki>
- <http://www.testing.com/agile/>
- <http://www.pragmaticprogrammer.com>

## Thank You!

Naresh Jain

[nashjain@gmail.com](mailto:nashjain@gmail.com)

<http://jroller.com/page/njain>

<http://sf.net/projects/fitdecorator>

<http://sf.net/projects/patang>